Coordinated Multimodal Transportation

An Agent-Based Approach

TRAIL Research School, Delft, November 2002

Authors
Ir. Jeroen Valk
Faculty of Information Technology and Systems, Delft University of Technology

© 2002 by Ir. Jeroen Valk and TRAIL Research School
Abstract

1 Introduction ................................................................. 75
1.1 Multi-agent planning approaches to multimodal transportation .... 76

2 The Coordination Problem for Task-Oriented Agents............... 78

3 A Simple Coordination Method ........................................ 82

4 Plan coordination by partitioning local plans....................... 84
4.1 Partitioning using only local dependency information ........... 84
4.2 A refined partitioning strategy ...................................... 85

5 Conclusions .................................................................... 88

References........................................................................... 89
We consider the problem of coordinating the activities of autonomous companies in multimodal transportation problems. An agent-based approach is presented to solve these coordination problems based on the design of coordination protocols and coordination algorithms. The emphasis is on the possible effects of coordination protocols and algorithms on the performance of multi-agent systems. In particular, we use a systematic method to design coordination algorithms with good lower bounds on the performance of the multi-agent planning system. In this way, we test protocols for their ability to coordinate the multi-agent system.

A coordination protocol is studied that can successfully coordinate the activities of agents that minimize their total workload. This protocol is designed for task-oriented problems where several agents, each assigned to some subtask of a complex task, solve their own subtask by making minimal plans and want to find a common plan based on their individual plans. A task is conceived as a set of primitive tasks (operations), partially ordered by a set of precedence constraints. Operations are distributed among agents dependent on their capabilities and constitute the subtasks the agents have to solve. The precedence constraints between operations in subtasks are inherited from the overall precedence constraints occurring in the task. Since it is assumed that every agent is capable to find a suitable (minimal) plan for its own subtask, the main problem for the agents is to coordinate their plans in order to solve the complete task. Depending on how much knowledge agents have about the global task to be carried out the agents can coordinate their activities into a solution that approximates a globally optimal plan within a factor \( d \) where \( d \) the the depth of the original task. If agents have knowledge about the distribution of tasks over the agents a better approximation can be achieved.

**Keywords**

multimodal transportation, coordination, multi-agent systems
In our research we focus on the application of multi-agent technology into multimodal transportation problems. Such problems are characterized by transportation orders that require an integration of existing transportation modes into seamless transportation services. Typically, these existing transportation modes are offered by autonomous transportation companies that each individually are not capable to perform multimodal transportation orders. Hence, the solution to a typical multimodal transportation problem requires the cooperation of several autonomous organizations, each having their own capabilities, interests and resources. For example, intercontinental freight transport orders require the cooperation of truck companies responsible for the pre- and post-transportation part to and from an airport and airline companies carrying out the long-distance part of the order.

In real life, transportation companies are not involved in handling just one transportation order, but they have to handle several (parts of) different orders simultaneously. So, in reality, we have to deal with the problem of handling of several orders, where each order may require the coordinated application of several parties to complete it. So, let us assume that we have such a set of multimodal orders and a set of transportation companies (parties) to complete them. Then these parties are confronted with (at least) three interrelated problems.

First of all, assuming that their transportation capabilities might overlap, it has to be decided which party takes which part of an order to complete. That is, assuming that each order can be split up in elementary tasks such that each task can be handled by one party, these elementary tasks have to be assigned to the companies.

Secondly, each party has to come up with a plan for its own activities in order to handle the different parts of the orders it is responsible for. So, for example, a truck company responsible for carrying out parts of the intercontinental orders pertaining to the European continent will be interested in finding a good transportation plan for its parts of the orders. Likewise, an airline will be interested in finding an efficient air cargo schedule to complete its part.

Thirdly, since the tasks (within each order) are interrelated, the parties have to coordinate their plans in order to be able to execute the orders. This requires an analysis of their task dependency relations. Note that the quality of the resulting solution (i.e. the time and costs of executing the orders) in general will be a hard to compute function of task assignments, individual plans and their coordination.

