Approximation Results for Multi-Agent Planning Systems

Jeroen Valk[1], Cees Witteveen[1], and Jonne Zutt[1]

Delft University of Technology,
Dept. of Information Technology and Systems,
P.O. Box 356, 2600 AJ Delft, The Netherlands,
{J.M.Valk,C.Witteveen,J.Zutt}@its.tudelft.nl

Abstract. We study task allocation policies for multi-agent planning systems. In particular, we are interested in policies that can be used to establish lowerbounds on the performance of distributed planning systems if coordination between the subsystems would fail. Taking a simple multimodal logistic problem, we discuss a task allocation policy that decomposes the problem into a number of nearly independent local planning problems such that *(i)* each of these subproblems can be solved by the agents independently and *(ii)* the resulting plans can be easily assembled into a solution for the complete problem, without requiring almost any coordination between the agents. We present an analysis of the performance ratio of a *distributive planner* based on this task allocation policy. Next, these results are used to provide upperbounds for coordination strategies that can be employed by the agents to improve the planning.
Finally, we show that this approach can also be used to obtain a highly competitive logistic planner. We compare a planner developed on the basis of this task allocation policy with several well-known planners using the AIPS-2000 logistic benchmark set and we show that it outperforms all present planners both in speed and in plan quality.

1 Introduction and Motivation

In multi-agent approaches to distributive planning and problem solving, a *task allocation* process and a *task coordination* process can be distinguished (cf. [Kra97,WJ99]. In a task allocation process, agents are allocated to parts of the problem and required to solve them, while a task coordination process has to guarantee that the distributed problem solving activities are geared toward a solution of the complete problem. Clearly, these two processes are tightly interwoven: in order to guarantee a satisfactory solution, a poor task allocation policy will need an elaborate coordination process to achieve an acceptable solution and vice-versa. Our general research focus is on the development of *efficient* and *nearly optimal* task allocation policies, as well as efficient coordination strategies, to solve complex *logistic* problems. The research approach we will act upon is first to concentrate on task allocation policies and problem solving strategies based on them. We consider them as approximation algorithms used to allocate the tasks in a complex multi-agent setting to different agent in such a way that *(i)* the agents are able to solve their task almost independently; *(ii)* the resulting parts of the plan can be easily combined in a solution to the complete problem without requiring elaborate coordination and *(iii)* the resulting assembled solution is nearly optimal.

There are several reasons to concentrate on task allocation policies first. First of all, we would like to find *lowerbounds* on the performance of multi-agent approaches to complex problems if, on the one hand, coordination is expected to be needed to optimize the solutions found, but on the other hand, there is a non-negligible risk that such coordination processes might fail. In such cases we would like to use task allocation strategies that guarantee a lowerbound on the quality of the solution, even if the agents fail to coordinate. Secondly, the performance of coordination methods is notoriously hard to analyse. It is our idea that almost every logistic problem solution strategy can be reduced to a task allocation strategy together with negotiation and coordination routines. In this setting, the lowerbound obtained by analysing the worst-case performance of the task allocation core can also be used to prove lowerbounds on the performance of the complete solution strategy.

We apply this approach on multimodal logistic problems, i.e., problems where goods or persons have to be transported from one place to another using different forms of transportation facilities which are restricted to travel within a subset of places using connections between places. Although we plan to study these problems in full generality, for the moment we use some simplifying assumptions to get a clearer view on the basic task allocation policies that might be used in these problems[1].

The main results obtained in this paper can be summarized as follows: *(i)* We show that there exist surprisingly simple task allocation processes that can be used to solve complex logistic problems in a nearly optimal way, such that any feasible coordination process can hardly improve the resulting solution; *(ii)* the logistic problem we have chosen to study is a simple multimodal transportation problem that has been used as a benchmark problem for a variety of planners. We have developed a distributed approximation algorithm that is the result of a task allocation policy with the above mentioned properties. This task allocation policy has been implemented as a logistic problem solver for these benchmark problems that is highly competitive with other planners used to solve these logistic problems.

The paper is organised as follows. First, we will discuss the logistic benchmark problem, then we present a simple task allocation policy and use it in a multi-agent approach to solve the problem. We analyse the resulting plan quality if this policy is used by computationally super efficient agents. Then we show that the logistic problems each agent has to solve are intractable, but that a

[1] For example, we will not deal with time constraints and in most cases assume that capacity problems can be avoided. In the near future these properties will be added to the problems studied.

reasonable approximation algorithm might be used to produce nearly optimal local plans. Using the local approximation algorithms, we prove that there exists an approximation algorithm for the multimodal logistic problem that is able to solve the complete problem with an approximation ratio of only 1.25 (in totally uniform cost models) or 2 (in transportation uniform models) and that it is very unlikely that we are able to find approximation algorithms with a performance ratio less than 1.20. We discuss a simple negotiation scheme for multi-agents to coordinate their actions on solving multimodal logistic problems and we show that the task allocation policy discussed before can be used to analyse its performance. Finally, we show that an implementation of this policy can be used to solve all logistic benchmark problems of the AIPS-2000 planning competition.

