Air Cargo Transport by Multi-Agent Based Planning

Keqiang Zhu, M.W.Ludema and R.E.C.M.van der Heijden
Department of Transportation and Logistics’ Organization
Faculty of Technology, Policy and Management
Delft University of Technology
P. O. Box 5015 2600 GA Delft
the Netherlands
{zhuk | m.w.ludema | robh}@sepa.tudelft.nl

Abstract
Most major airlines decide to take new strategies for the competition in the air cargo industry. They are concerned with assigning cargo over flights in creative ways. This paper considers the multi-agent solution for cargo assignment. First we present cargo control principles and introduce the pipelining assignment concept, then the difficulties of traditional operations research solving this assignment are pointed out. After explaining the motivations for agent solution, we propose a robust solution in which the flights are represented as agents and the cargo assignment is viewed as coalition formation process among these agents. Finally, future perspectives are summarized. The main solution idea is that the agents communicate and cooperate with each other to form the best coalitions of delivering the planned shipments, and coalitions get along with available information so that the control principles can be fully performed and air capacity can be best managed.

1. Introduction
Forecasts show that the air cargo industry will continue to grow and the competition will be strong. Many airlines find themselves being squeezed and their yields are dropping constantly. In contrast, however, excessive yields and profits reaped by successful integrated carriers like UPS and Federal Express suggest that customers might pay extra premium for very reliable and tailor-made services.

There are several factors that take airlines into the disadvantaged situations. First of all, the cargo delivery is closely tied up to the passenger flight schedules, which are normally designed to facilitate passenger transport. In contrast, integrators, who operate dedicated all-cargo aircraft, and sometimes even operate their own airports and/or terminals, can directly control the air transport networks. Second, airlines do business with forwarders, who act as representatives of their shippers, while integrators can directly access the shippers and therefore the market information. Third, a large volume of cargo forwarders promise do not show up, whereas forwarders contend their cargo doesn’t travel on the flight they booked.

Most major airlines have decided to use the following strategies to solve the problems:
- efficient usage of air network capacity by using new logistic concepts.
- reinforcement of collaboration with other air cargo companies, truck/train companies and forwarders.
- introduction of “time-definite” services, such as KLM Cargo’s SELECT range and Lufthansa Cargo’s td. series, which means that the cargo will be available at destination at an exact given time.

A number of new operating solutions should be developed to support the successful execution of those strategies, for some traditional ones are outdated. For example, traditional planning solutions to assigning cargo over flights do not support the "time definite” services, which promise “right time” and “right place” rather than “specific flight”.

One solution is the pipeline concept, which is raised by KLM Cargo with threefold purposes. The first purpose is for the reduction of its current 4.5 days average delivery time. The second one is for the operation of the above mentioned strategies. The third one is for the best use of the available air cargo transport capacity, since it is expected in the future the market demand will exceed the available air cargo transportation capacity at a faster pace.

To what extent the pipeline concept is helpful to solving the problems? what conditions are indispensable to a successful use of the concept? how airlines can tactically perform the concept? A lot of questions still have to be answered.
This paper focuses on the last question and considers the cargo assignment over flights using agent technology. The rest of the paper is structured as follows. In section 2, we first elaborate three cargo control principles for defining the pipelining assignment concept, and then explain the motivations for a multi-agent solution. In section 3, a pipelining assignment solution based on the creation of a multi-agent system is highlighted. Finally, in section 4, we summarize the major points and discuss future research.

2. Problem formulation

2.1. Pipeline concept and control principles

Pipelined networks are common in other industries such as telecommunications and gas supply, but they are quite new in the air cargo transportation and therefore there is no widely accepted definition for the pipeline concept.

According to KLM Cargo, the pipeline concept is a set of control principles by which shipments (transport orders with associated charges) are processed through a network of handling and transportation capabilities (with associated cost) at agreed quality and efficiency levels aiming at maximizing the system profit.

There are several points to note about this concept. Firstly, not being a single operating concept, the pipeline concept tries to reshape the way of thinking with regard to running an air cargo business [1]. Secondly, a new type of network, called the pipeline network, will be the result of this concept. Boubby Grin[2] classifies air transportation networks into three types: point-to-point, hub-and-spoke, and pipeline networks. The pipeline network is a multi-hub core system where hubs are connected by thick links through which the big volume aircraft frequently pass back and forth, and each hub has its own sub-hubs, origins and destinations serving a local region. Thirdly, a set of cargo control principles called pipeline control principles should be followed, so that network assets can be used optimally, while gaining a much more effective proximity to markets. Fourthly, the network structures and the control principles should work together. Pipeline control principles can be performed best under pipeline networks. Pipeline networks also require pipeline control principles running on them to be most effective.