To solve these problems, so-called multi-agent technology, dealing with autonomous systems that have to cooperate in order to accomplish a joint set of tasks, is becoming increasingly important. Here, an agent is an entity that is capable to make plans given some set of resources in order to fulfill some goals. In our multimodal transportation problems, agents represent autonomous transportation companies that are willing to cooperate to perform multimodal transportation jobs, but may have their own interests and private plans that are not visible to other parties.

The goal of our research is to design and analyze automated or semi-automated agents representing the companies involved. Due to advancements in information and communication technology, such automated agents should be able to interact in an open environment and should perform a large amount of planning, negotiation- and coordi-
Abstract

1 Introduction ................................................................. 75
1.1 Multi-agent planning approaches to multimodal transportation ....... 76

2 The Coordination Problem for Task-Oriented Agents .................... 78

3 A Simple Coordination Method ........................................... 82

4 Plan coordination by partitioning local plans ............................... 84
4.1 Partitioning using only local dependency information .................. 84
4.2 A refined partitioning strategy .......................................... 85

5 Conclusions ......................................................................... 88

References ............................................................................. 89
We consider the problem of coordinating the activities of autonomous companies in multimodal transportation problems. An agent-based approach is presented to solve these coordination problems based on the design of coordination protocols and coordination algorithms. The emphasis is on the possible effects of coordination protocols and algorithms on the performance of multi-agent systems. In particular, we use a systematic method to design coordination algorithms with good lower bounds on the performance of the multi-agent planning system. In this way, we test protocols for their ability to coordinate the multi-agent system.

A coordination protocol is studied that can successfully coordinate the activities of agents that minimize their total workload. This protocol is designed for task-oriented problems where several agents, each assigned to some subtask of a complex task, solve their own subtask by making minimal plans and want to find a common plan based on their individual plans. A task is conceived as a set of primitive tasks (operations), partially ordered by a set of precedence constraints. Operations are distributed among agents dependent on their capabilities and constitute the subtasks the agents have to solve. The precedence constraints between operations in subtasks are inherited from the overall precedence constraints occurring in the task. Since it is assumed that every agent is capable to find a suitable (minimal) plan for its own subtask, the main problem for the agents is to coordinate their plans in order to solve the complete task. Depending on how much knowledge agents have about the global task to be carried out the agents can coordinate their activities into a solution that approximates a globally optimal plan within a factor $d$ where $d$ the the depth of the original task. If agents have knowledge about the distribution of tasks over the agents a better approximation can be achieved.

**Keywords**

multimodal transportation, coordination, multi-agent systems
In our research we focus on the application of multi-agent technology into multimodal transportation problems. Such problems are characterized by transportation orders that require an integration of existing transportation modes into seamless transportation services. Typically, these existing transportation modes are offered by autonomous transportation companies that each individually are not capable to perform multimodal transportation orders. Hence, the solution to a typical multimodal transportation problem requires the cooperation of several autonomous organizations, each having their own capabilities, interests and resources. For example, intercontinental freight transport orders require the cooperation of truck companies responsible for the pre- and post-transportation part to and from an airport and airline companies carrying out the long-distance part of the order.

In real life, transportation companies are not involved in handling just one transportation order, but they have to handle several (parts of) different orders simultaneously. So, in reality, we have to deal with the problem of handling of several orders, where each order may require the coordinated application of several parties to complete it. So, let us assume that we have such a set of multimodal orders and a set of transportation companies (parties) to complete them. Then these parties are confronted with (at least) three interrelated problems.

First of all, assuming that their transportation capabilities might overlap, it has to be decided which party takes which part of an order to complete. That is, assuming that each order can be split up in elementary tasks such that each task can be handled by one party, these elementary tasks have to be assigned to the companies.