2 A Multimodal Planning Problem

The multimodal logistic problem we will discuss consists of a set L of locations $l_{i,j}$ where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots n$. A city is a subset $C_i = \{l_{i,j} \mid j = 1, \ldots, n\}$ of locations, where each location can be visited by a truck. In each city C_i we distinguish a special location $l_{i,1}$ as the airport of city C_i. All airports are connected by directed flights and are visited by airplanes. There is also a set of packages and for each package an order consisting of a pick-up location $l \in L$ and a (different) delivery location $l' \in L$. The multiset of orders is denoted by O. Given an instance (L, O) of such a multimodal logistic problem, we are looking for a *plan* for executing the orders in O. Such a plan p is a sequence of load/unload and move actions completing all the orders in O. To determine the cost $c(p)$ of a plan p we will use a simple *transportation uniform cost model* where (i) the costs of a truck move action is $c^t_m \in \mathbb{N}$ (the same for all trucks), and the costs of plane move actions are given by $c^p_m \in \mathbb{N}$, (ii) the costs of every truck an plane load/unload action are $c^t_l \in \mathbb{N}$ and $c^p_l \in \mathbb{N}$, respectively. If all transportation costs are the same, we call the cost model a *totally uniform cost model*. The cost $c(p)$ then equals the sum of the costs of all actions occurring in p. Of course, we would like to obtain an *optimal plan* $p^\star$, i.e., a plan with minimum cost[2]. In particular, we would like to see the trucks and planes as separate *agents* solving the planning problem by using their own transportation facilities.

3 A Task Allocation Policy for the Multimodal Planning Problem

The multimodal planning problem sketched above is easily seen to be intractable[3] (see e.g. Section 4) and it can be easily seen that in minimum plans, the transportation agents are strongly dependent on each other in order to minimize the

[2] Note that we are looking here for a plan of minimum cost and not a plan of minimum time. Such a property is also of interest if actions can be performed concurrently.

[3] The problem is known to be in $P^{NP[\log n]}$.

number of move actions. We will use a simple task allocation policy to solve the multimodal planning problem. This policy consists of the following steps:

1. decompose the total set of orders O into a set of *local-transportation* orders, *pre-transportation* orders, *plane-transportation* orders and *post-transportation* orders[4];
2. combine the local-transportation orders and the pre-transportation orders and then split up by city;
3. let the local truck agents come up with their (local) transportation plan to complete the local- and pre-transportation orders;
4. let the planes make a plan for the plane orders;
5. let the truck agents independently make plans for the post-transportation orders;
6. assemble the local and plane plans into a complete plan.

We call this policy the *arbiter-0 policy*. We denote the multiset of local-transportation and pre-transportation orders by O_{pre}, the multiset of plane-transportation orders by O_{plane}, and the multiset of post-transportation orders by O_{post}. It is not difficult to see that the transportation of all orders in O is performed if first the orders in O_{pre} are executed, then the orders in O_{plane} are executed, and finally the orders in O_{post} are executed.

Given an instance I of the multimodal logistic problem, we will write $p_I^\star$ and p_I^0 to denote an optimal plan and a plan found by the arbiter-0 policy for I, respectively. For convenience, if the problem instance is clear from the context, we will omit the subscript and write $p^\star$ and p^0. According to the decomposition of the orders O into O_{pre}, O_{plane} and O_{post}, *any* feasible plan p can also be partitioned into three *partial plans*: (i) p_{pre} containing the actions concerning the execution of orders in O_{pre}, (ii) p_{plane} containing the actions concerning the execution of orders in O_{plane}, and (iii) p_{post} containing the actions concerning the execution of orders in O_{post}.

The performance ratio ε of the arbiter-0 policy is defined to be the supremum over all problem instances I of the cost $c(p_I^0)$ devided by the cost $c(P_I^\star)$. Next, we will derive the performance ratio when optimal local solvers are used. Section 4 presents polynomial-time algorithms for the local planning problem. When these algorithms are used together with the arbiter-0 policy, the result is a polynomial-time algorithm for the multimodal planning problem. The performance ratio of this algorithm is discussed in Section 5.