There are three core cargo control principles in the pipeline concept, i.e. time split, place split and load split. The time split principle adjusts the arrivals and departures of shipments to fit the transport capacity. It can spread the variance in the number of orders and change the planned departure times of shipments. When, at a given moment, there is insufficient capacity, shipments planned to arrive later can be given priority. The principle is shown in Figure 1.

The place split principle considers rerouting a shipment. When there is not sufficient transport capacity on a certain route, the remainder of cargo can be transported by an alternative routing. This can improve the utilization of capacity on both routes. The principle of spreading shipments via alternative routes to the destination is illustrated by Figure 2.

The load split principle can be applied in two ways. Firstly, shipments composed of more than one package can be separated in different parts. These different parts can be treated differently (different route and arrival time), but have to be consolidated at the destination before the final delivery time. Secondly, when shipments are accepted, they are considered individually. Once they enter the network, they can be consolidated in containers called ULDs and thus a simpler control is possible. The different control levels are visualized in Figure 3.
parallel fulfillment of the three control principles as pipelining assignment. Pipelining assignment allow a shipment to be decomposed into parts (load split) that travel along diverse routes (place split) in different time frames (time split).

In this paper we focus on finding the solution being able to perform the pipelining assignment. In addition, to be able to consider real world details, the solution should also be robust, which means that:

- the assigning process should be incremental, that is, it is not necessary to wait for all the information (e.g., shipment orders and flight schedules) is available, and then to start the computation, and also it is not necessary to redo the computation from the very beginning each time when new information comes.
- the process is capable of exception handling, that is, in case of unanticipated events (e.g., a flight arrives late or a booked shipment is canceled), it should be easy to find alternative ways to handle the cargo.

### 2.2. Motivations for multi-agent techniques

The pipelining assignment can be considered to be a broad problem: transport resources allocation, which has been studied intensively by multiple Operational Research (OR) branches, including traffic assignment and vehicle routing since the 1950s. However, to our knowledge, no solutions can be directly derived from those branches for pipelining assignments, because they do not fit our context and concern. In an air cargo transportation system the legs and handling stations such as airports, consolidation centers, and warehouses constitute the transportation network while the movement of shipments carried by legs constitute the flow on the network. Furthermore, the planner cannot totally control schedules of legs as pointed out. In contrast, the traffic assignment studies the process by which demands in the form of trips are loaded on to the transportation network under the context where the road and rail infrastructure constitute the network while the movement of vehicles constitute the flows on the network[3]. The traffic assignment is primarily concerned with cost-driven travel behavior and the behavior-driven network equilibrium. At the equilibrium, all used paths are least cost paths and all paths that are not least cost are not used. The vehicle routing involves the design of a set of minimum cost routes for a fleet of vehicles of known capacity which services a set of customers with known demands. In the vehicle routing problem all routes have the same start/end locations, and decision makers can totally control those vehicles[4]. In spite of those differences the concepts and methods such as least cost path and multinomial logit model in the traffic assignment and the vehicle routing can give us insight when designing a pipeline assignment solution.

Relatively speaking, solutions just reflecting one or two pipeline control principles are easier to be found in OR literature. One of the recent findings is the solution of POWELL et al. for dynamic fleet management problem. Using a novel formulation, called a Logistics Queues Network (LQN)[5], POWELL et al. replace a big and complex problem by a series of very small problems. His solution is supposed to be able to consider several real-world details that cannot be modeled in traditional approaches to fleet management, such as shipment priorities. Allowing shipments to move within their time windows, the solution can realize the time split principle, but their assumptions that all legs are homogenous, and that the travel time of legs and time windows of shipments is exactly one or more times of standard stage (period), are rigid. Moreover, the other two principles cannot be realized.