Secondly, each party has to come up with a plan for its own activities in order to handle the different parts of the orders it is responsible for. So, for example, a truck company responsible for carrying out parts of the intercontinental orders pertaining to the European continent will be interested in finding a good transportation plan for its parts of the orders. Likewise, an airline will be interested in finding an efficient air-cargo schedule to complete its part.

Thirdly, since the tasks (within each order) are interrelated, the parties have to coordinate their plans in order to be able to execute the orders. This requires an analysis of their task dependency relations. Note that the quality of the resulting solution (i.e. the time and costs of executing the orders) in general will be a hard to compute function of task assignments, individual plans and their coordination.

To solve these problems, so-called multi-agent technology, dealing with autonomous systems that have to cooperate in order to accomplish a joint set of tasks, is becoming increasingly important. Here, an agent is an entity that is capable to make plans given some set of resources in order to fulfill some goals. In our multimodal transportation problems, agents represent autonomous transportation companies that are willing to cooperate to perform multimodal transportation jobs, but may have their own interests and private plans that are not visible to other parties.

The goal of our research is to design and analyze automated or semi-automated agents representing the companies involved. Due to advancements in information and communication technology, such automated agents should be able to interact in an open environment and should perform a large amount of planning, negotiation- and coordi-
2 The Coordination Problem for Task-Oriented Agents

We consider a multi-agent approach to planning problems where completing a task requires the cooperation between several parties.

To further motivate our approach, let us consider a running example of a simple multi-modal transportation problem. Suppose we have three transportation companies (agents), denoted \{train, truck, ferry\}: agent train can pickup and deliver cargo in cities \{LON, PAR, ROM\}, agent truck in cities \{AMS, BER, PAR\} and ferry in cities \{AMS, DUB, PAR\}. For each agent and pair of cities \(s, t\) it can handle, the time is listed to deliver packages starting in \(s\) and delivering it in \(t\) (see Figure 1). Suppose there are two packages to be transported\(^1\): a package from BER to LON and a package from ROM to AMS. In this simple setting, coordination between train and truck is required to accomplish the two transportation tasks. A possible way to execute the tasks is easy to find: first travel from Berlin to Paris with the truck and from Rome to Paris with the train, simultaneously, exchange the orders at Paris and travel by train to London and by truck to Amsterdam. As this example shows, a multimodal transportation problem can be splitted up in several to-be-coordinated single-agent planning problems. The focus in this paper will be on the coordination part.

To coordinate the activities in a multi-agent system, it is common practice to implement: (i) a communication protocol, and (ii) distributed coordination algorithms. The coordination algorithms are responsible for computing proposals to increase the revenues of agents by achieving consensus and to evaluate proposals received from other agents. Communication of proposals is carried out following the rules of the communication protocol. The result of a coordination method consists essentially of a family of agreements among groups of agents. Each of these agreements are restricting in a sense that each participant is supposed to act according to the agreement. If agent conduct is coordinated before planning and plan execution, the agreements among a group of agents consists of a is a family \(\{G_a\}_{a \in C}\) of goals: each goal \(G_a\) constrains the possible courses of action of an agent. Indeed, each agent \(a \in C\) is expected to come up with a so-called feasible plan \(p_a\) that realizes the goal \(G_a\). Next, these feasible plans \(\{p_a\}_{a \in C}\) can be combined into of joint plan for the agents. If coordination before planning is successful, the plans \(p_a\) can easily be combined into a joint plan, but sometimes combining the plans requires a plan merging coordination method after planning by the individual agents has taken place.

We study coordination in a task-oriented setup where the plans of the agents can easily be combined into a joint plan, but where additional plan merging can be used to further improve the quality of the joint plan. In this paper, we will focus on coordination before planning methods only.