For a (partial) plan p we write $LU(p)$ to denote the amount of *load and unload* actions in p and $MV(p)$ to denote the amount of *move* actions in p.

Proposition 1. *Let $p^\star$ be an optimal plan and let p be a plan found by an arbitrary scheduling policy. Under the assumption that the local problem solvers are efficient for loading and unloading, we have*

$$LU(p^\star) = LU(p) = 2 \cdot (|O_{\mathsf{pre}} \cup O_{\mathsf{plane}} \cup O_{\mathsf{post}}|).$$

[4] Note that every inter-city order can be decomposed into 2 (possibly empty) intra-city orders (a pre-transportation order and a post-transportation order) and a plane order.

Proof Any single order $o \in O_{\mathsf{pre}} \cup O_{\mathsf{plane}} \cup O_{\mathsf{post}}$ can be performed by a single transportation unit. Hence, a local problem solver that is efficient for loading and unloading only needs one load and one unload action. The same holds for any subset $O \subseteq O_{\mathsf{pre}} \cup O_{\mathsf{plane}} \cup O_{\mathsf{post}}$, because transportation units are assumed to have unlimited capacity. ■

Proposition 2. *Under the assumption that the local transport problems are solved optimally by the local planners in the arbiter-0 policy, the overhead is bounded by* $c(p^0) - c(p^\star) \leq c^t_m \cdot \min\{|L_{\mathsf{pre}}|, |L_{\mathsf{post}}|\}$*, where* L_{pre} *and* L_{post} *denote the set of locations occurring as source or destination in* O_{pre} *and* O_{post}*, respectively.*

Proof The first two transportation phases are solved optimally by the arbiter-0 policy, assuming that the local problem solvers are exact solvers. Hence, $MV(p^0_{\mathsf{pre}} \cup p^0_{\mathsf{plane}}) = MV(p^\star_{\mathsf{pre}} \cup p^\star_{\mathsf{plane}})$. However, post-transport moves in $p^\star$ might be combined with pre-transport moves if orders in O_{post} do overlap with orders in O_{pre}. More exactly, suppose that L_{pre} is the set of locations occurring in O_{pre} and L_{post} is the set of destinations to be visited in O_{post}, then for every order $(l, l') \in O_{\mathsf{post}}$ such that $l, l' \in L_{\mathsf{pre}}$, the corresponding move action could possibly be combined with a move action carried out in the pre-transportation phase. Clearly, the total set of moves that could be combined is bounded above by $MV(p^0_{\mathsf{post}}) - MV(p^\star_{\mathsf{post}}) \leq |L_{\mathsf{pre}} \cap L_{\mathsf{post}}| \leq \min\{|L_{\mathsf{pre}}|, |L_{\mathsf{post}}|\}$. Combining the results we obtain: $c(p^0) - c(p^\star) = c^t_m \cdot (MV(p^0_{\mathsf{post}}) - MV(p^\star_{\mathsf{post}})) \leq c^t_m \cdot min\{|L_{\mathsf{pre}}|, |L_{\mathsf{post}}|\}$. ■

Let $n = |L_{\mathsf{pre}}|$ and $m = |L_{\mathsf{post}}| > 0$. The following observations can be used to establish a lower bound on the cost of an optimal plan.

1. $LU(p^0_{\mathsf{pre}}) \geq n$ and $MV(p^0_{\mathsf{pre}}) \geq (n-1)$,
   since every location occurring in L_{pre} requires either a load or an unload action and at least $n-1$ move actions are required in the pre-transportation phase,
2. $LU(p^0_{plane}) \geq 2m$ and $MV(p^0_{\mathsf{plane}}) \geq 1$,
   since every post transportation order has to be preceded by a plane order and has to be loaded and unloaded by a plane,
3. $LU(p^0_{\mathsf{post}}) \geq 2m$ and $MV(p^0_{\mathsf{post}}) = m$,
   since every final location to be visited requires at least 2 load/unload actions and requires one move action.

Now $c(p^\star)$ can be computed as follows:

$$\begin{aligned} c(p^\star) &= c^t_l \cdot (LU(p^0_{\mathsf{pre}}) + LU(p^0_{\mathsf{post}})) + c^t_m \cdot (MV(p^0_{\mathsf{pre}}) + MV(p^\star_{\mathsf{post}})) \\ &\quad + c^p_l \cdot LU(p^0_{\mathsf{plane}}) + c^p_m \cdot MV(p^0_{\mathsf{plane}}) \\ &\geq c^t_l(2m + n) + c^t_m \cdot (n-1) + 2c^p_l m + c^p_m \end{aligned}$$