Of all the found relevant published literature, the Multi-Commodity Path Flow Formulation (MCF-PATH) model, developed by J. Antes[6], is nearest to the pipelining assignment, and his Lufthansa Cargo research background also very fits our context. In this model they denote a shipment by an (o, d) - pair where the o represents the origin of the shipment and the d represents the destination of the shipment, then find the set of all feasible paths connecting the o and d by legs and finally treat the assignment of shipments into legs as a selection of the optimal combination of paths, which leads to a standard linear programming model. The model reaches a pipelining assignment solution, and is conceptually simple. There are some limitations to J. Antes’s method, however; shipments and legs are not allowed to be added dynamically during computation, and computation is easily intractable.

Large-scale legs and shipments, diverse possibilities for the split and multiple time and capacity constraints make pipelining assignments particularly complex. One reason for most OR solution failures in pipeline assignments is that they take centralized, top-down and sequential control principles that are not suitable to solve highly dynamic and complicated problems.

In the process of pipelining assignment, two kinds of actors are identified, i.e. legs and shipments. Actually, pipeline control principles remind us of the important fact that a group of legs cooperating together for the execution of a shipment can dig out potential benefits.

Recently, research in Multi-Agent Systems (MASs) has considered how individual ‘agents’ coordinate their knowledge and capacities to solve complicated and large-scale problems in situations where there is no possibility for centralized control. An MAS is formally defined as a loosely coupled network of problem solvers that interact to solve problems being beyond the individual capacities or knowledge of problem solvers[7]. MASs offer modula-
rity and distribution for handling complexity. If a problem domain is particularly complex, large, or unpredictable, a reasonable way is to develop a number of functionally specific and distributed modular components (agents) being tailored to solve a particular problem aspect [8]. When interdependent problems arise, the agents in the system must coordinate each other to ensure interdependencies to be properly managed. MASs are supposed to offer a way to relax the constraints of centralized, top-down and sequential control [9].

The agent metaphor provides a useful and natural way of modeling real world autonomous components. Over the last decade, there has been an increasing number of MASs for traffic applications, since traffic is highly dynamic, geographically and functionally distributed, and subsystems have a high degree of autonomy [10]. Agents in the transportation domain have been identified and modeled on various perspectives (see for instance [11], [12] and [13]). Some agents are users involved in traffic, others are means of transport (cars, trucks, trains, flights), elements of the traffic infrastructures, subsystems, and processes, or different branches and traffic modalities.

To improve performances of MASs, many researchers have focused on problems allocation, communication, coordination, negotiation and cooperation among agents [14]. What we propose in this paper is a pipelining assignment solution based on the creation of a multi-agent system, where the agents form coalitions to carry out their tasks optimally.

3. Definitions and solution

Before going into detailed solutions, it may be useful to specify some definitions.

3.1. Definition

First let us define agents in the multi-agent system. Corresponding to the natural entity, we refer to a leg of a certain capacity and arrival/departure times as an agent. A shipment may be performed in one of the following situations:

- a shipment is carried out by only one agent;
- a shipment is carried out by a group of agents and we refer to this group as a coalition;
- a shipment is firstly decomposed into several smaller sub-shipments and then each sub-shipment is carried out by an agent or a coalition.

If we regard an agent a special coalition consisting of a single member, then the shipment assignment process can be viewed as a coalition formation process. For example, suppose that a shipment transporting two tons of goods from A to E is assigned to a set of legs as shown in Figure 4. It can be said that we have formed three coalitions. The first coalition transports 0.5 ton of goods and includes three legs along the route A-B-C-E; the second coalition transports another 0.5 ton and includes two legs along the route A-D-E, and the third coalition transports the remaining 1.0 ton along the route A-E.

Three coalitions of legs for transporting a 2.0 tons’ shipment from A to E:

- Coalition 1: A-B, B-C, and C-E
- Coalition 2: A-D, D-E
- Coalition 3: A-E

Figure 4. An illustration for coalitions of legs

We assume that there is a set of agents, \( N = \{A_1, A_2, ..., A_n\} \) in the system, and each agent \( A_i \) represents one leg characterized by a vector of attributes:

\[
A_i := (\text{origin}_i, \text{destination}_i, \text{departure}_i, \text{arrival}_i, \text{capacity}_i, \text{space}_i, \text{the others}_i)
\]

There is a set of independent shipments, \( S = \{S_1, S_2, ..., S_m\} \), and each shipment \( S_i \) is described by a vector of attributes:

\[
S_i := (\text{origin}_i, \text{destination}_i, \text{available}_i, \text{due}_i, \text{weight}_i, \text{volume}_i, \text{charge}_i, \text{the others}_i)
\]

Here the capacity means the total weight a leg can carry out during a flight while the space means the whole volume a leg can contain during a flight. The available time refers to the earliest time a shipment is allowed to start transportation whereas the due time refers to the latest time a shipment can end transportation by air. The goods transported are generally heterogeneous. Therefore, the other attributes like fresh or frozen, live animals, flowers and plants, valuables and some types of chemicals are also set aside to imply special handling necessary.