In our setup, a complex task \((T, \prec)\) consists of a partially ordered set of operations, called primitive tasks. These primitive tasks \(t \in T\) have to be executed in a sequence specified by the partial order \(\prec\). Agents are systems capable to perform those primitive tasks and to make plans for executing them. Every single agent, however, is only capable to perform some subset of primitive tasks. Therefore, execution of the complete task will usually require several agents to cooperate on performing all operations. In
real life, agents will accept several complex tasks $T_i$ simultaneously. So we consider multi-agent planning and scheduling problems specified by:

- a set $\mathcal{T}$ of primitive tasks the agents can collectively perform;
- a set $\mathcal{A}$ of agents: each agent $a \in \mathcal{A}$ is capable of performing a subset $\mathcal{T}_a \subseteq \mathcal{T}$ of primitive tasks, i.e., $\mathcal{T}_a$ specifies the skills of agent $a$;
- a family $\{(T_i, \prec_i)\}_{1 \leq i \leq n}$ of complex tasks to be executed: the primitive tasks $t \in T_i$ must be executed in an order that respects the precedence constraints specified by the partial order $\prec_i$.

For example, pickup/delivery pairs in our simple logistic problem are primitive tasks, i.e., $\mathcal{T}$ consists of ordered pairs $(v_1, v_2)$ with $v_1, v_2 \in \{\text{AMS}, \text{BER}, \text{DUB}, \text{LON}, \text{PAR}, \text{ROM}\}$. If a pickup/delivery pair is part of a logistic chain it may be dependent on other primitive tasks, e.g., $(\text{ROM}, \text{PAR})$ must have been completed before execution of $(\text{PAR}, \text{AMS})$ is started. The agents can only perform subsets $\mathcal{T}_a \subseteq \mathcal{T}$ such that for each $(v_1, v_2) \in \mathcal{T}_a$ agent $a$ is capable to visit both $v_1$ and $v_2$. Therefore, the transportation orders $(\text{BER}, \text{LON})$ and $(\text{ROM}, \text{AMS})$ cannot be carried out by a single agent and need to be decomposed into complex tasks. Routing both transportation orders through Paris, we obtain the complex tasks $(T_1, \prec_1)$ and $(T_2, \prec_2)$ with primitive tasks

$$T_1 = \{(\text{BER}, \text{PAR}), (\text{PAR}, \text{LON})\}, \text{and}$$

$$T_2 = \{(\text{ROM}, \text{PAR}), (\text{PAR}, \text{AMS})\},$$

and the precedence constraints shown in Figure 2. In general, we assume that the cus-
Figure 2: Precedence constraints of transportation tasks $T_1$ and $T_2$.

Figure 3: An optimal task schedule.

tomer selects a logistic chain. So we avoid problems of finding chains for routing the packages, which can become very difficult for large infrastructures with many orders.

To avoid cumbersome notation, without loss of generality, we may write the family of complex tasks $\{ (T_i, \prec_i) \}_{1 \leq i \leq n}$ as a single complex task $(T, \prec)$, since the union $T = \bigcup_{i=1}^{n} T_i$, together with the union of their precedence constraints $\prec_i$, is just a partially ordered set of tasks. Therefore, we will sometimes specify a (complex) task by a single tuple $(T, \prec)$. Such a task $(T, \prec)$ can be completed by a set of agents $a \in \mathcal{A}$ if (i) all primitive tasks contained in $T$ have been assigned to agents $a$ capable of performing them and (ii) the agents have agreed upon a joint plan $p_A$ to execute the tasks in $T$, respecting the precedence constraints $\prec$.

Given this setup, the coordination problem is: given a complex task $(T, \prec)$, find subgoals that, when assigned to the agents, restrict the agents to execute only plans that can be easily combined into a coordinated solution. We assume that the subgoals assigned to an individual agent $a$ are given by a partially ordered set $G_a = (T_a, \prec_a)$. A plan $p$ specifies when the execution of tasks is started and when the execution is finished. Thus, a plan is a kind of task schedule specifying for each task the time interval in which the task is carried out. Figure 3 shows an example task schedule for an optimal solution of our running example. A plan $p$ is feasible for a subgoal $G_a = (T_a, \prec_a)$ if: (i) the transportation unit is capable to execute the plan, (ii) all tasks in $T$ are scheduled in $p$, and (iii) for all tasks $t_1, t_2 \in T$ with $t_1 \prec t_2$, the execution of $t_1$ is scheduled to be
the agents perform all tasks in $T$, i.e., $T \subseteq \bigcup_{a \in A} T_a$, and the plans of the agents respect the global precedence constraints $\prec$.