Hence, we obtain the following approximation ratio:

$$\begin{aligned}\frac{c(p^0)}{c(p^\star)} &\leq 1 + \frac{c^t_m \cdot min\{n,m\}}{(2m+n)c^t_l + (n-1)c^t_m + 2mc^p_l + c^p_m} \\ &\leq 1 + \frac{n \cdot c^t_m}{3n \cdot c^t_l + (n-1)c^t_m + 2n \cdot c^p_l + c^p_m}.\end{aligned}$$

The right-hand side is maximized if $n = m$. Thus, $\epsilon \leq 1 + (1 + (\frac{3c^t_l + c^p_l}{c^t_m}))^{-1}$ To differentiate between several cases, let $c^p_l = a \times c^t_m$ and $c^t_l = b \times c^t_m$ for some $a, b \in R^+$. Then $\epsilon \leq 1 + (1 + 2a + 3b)^{-1}$ from which the following special cases can be deduced:

1. if $a, b \geq 1$, the approximation ratio is bounded above by $\epsilon \leq 1.167$; in particular, this holds in a strict uniform cost model
2. if $a \geq 1$, i.e., the plane load/unload costs dominate the truck move costs,it follows that ϵ is bounded above by 1.34;
3. if $b \geq 1$, i.e., the truck load/unload costs dominate the truck move costs, we obtain an upperbound $\epsilon \leq 1.25$;
4. finally, if both a and b can take arbitrary values, the approximation ratio is bounded above by $\epsilon \leq 2$.

The following theorem summarizes the results obtained above:

Theorem 1. *Under the assumption that in the local transport problems are solved exactly, the global multimodal planning problem is approximable within* $\epsilon \leq 2$.

Remark 1. Note that in the above analysis only an upperbound on the performance ratio is given. To show that an upperbound ε is *tight*, we have to provide *tight examples*, i.e., problem instances I that reach the upperbound. For example, in case the costs are assumed to be strictly uniform, i.e., $c^t_l = c^t_m = c^p_l = c^p_m$, we have the following family as a tight example. There are two cities 1 and 2 with two locations $l_{1,1}$ and $l_{1,2}$ in city 1 and $n+1$ locations $l_{2,j}$, $j = 1, \ldots, n+1$ in city 2. In each city the location l_{i_1} denotes the airport. The trucks are located at the airport, the plane is located at $l_{1,1}$. Consider the following set of orders: $O = \{(l_{2,2j}, l_{2,2j+1}) \mid j = 1, \ldots, \frac{n}{2}\} \cup \{(l_{1,1}, l_{2,j}) \mid j = 2, \ldots, n+1\}$. A detailed analysis shows that for this class of examples we obtain $\frac{c(p^0)}{c(p^\star)} = 1 + \frac{nc^t_m}{(n-1)[2c^p_l + 3c^t_l + c^t_m] + c^p_m}$. Hence, for $n \to \infty$, ϵ approaches $\frac{7}{6}$.

4 Solving the Local Planning Problems

Note that Theorem 1 is the result if the planners for the local problems are assumed to be optimal. However, it turns out that some of the local planning problems themselves are NP-hard. Therefore, we will first analyse these local planning problems and thereafter present a polynomial approximation algorithm for the arbiter-0 policy.

A local planning instance is a pair (L, O), where L is a set of locations and $O \subseteq L \times L$ is a set of pickup-delivery orders. We say that a sequence $s \in L^*$ is a *visiting sequence* for O if, for every order (l_1, l_2), s contains a subsequence starting with l_1 and ending with l_2. We are interested in finding a *minimum* visiting sequence, i.e., a visiting sequence with smallest length. It is not difficult to show that this problem is NP-hard, because the *Minimum Directed Feedback Vertex Set*[5] problem (MDFVS) reduces to it.

Section 4.1 presents a simple approximation algorithm for determining a minimum visiting sequence and in Section 4.2 we also account for the cost of loading and unloading.

4.1 Approximating Optimal Move Costs in Local Planning Problems

The approximation algorithm we have in mind invokes an efficient polynomial-time algorithm for solving the MDFVS problem.