A coalition is a group of agents who have decided to cooperate in order to achieve a common task. Here we are especially interested in a special type of coalition where the agents can form a new route by working together, and their arrival/departure times are in sequences. This type
of coalition is called the Pipelining Coalition (in short, PC). For instance, the agent flying from A to B and the agent flying from D to E cannot form a PC in Figure 4, but the agent with the attribute vector \( \{ A, D, 1998-09-22-10.00a.m., 1998-09-22-12.00a.m., \ldots \} \) and the agent with the attribute vector \( \{ D, E, 1998-09-22-2.00p.m., 1998-09-22-4.00p.m., \ldots \} \) can form a PC enabling a route from A to E. We further denote \( PC_j \) as any PC with a route starting from node i and ending at node j.

A PC will perform a shipment or part of a shipment along its route with a certain amount of consumption of each member’s capacity and space. Since a shipment may not consume all the resources of an agent, an agent may join another PC. When each agent can be a member of more than one coalition, we speak of overlapping coalitions.

Actually, a \( PC_j \) is a path or a tube connecting the origin i and destination j by a sequence of legs \( \{ A_1, A_2, \ldots, A_q \} \) with the following property:

- \( \text{origin}_{i_t} = i \)
- \( \text{destination}_{i_t} = \text{origin}_{i_{t+1}} \)
- \( \text{arrival}_{i_t} + \text{handling}_{i_t} + \text{departure}_{i_{t+1}} = t \)

For stage 1 of the solution, we will omit the consideration of handling activities in handling stations during transshipment stages, since a handling time period should be inserted into that between arrival_time of a former leg and departure_time of a latter leg to make a PC reasonable.

Extending the method of Onn Shehory et al. into a pipelining assignment and adapting it in the context of the air cargo transportation, we develop an optimal solution of three stages which can best use legs to maximize net incomes while satisfying constraints. The first stage, an initializing stage, serves to find all the permitted PCs and distribute, in the long term, calculation of their values among the agents. The second stage serves to make one PC perform one shipment, whereas the third stage allows more than one PC to perform one shipment. As a matter of fact, stage 2 makes PCs take initiative to select shipments while stage 3 makes shipments take initiative to select PCs, and their combination can accumulate net incomes achievable by performing shipments in most effective steps.

To simplify statements, we will omit the consideration of the other attributes of an agent and a shipment, but keep the designed solution easily generalized to those attributes.

### Stage 1:

Our pipelining assignment solution is based on the research in MASs, especially in distributed coalition formation. The coalition formation problem was first studied by N-Persons Game Theory[16]. Traditionally, Game Theory is concerned with payoff allocation among the members of a coalition and not with the coalition formation process itself. During the past few years, however, this process has received attention by researchers in the field of MASs and several solutions have been suggested. Among those, the work of Onn Shehory et al. [17] is most instructive and relevant. They propose an anytime, distributed method for task allocation via agent coalition formation where computational efforts are allocated to agents and the best coalitions are given priorities to achieve corresponding tasks. The deficiency of this method is that it is not a pipelining assignment since the decomposition of tasks is not considered, but it is still a good start for us to design the pipelining assignment solution because:

- the method is theoretically proven to be of low ratio bounds and low computational complexities and also tested by simulations and an implementation in an agent system. Therefore, based on his method, we need not worry too much about the computational complexity of distributed coalition formation process.
- the method is based on the assumption that no central controller distributes the tasks among agents and this assumption is desirable with respect to our situation where there are many dynamic agents. A central authority for coalition formation is costly in time and computational efforts in this situation. Moreover, a distributed manner can intuitively reflect simultaneous and parallel use of the three pipeline control principles.

Stage 1:
This stage is to find all the permitted PCs and arrange calculation of their values into each agent’s long-term commitment set.

1. For each \( i \) ( \( i = 1 \) to \( n \)), announce the origin, destination, arrival time and departure time of \( A_i \) to the other agents.