Given a set of agents $A$ and a complex task $(T, \prec)$, we will assume that agents have already agreed upon a particular assignment of subsets $T_a$ to agents $a$. By distributing the set $T$ of tasks, the total set of precedence constraints $\prec$ is split up into sets $\prec_a$ of \textit{intra}-agent constraints and a set $\prec_{\text{inter}}$ of \textit{inter}-agent constraints. That is:

- $\prec_a = \prec \cap (T_a \times T_a)$ is the set of constraints agent $a$ has to take into account, while
- $\prec_{\text{inter}} = \prec - (\bigcup_{a \in A} \prec_a)$ is the set of constraints \textit{between} the agents.

So $(T_a, \prec_a)$ denotes the \textit{subtask} which agent $a \in A$ is minimally required to carry out. Note that $\prec_a$ and $\prec_{\text{inter}}$ again are partial orders. We call $d_a$ the \textit{depth} of $(T_a, \prec_a)$, that is the length of the longest chain in $(T_a, \prec_a)$.

For every agent $a \in A$ it is important to execute its task efficiently. We assume that the cost $c(p)$ of a plan equals the total amount of time that at least one task is scheduled in $p$. Thus, in our running example, the cost of a plan equals the total amount of time a transportation resource is traveling to carry out a pickup/delivery order. A plan $p$ is minimal for a subgoal $G_a$ if it is feasible for $G_a$ and its cost $c(p)$ is minimal among all plans $p'$ feasible for $G_a$.

From now on we assume that every agent is able to construct locally optimal and feasible plans, i.e. the plan $p \in G_a$ chosen by agent $a$ is locally optimal.
3 A Simple Coordination Method

Given this simple setup we would like to investigate how to coordinate by computing subgoals such that a global plan can be constructed satisfying all precedence constraints by simply combining single-agent plans for the subgoals and how these coordination methods affect the quality of the resulting plan, i.e., what the costs are of the resulting coordinated plan compared with a globally optimal plan for the task at hand.

A very simple coordination method is to give the agents a complete freedom to come up with its minimal plan \( p_a \) feasible for \((T_a, \prec_a)\). Hereafter, by a simple cooperation protocol, the agents would coordinate their plans to satisfy also the inter-agent precedence constraints without affecting their own local plan, i.e. the coordination should be accomplished in such a way that it only affects the ordering in which agents are allowed to execute their plan. That is, ideally, such a global plan \( p_A \) should consist of a simple ordering of the plans \( p_a \) of the agents instead of ordering the primitive tasks of the agents. Note that such cooperation protocols would not assume any knowledge about the details of the local plans made by the agents; it should be sufficient to have knowledge about the (task) dependencies between the agents. To represent such agent dependencies explicitly, we therefore define the agent dependency graph \( D_A \) w.r.t. the task \((T, \prec)\):

**Definition 1** Let \((T, \prec)\) be a task and \((T_a, \prec_a)\) be the subtasks assigned to the agents \( a \in A \). The agent dependency graph \( D_A \) is the graph \( D_A = (A, \prec_A) \), where \((a \prec_A b)\) iff there exists primitive tasks \( t \in T_a \) and \( t' \in T_b \) such that \( t < t' \).

We say that a global plan \( p_A \) feasible for \((T, \prec)\) can be constructed by simple plan coordination from the plans \( p_a \) feasible for \((T_a, \prec_a)\) of the agents \( a \in A \) if \( p_A \) is feasible for \((T, \prec')\) with:

1. \( \prec' = \bigcup_a \prec_a \cup \{(t, t') | t \in T_a, t' \in T_b \text{ and } a \prec_A b \} \)

2. \( \prec' \) is a partial order.

That is, \( p_A \) is the result of simply ordering the plans of the agents. It is easy to see that \( p_A \) exists iff \( D_A \) is a directed acyclic graph (DAG). Note that whenever simple plan
every agent is assumed to be able to produce a minimal plan. Hence, an optimal plan \( p_A \) feasible for \((T, \prec)\) can be obtained from optimal local plans \( p_a = (T_a, \prec_a) \) by simple plan coordination iff \( D_A \) is acyclic. In most cases, however, global plans cannot be obtained by simple plan coordination the reason being that, in general, an agent \( a \) will come up with a local plan extending \( \prec_a \). It may then be the case that, no matter how these local plans are ordered, the global precedence constraints are violated.

Figure 4 shows the dependency graph for the tasks \( T_1 \) and \( T_2 \). Note that this dependency graph consists of a cycle \{train, truck\} which means there is a conflict about whether train should wait for truck or vice versa.

In the following section we will discuss two algorithms to enable agents to come up with a global plan respecting all constraints while affecting their original local plans as little as possible. The price to be paid however is a loss of plan quality. We present some upper bounds on this loss of plan quality.
4 Plan coordination by partitioning local plans

By the previous example, it will not always suffice to add constraints to the local plans of the agents in order to guarantee a successful simple coordination process. Hence, in some cases it seems to be inevitable to apply a partitioning strategy: in order to get rid of circular dependencies between the (plans of) the agents, local plans have to be partitioned in separate subplans such that circular dependencies are removed. The coordination process then should only pertain the ordering of subplans between and within agents. In principle, since the original task is a partially ordered set, such a strategy should succeed: take a partitioning in which every agent has to execute a set of subplans each containing exactly one elementary task. In that case every global plan extending $\prec$ is a simple coordinated plan. But, clearly, such a solution could increase the costs of such a plan to its maximal value: since every agent has to execute every elementary task sequentially, the total cost of such plan $p$ could be $|T|$-times the optimal cost of a coordinated plan, because the tasks could possibly be performed concurrently. Therefore, in this section we will present two algorithms that guarantee a better performance: the first algorithm obtains a plan whose costs are never more than $d$ times the cost of an optimal plan where $d$ is the depth of the task order and the second algorithm realizes an even better ratio, but assumes more knowledge from the participating agents.

4.1 Partitioning using only local dependency information

We construct a polynomial-time distributed algorithm capable of adding sufficient constraints between the tasks an agent has to perform such that simple plan coordination becomes possible. The intuitive idea behind the algorithm is that each agent (only knowing from which other tasks its own tasks are dependent) tries to schedule a task $t_a$ whenever all tasks preceding it are already scheduled by the other agents. It then separates those tasks from the remaining tasks, adds them to a common store $\text{done}$ and splits its local task into two subtasks: one containing all tasks that can be scheduled now and the set of remaining tasks to be scheduled. As a result, the agent dependency graph will be refined to a new graph where instead of one node per agent several nodes per agent are created that are linearly ordered. It can be easily shown that the resulting refined agent dependency graph is acyclic, offering an easy way to construct a plan by simple plan coordination.

The algorithm uses a global store $\text{done}$ that is distributed among all the agents. Initially this store is empty. Each agent is capable of inspecting and updating the store $(\text{update}(\text{done}, \phi)$ means that $\phi$ is added to the store). Each agent only knows that in performing some tasks it is dependent upon other agents whose identity may be unknown. We specify for each agent $a$ a procedure to find out how to split its current task assignment $T_a$ into a number $k_a$ of linearly ordered task assignments $(T_{a,1}, \ldots, T_{a,k_a})$ such that simple plan coordination on the resulting set of subtasks becomes possible:

Algorithm 4.1
(partitioning for agent $a$)
begin
1. $k_a := 1 \quad \varepsilon_a := \emptyset$
2. while $T_a \neq \emptyset$ do
   2.1. $T_{a,k_a} := \{ t \in T_a \mid \forall t' \not\in T_a \mid t' < t \implies t' \in \text{done} \}$
   2.2. while $T_{a,k_a} = \emptyset$ do
       2.2.1. skip; % wait for tasks to occur in $T_{a,k_a}$
       2.2.2. $T_{a,k_a} := \{ t \in T_a \mid \forall t' \not\in T_i \mid t' < t \implies t' \in \text{done} \}$
   2.3. $T_a := T_a \setminus T_{a,k_a}$;
   2.4. update(done, $T_{a,k_a}$);
   2.5. $\varepsilon_a := \varepsilon_a \cup T_{a,k_a-1} \times T_{a,k_a}$ if $k_a > 1$;
   2.6. $k_a := k_a + 1$
end

As a result of this algorithm, a single agent plan $p$ feasible for $(T_a, \varepsilon_a \cup \varepsilon_a)$ is refined to a linear sequence of subplans $p_{a,1}, \ldots, p_{a,k_a}$. Let us associate each such a plan $p_{a,i}$ with a subagent $a_i$ of $a$. Now construct the refined dependency graph $D'_{\mathcal{A}} = (\bigcup_{a \in A} \{ a_1, \ldots, a_{k_a} \}, \prec'_{\mathcal{A}})$, where $a_i \prec'_{\mathcal{A}} b_j$ iff $\exists t \in T_{a,i}, t' \in T_{b,j}$ such that $t \prec t'$. Observing that as soon as subplan $T_{a,k_a}$ is created, all tasks that tasks in $T_{a,k_a}$ depend upon occur in done it is easy to show that the refined agent dependency graph $D'_{\mathcal{A}}$ is an acyclic graph.

Hence, by a simple version of e.g. distributed topological sort, a global plan $p$ found by simple coordination of all subplans can be easily obtained. The price of course to be paid is loss of plan efficiency. In fact, it is not difficult to prove that the efficiency of the plan obtained is no more than $d$-times the efficiency of an optimal plan, where $d$ is the depth of the task involved.

### 4.2 A refined partitioning strategy

A problem with the previous algorithm is that sometimes an agent puts its tasks on the global store too soon. If an agent waits until other agents have put a sufficient amount of tasks on the global store, more prerequisites will be satisfied allowing the agent to put more tasks on the global store, simultaneously, and thereby, reducing the number of plan splits. On the other hand, waiting too long may result in a deadlock situation in which none of the agents puts tasks on the global store. A better approximation of an optimal global plan therefore might be obtained if the agents also have knowledge about direct inter-agent dependencies, i.e., each agent $a$ exactly knows upon which tasks $t'$ assigned to other agents $b$ his own tasks $t$ are dependent. For each task $t$, the direct inter-agent dependencies for an agent consists of the following components: (i) the direct prerequisites $t'$ for each task $t$, and (ii) the agents assigned to these prerequisites. Using these information components, like in the previous algorithm, each agent $a$, iteratively selects a subset of its tasks $T_a$ and puts it on the global store done while respecting the constraint that $T_{a,i}$ may only be put there if all its prerequisites are already there. The following algorithm uses this dependency information and offers a suitable trade-off between efficiency and deadlock prevention.
Algorithm 4.2
(refined partitioning for agent $a$)

**Input:** a global task $(T, \prec)$ and the assignments of tasks $T_a$ to agents

**Output:** a subgoal $(T_a, \prec_a \cup \prec_a)$

begin

1. $k_a := 1; \prec_a := \emptyset$

2. while $T_a \neq \emptyset$ do

   2.1. $R_a := \emptyset; \; \text{todo} := \emptyset; \; \lambda(t) := \emptyset$ for all $t \in T_a$

   2.2. for $t \in T_a, b \in \mathcal{A}$ do

      2.2.1. let $S_{t,b}$ be the set of those direct prerequisites of task $t$ that have
              been assigned to agent $b$