2. Find an agent \( A_j \) whose origin is the destination of \( A_i \) and whose departure time is later than the arrival time of \( A_i \).

3. If it is the first time when \( A_i \) contacts \( A_j \), collect information about its attributes.

4. Put a coalition \( \{ A_i, A_j \} \) into \( SPC \), the set of all the potential PCs including \( A_i \).

5. Commit \( A_i \) or \( A_j \) to calculation of the value of coalition \( \{ A_i, A_j \} \), and add \( \{ A_i, A_j \} \) into long-term commitment set \( LTC_i \) or \( LTC_j \).

6. Repeat contact other agents until there are no more agents to contact.

After the above steps, all PCs having two members can be found. Replacing \( A_i \) by a PC having two members, the steps similar to the above 2 to 6 steps could be repeated to get all the PCs that have three members. In the same way, all the permitted PCs including \( A_i \) are obtained.

As an example of stage 1, consider the network in Figure 5. In this network, PCs including agent \( A_1 \) and starting from \( A_1 \) are \( \{ A_1 \}, \{ A_1, A_2 \}, \{ A_1, A_4 \}, \{ A_1, A_2, A_3 \} \) and \( \{ A_1, A_2, A_3, A_6 \} \) while PCs including agent \( A_2 \) and starting from \( A_2 \) are \( \{ A_2 \}, \{ A_2, A_3 \} \) and \( \{ A_2, A_3, A_5 \} \). Although the number of PCs including agent \( A_1 \) and starting from \( A_1 \) is twice as many as the number of PCs including agent \( A_2 \) and starting from \( A_2 \), it does not necessarily mean that agent \( A_1 \) has to undertake the computational task for all those PCs. Agent \( A_2 \), for example, can share the computational task of agent \( A_1 \) by putting \( \{ A_1, A_2, A_3 \} \) into its long-term commitment set.

**Figure 5. An example network for the solution**

Note that the PCs of ‘cyclic flows’ are not permitted and that the number of members in a permitted PC without cyclic flow is less than the number of all the handling stations. By ‘cyclic flow’ is meant a PC that has connected some members to themselves at the ends. For instance, \( \{ A_1, A_2, A_3, A_6 \} \) forms a cyclic flow because the origin of its member \( A_2 \) is the destination of its another member \( A_6 \), and coalition \( \{ A_1, A_2, A_3, A_6 \} \) is obviously inferior to the simple coalition \( \{ A_1 \} \).

After the first stage, each agent \( A_i \) gets a set \( LTC_i \) consisting of the PCs for which it has been committed to calculate the values. In addition, \( A_i \) has all the necessary information about the attributes of the members of the PCs.

Stage 1 is executed only once unless there are new legs joining in during the computation.

**Stage 2:**

This stage is to compute values of all PCs and then select the best value among them to execute a corresponding shipment. Different from the traditional concept of coalition value, the value of a PC means the best achievable net income by performing one shipment selected from the remaining shipments to be assigned.

For each \( A_i \), select one PC called \( C_k \) from \( LTC_i \), and perform the following steps 1 – 3:

1. Calculate the maximum allowable throughput capacity and space of \( C_k \) by searching for the member with the smallest unused capacity and one of the smallest space in \( C_k \).
2. Calculate the throughput time period of $C_k$ by finding the member in $C_k$ with the earliest departure time and the member in $C_k$ with the latest arrival time.

3. Form the set $EC_k$ listing the expected net incomes from the shipments in $S$ when $C_k$ can perform them. For each shipment $S_j$ having the same origin and destination as $C_k$, perform:

   3.1. Check what capacity, space and due time are required for satisfying $S_j$.
   3.2. Compare the maximum allowable throughput capacity, space and the throughput time period with such requirements.
   3.3. If all the requirements are satisfied, calculate the expected net income $e_j$ of $S_j$ with regard to $C_k$ by an evaluation function as detailed in section 3.3 and put it into $EC_k$.

4. Announce the $EC_k$ that choose the best. This will be the coalition value $V_{C_k}$.

   Repeat the above steps for each PC in $LTC_i$.

5. Announce the $V_i$.

   Choose the best $V_{best}$ among all the announced values and the corresponding coalition $C_{best}$ and shipment $S_{best}$ will be selected.