      2.2.2. for every $S_{t,b} \neq \emptyset$, send request $(a, b, S_{t,b})$ to agent $b$

   2.3. for each request $(b, a, S_{t,a})$ received from agent $b$ do

      2.3.1. $R_a := R_a \cup \{ (t, S_{t,a}) \}$

      2.3.2. for each primitive task $t'$ in $S_{t,a}$ without prerequisites, send an
              inform $(a, b, t'; id)$ with $id$ a unique identifier to agent $b$

   2.4. for each inform $(b, a, t; id)$ received from agent $b$ do

      2.4.1. if $\exists t' \in T_a \mid id \in \lambda(t)$ (i and $t'$ belong to the same chain) then
              put this task $t'$ in the set todo;

      2.4.2. else

              $\lambda(t) := \lambda(t) \cup \{ I \}$

              for $(t', S) \in R_a$ s.t. $t \in S$ do

              send inform $(a, c, t'; I)$ to agent $c$

   2.5. $T_{a,k_a} := \{ t \in T_a \mid \forall t' \not\in T_a \mid t' \prec t \Rightarrow t' \in \text{done} \}$

   2.6. if todo $\subseteq T_{a,k_a}$ then

      2.6.1. $T_a := T_a \setminus T_{a,k_a}$

      2.6.2. update (done, $T_{a,k_a}$)

      2.6.3. $\prec_a := \prec_a \cup T_{a,k_a-1} \times T_{a,k_a}$ if $k_a > 1$

      2.6.4. $k_a := k_a + 1$

end

In Steps 2.2 and 2.3, the agents are informed which other agents are dependent on their tasks. In Step 2.3.2, the minimal elements of the task chains are determined and each minimal element is given a unique identifier that is used to identify the chain of primitive tasks. In Step 2.4, the identifiers are propagated along the task chains to determine which tasks in $T_a$ belong to the same chain. Based on this information, a set of primitive tasks (todo) is determined and the agent waits (Step 2.6) until all prerequisites of todo are in done.

We can show that the approximation ratio of this algorithm, in general, is much better than the previous algorithm: Algorithm 4.2 is a $\max\{2, k\}$-approximation algorithm to compute a global plan $P$ for a task $(T, \prec)$, where $k$ is the maximum over all agents.
geographically located which means that they occur at most once in a logistic chain, resulting in a 2-approximate algorithm.
5 Conclusions

We have followed an analytic approach to design coordination protocols and algorithms that can be used to coordinate multi-agent conduct into nearly optimal joint multi-agent plans. We have shown that the application of protocols that are based on allocation of primitive tasks without accounting for the precedence constraints between tasks to multimodal logistics has limitations and that better results can be obtained if inter-task dependencies are coordinated. Both theoretical analysis Valk and Witteveen (2002) and empirical evidence Valk et al. (2001) show that our coordination algorithms for resolving inter-task dependencies perform well on multimodal transportation problems. In particular, in multimodal transportation systems, agents are usually situated in a specific geographic region and, therefore, occupy one consecutive part of the logistic chains. In such a situation, the worst case performance ratio of our refined algorithm equals two.

Future research will, first of all, focus on following a similar analytic approach to deal with different kinds of time constraints and optimization criteria such as time windows in combination with precedence constraints and time optimization. Since the efficiency of coalition formation and task allocation methods depends on the way tasks are scheduled, the combination of methods for coalition formation and task allocation with plan coordination methods also needs attention.

Furthermore, we will pay attention to a more precise modeling of the utilities agents participating in a multimodal transportation setting. For example, in order to specify the quality of a multi-agent plan in a more detailed way, we will focus on special utility models specifying the utility of an agent as the reward received for the accomplished tasks minus the costs of the plans executed by the agents. These utility models then can be used to specify the total utility of the multi-agent plan as a function of an agent cooperation strategy.

Acknowledgements

This research has been supported by the TNO-TRAIL project IT-architecture and coordination in transport chains carried out within the research school for Transport, Infrastructure and Logistics (TRAIL).