6. Delete the shipment $S_{best}$ from $S$.

7. Update the capacities and space for members $C_{best}$ in terms of the consumption for executing the $S_{best}$.

8. Announce the remaining capacities and space.

9. Return to Stage 2.

As an example of stage 2, consider a simple situation where there are four shipments with the following attributes:

- $S_1 := (A, B, _ , _ , 0.5, _, 5, _ )$
- $S_2 := (A, D, _ , _ , 1.5, _, 20, _ )$
- $S_3 := (A, E, _ , _ , 1.0, _, 15, _ )$
- $S_4 := (B, E, _ , _ , 1.0, _, 21, _ )$

and where there are six agents in Figure 5 with the following attributes:

- $A_1 := (A, B, _ , _ , 10, _ , 1.0, _ )$
- $A_2 := (B, C, _ , _ , 10, _ , 1.5, _ )$
- $A_3 := (C, D, _ , _ , 5, _ , 0.5, _ )$
- $A_4 := (B, D, _ , _ , 5, _ , 1.0, _ )$
- $A_5 := (D, E, _ , _ , 10, _ , 1.0, _ )$
- $A_6 := (D, B, _ , _ , 10, _ , 0.5, _ )$

and the seventh attribute of an agent is the cost for transporting one unit weight’s shipment.

If the evaluation function is based on traditional accounting as indicated in section 3.3 and the handling cost is set to zero, we can get Table 1 after following stage 2 once and the table indicates which shipment should be first selected for execution and by which PC. According to the table and this example, shipment $S_4$ should be first selected to be carried out by coalition $\{A_4, A_6\}$.

<table>
<thead>
<tr>
<th>PCs</th>
<th>Coalition values (net income)</th>
<th>Best</th>
<th>Best of the best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Including and starting from $A_1$</td>
<td>$A_1$</td>
<td>4.5</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>$A_1, A_2$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$A_1, A_3$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$A_1, A_4, A_5$</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>$A_1, A_2, A_5$</td>
<td>15.5</td>
<td>15.5</td>
</tr>
<tr>
<td></td>
<td>$A_1$</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>$A_1, A_6$</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Including and starting from $A_2$</td>
<td>$A_2$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$A_2, A_3$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$A_2, A_4, A_5$</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>$A_2$</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>Including and starting from $A_3$</td>
<td>$A_3$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$A_3, A_4, A_5$</td>
<td>29</td>
<td>29</td>
</tr>
</tbody>
</table>

Table 1. An illustration for calculation in stage 2

where, -- represents no shipment is available to be carried out by this coalition.

Generally the second stage should repeat until no shipments can be assigned to a single PC. The remaining shipments are denoted as the set $S_i = \{S_1, S_2, ..., S_i\}$. Because it is not possible that a single PC can carry out a shipment at this moment, coalitions among the remaining PCs are necessary to carry out the remaining shipments. The next stage is to allocate a shipment to more than one PC. To do this, a shipment should be decomposed into several pieces and each piece is carried out by a PC. Here we still take an agent’s view towards PCs, so that multi-
agent theories can further be used to guide the design of relevant steps.

The implementation of the interaction among agents in MAS is often based on protocols by which it is possible for agents to get access to relevant information each other. The most early and widely applied protocol is the Contract Net Protocol of Smith and Davis. Its main idea is: a certain task is assigned to a society of agents. One agent, called the manager, processes the task and decomposes it into a group of subtasks. He announces these tasks step by step to a set of capable agents, based on his knowledge about them. They respond to the task announcement by ranking it into stored tasks currently under consideration. When time is available, the agents prepare bids for these tasks and send them to the manager in question. The manager processes the incoming bids and chooses the best one before an expiring time.

Although this protocol is powerful in many cases, it does not match our assumption that there is no central authority to allocate the tasks and so modifications are required. To design stage 3, we follow the similar spirit of the ECN protocol of Klaus Fischer et al. [12], a variant of the Contract Net Protocol where, instead of offering partial shipments, shipments are directly announced to each \( A_i \), and bids for partial shipments are possible. The main difference between the ECN protocol and the contract net protocol can be illustrated by a simple example. Suppose that there are two legs \( \{ A_1, A_2 \} \) in a society of agents, where \( A_1 \) has 2 tons transportation capacity available and \( A_2 \) has 3.5 tons capacity. A manager without precise knowledge of each leg’s capacity and willingness to carry out some task but understanding that no leg can handle the whole shipment individually, has received a shipment transporting 5 tons freight. In the light of Contract Net Protocol, it is likely that he decomposes the 5 tons shipment into two 2.5 tons sub-ships, and announces the first 2.5 tons to each of the two legs. Naturally he may get no bid from \( A_1 \) but gets a bid from \( A_2 \). Then he announces the second one and receives no bid from either of them. As a result, the whole shipment cannot be performed as expected. Instead of announcing sub-ships, the whole shipment in the ECN protocol is directly informed to each PC which can, to its benefit, cut a piece from the whole. In this way the manager is allowed to have more chances to understand these PCs and each PC has also more freedom to express its preference.

**Stage 3:**

This stage aims at allocating the remaining shipments to the remaining PCs and consists of the following steps:

1. For each \( i \) ( \( i = 1 \) to \( p \)),

   Announce the shipment \( S_i' \) to each remaining agent

2. For each remaining \( A_j \), select one PC called \( C_j \) from \( LTC_j \)

   If \( C_j \) can partially carry out \( S_i' \), then calculate a bid \( B^j_{iA} \) denoted as a vector:

   \[
   B^j_{iA} = \{ weight^j_{iA}, space^j_{iA}, V^j_i \}
   \]

   where \( V^j_i \) is the net income expected from partially performing \( S_i' \) by \( C_j \).

   Repeat until all the members in \( LTC_j \) are selected

3. Announce the \( B^j_{iA} \)

   Cover \( S_i' \) by these possible \( B^j_{iA} \) and find a feasible coalition of PCs with the best coalition value \( V_j \). Call this coalition \( BPC_j \)

4. Announce the \( V_j \)

5. Choose the best among all the announced values. The \( V_{best} \) will be chosen by all the remaining PCs. The \( V_{best} \) is corresponding to coalition \( BPC_{best} \)

6. Update \( S_i' \) and the capacities and space in terms of consumption for executing the \( S_{best}' \)

7. Return to step 1

As an example of stage 3, consider a situation where there is one shipment \( S_3 \) with the following attributes:

\[
S_3 := ( A, E, _ , _, 1.0, _ , 15, _ )
\]

and where there are six agents in Figure 5 with the following attributes:

\[
A_1 := ( A, B, _ , _, 1.2, _ , 1.0, _ )
A_2 := ( B, C, _ , _, 0.7, _ , 1.5, _ )
A_3 := ( C, D, _ , _, 0.8, _ , 0.5, _ )
A_4 := ( B, D, _ , _, 0.4, _ , 1.0, _ )
A_5 := ( D, E, _ , _, 1.5, _ , 10, _ )
A_6 := ( D, B, _ , _, 3.0, _ , 0.5, _ )
\]

and the fifth attribute of an agent refers to the remaining capacity of the agent.

When the shipment \( S_3 \) is announced to the agents, there will be two bid vectors: \((0.7, _, 7.7)\) and \((0.4, _, 4.8)\) issued by coalition \( \{ A_1, A_2, A_3, A_4 \} \) and \( \{ A_1, A_5, A_6 \} \) respectively. Since \( \{ A_1, A_2, A_3, A_4 \} \) can reach more net income per unit weight than \( \{ A_1, A_2, A_3, A_5 \} \), \( \{ A_1, A_2, A_3, A_6 \} \) is being negotiated to carry out 0.6 ton’s shipment \( S_3 \) rather than 0.7 ton, so that the net income by \( \{ A_1, A_2, A_3, A_4 \} \) and \( \{ A_1, A_2, A_3, A_5 \} \) together for carrying out \( S_3 \) is 11.4.
It should be pointed out that stage 2 and stage 3 are not necessarily in sequence during the computational process. Such an arrangement is due to the consideration that much more computational time is required to decompose a shipment. Also in practice, the number of whole shipments is generally much greater than divided shipments. Fortunately, to some special shipments it is rather easy to start stage 3 first and then stage 2 because stage 2 and stage 3 have been intentionally designed to be independent enough. As a matter of fact, stage 2 and stage 3 can easily exchange their computational orders as much as needed.

Obviously, the solution is very loyal to the three cargo control principles. In addition, this solution is also robust, for it can be easily extended to deal with the situation where a shipment and/or a leg join into the system dynamically. When a new shipment joins in, there is no necessity to execute stage 1. When a new leg joins in, due to a new flight or a new handling station connected, reinitializing the execution of stage 1 is necessary.

3.3. Evaluation function

An evaluation function is constructed to calculate the net income of a PC = {A_1, A_2, ..., A_p} to perform a shipment. The net income relates to the charge received from performing a shipment and the corresponding costs. We introduce two alternative methods to build up an evaluation function. Based on traditional accounting, one method is to compute the net income of a shipment. The net income relates to the charge received from performing a shipment and the corresponding costs.

\[
\text{net income} = \text{charge} - \text{costs}
\]

where, \( \text{charge} \) is the charge received from the shipment, and \( \text{costs} \) is the cost associated with the shipment.

The other method is based on the theory of constraints (TOC) [18], implying that the most scarce resources should first be used by the most valuable shipments. An advantage of the second method is that the former tries to maximize system throughput whereas the latter attempts to optimize both profit and capacity. An advantage of the second method is that

![Table 2. Identifying the bottleneck leg in a PC](image)

Secondly, we locate the most valuable shipment \( S_{\text{moc}} \) from \( S \) by calculating the following value for each shipment in \( S \):

\[
\text{charge}_{S_{\text{moc}}} = \sum_{k=1}^{q} \text{weight}_{k} \times h(A_{(i-1)}, A_{i})
\]

and finding the maximal one. In TOC language,

\[
\text{charge}_{S_{\text{moc}}} = \sum_{k=1}^{q} \text{weight}_{k} \times h(A_{(i-1)}, A_{i})
\]

can be called the throughput of \( S_{\text{moc}} \) and denoted as \( \text{Thr}_{S_{\text{moc}}} \), and repeat the above for each PC in \( \text{LTC}_i \).

Thirdly, announce (A_{\text{moc}}, Thr_{S_{\text{moc}}}, PC_{i}).

Fourthly, locate the most frequently appearing leg \( A_{\text{moc}} \) from all announced triples (A_{\text{moc}}, Thr_{S_{\text{moc}}}, PC_{i}), search the \( PC_{\text{max}} \) in all the triples (A_{\text{moc}}, Thr_{S_{\text{moc}}}, PC_{i}) of \( A_{\text{moc}} \), which can produce the maximal throughput \( Thr_{S_{\text{moc}}} \), arrange \( PC_{\text{max}} \) to carry out corresponding shipment \( S_{\text{moc}} \) and return to step 7 in stage 2 for updating relevant information.

Although \( A_{\text{moc}} \) is a bottleneck leg in \( PC_{i} \), it is very likely that the route starting from origin to destination is connected and not a bottleneck in the whole air cargo network. Thus, announcing a vector (A_{\text{moc}}, Thr_{S_{\text{moc}}}, PC_{i}) instead of a scalar like \( Thr_{S_{\text{moc}}} \) and selecting the most frequently appearing leg from all announced triples, the bottleneck route in the whole air cargo network could be recognized.

The chief difference between the first method and the second one is that the former tries to maximize system profit whereas the latter attempts to optimize both profit and capacity. An advantage of the second method is that
it doesn’t require information about unit transport cost for each leg, which is not accessible in many cases.

4. Future research

Now a pilot study is being undertaken to implement the above-mentioned solution on a local transportation network of KLM Cargo. The study aims at understanding the effectiveness of the solution, investigating its computational complexity and laying a basis for final expansion of the solution to the whole network based on real data.

To be representative, the local network is selected on the basis of the following two criteria:

- demand of shipments exceeds supply of transport capacity,
- there are a lot of time defined shipments,

which aim to make the effectiveness of the solution evident.

At this stage the agent technique is just applied to the pipelining assignment, and only one type of agent (i.e. leg) is introduced. The agent is ‘stupid’ in the sense that it will cooperate with any other agents without considering its self-interest. The agent (i.e. leg) is assumed to have only one interest, i.e. to improve the global benefit wholeheartedly. All the agents seem to belong to one owner. Strictly speaking, such an assumption is unrealistic because no single company can efficiently operate the whole network. Generally an air cargo transportation network is collectively owned and run by multiple players, each of them having different interest, perception, intention, position, and resources. Later on, the ‘smart’ type of agent, representing those players will be introduced. In the future study, the agent technique will be applied to the network level. The agent structures, the architecture of the whole air cargo transportation multi-agent system, the interactions among smart agents, the interactions between a smart agent and a stupid agent, will be considered.

References