Algorithm 4.1

Input: *a local planning instance*[6] (L, O)*;*
Output: *a visiting sequence* $s \in L^*$*;*

begin

1. *compute a feedback vertex set* F *of the graph* (L, O) *using a polynomial-time algorithm;*
2. *compute a topological sort* s_1 *of* $(L, \{(l_1, l_2) \in O \mid l_2 \notin F\})$*;*
3. *let* s_2 *be an arbitrary permutation of the locations in* F*;*
4. *let* s *be the concatenation of* s_1 *and* s_2*;*

end

Proposition 3. *Algorithm 4.1 is a 2-approximation algorithm for finding a minimum visiting sequence.*

Proof See Proposition 4 below. ∎

Remark 2. Although it might seem that Algorithm 4.1 could be improved significantly, actually, we have little hope that there exist polynomial algorithms with a better approximation threshold. The reason being that if a polynomial-time algorithm exist with an approximation ratio strictly less than 2, this would immediately imply that the MDFVS problem is in APX [VWZ01]. It is well known that the minimum feedback vertex set problem is APX-hard and until now, the best known approximation algorithm guarantees an $\mathcal{O}(\log n \log \log n)$-factor approximation ([Sey95,ENSS95,FPR99]). However, to the best of our knowledge,

[5] The minimal directed feedback vertex set problem is defined as follows: given a directed graph $G = (V, A)$ find a minimum subset V' of V such that every directed cycle of G has at least one node in common with V'. The problem is known to be NP-hard

[6] Without loss of generality we assume that (L, O) contains no isolated nodes.

it is still an open question whether MDFVS is in APX or not. Therefore, we conclude that a 2-approximation algorithm for the optimal move cost in local planning problems as given, currently is the best we can hope for.

4.2 The Approximation Algorithm and Local Transportation Planning Problems

The total costs of a plan p solving a local planning problem (pre-order or plane) is related to the length of a visiting sequence s for the move cost planning problem, we discussed in the preceding paragraph. If (L, O) is an instance of the local planning problem and the visiting sequence s a solution, the cost of the corresponding transportation plan p is $c(p) = 2|O| \cdot c_l + |s|$. To find an upperbound on the performance ratio for this cost function, we have to analyse the algorithm that is applied for solving the underlying MDFVS-problem. Given a directed graph G, any efficient algorithm for solving MDFVS finds a feedback vertex set F such that the number of edges in G is twice the number of nodes in F. Otherwise, a simple polytime algorithm can be used to improve upon F.

Lemma 1. *Let (V, E) be a directed graph. There exists a polynomial time algorithm for computing a feedback vertex set F with the property that $|E| \geq 2 \cdot |F|$.*

Proof Consider the following algorithm: (i) select a node $v \in V$ that is in a cycle, (ii) remove v from V and all the edges incident to v from E, (iii) repeat steps (i) and (ii) until the graph is acyclic. The feedback vertex set F consists of those nodes that have been selected. Since each selected node is in a cycle, at least two edges are removed at each iteration. Hence, $|E| \geq 2 \cdot |F|$. ■

Proposition 4. *The local planning problem is approximable within $1 + \frac{c_m}{4c_l + c_m}$.*

Proof Let (L, O) be a local planning instance and suppose that a good approximation algorithm for MDFVS gives a feedback vertex set F. If F is used to construct a solution p then the cost of this solution is $c(p) = 2 \cdot c_l \cdot |O| + c_m \cdot (|L| + |F|)$. Since the cost of an optimal solution is bounded by $c(p^\star) \geq 2 \cdot c_l \cdot |O| + c_m \cdot |L|$ and $|L| \geq |F|$, we have:

$$\frac{c(p)}{c(p^\star)} \leq 1 + \frac{c_m \cdot |F|}{2 \cdot c_l \cdot |O| + c_m \cdot |L|} \leq 1 + \frac{c_m \cdot |F|}{2 \cdot c_l \cdot |O| + c_m \cdot |F|}.$$

Applying $|O| \geq 2 \cdot |F|$, which follows from Lemma 1, we obtain:

$$\frac{c(p)}{c(p^\star)} \leq 1 + \frac{c_m \cdot |F|}{4 \cdot c_l \cdot |F| + c_m \cdot |F|} \leq 1 + \frac{c_m}{4 \cdot c_l + c_m}.$$

■

Differentiating between several cases we have:

1. a performance ratio of 1.2 in a totally uniform cost model (i.e., $c_l = c_m$), and
2. a performance ratio of 2 when the cost of loading and unloading can be neglected (i.e., $c_l < c_m$).

5 An Approximation Algorithm for the Multimodal Planning Problem

In this section we combine the results obtained in the previous sections by showing how a polynomial approximation algorithm for a multimodal planning problem can be synthesized using the approximation algorithms for the task allocation policy and the local subproblems. In fact such an approximation algorithm can be viewed as implementing a special planner for multimodal logistic planning problems.

In Section 4, we have shown that a polynomial approximation algorithm exists that solves the local planning problems. It is easy to see that if we combine such an approximation algorithm with the algorithm based on the arbiter-0 policy, a polynomial planning algorithm for the complete multimodal planning problem is obtained. We will call this algorithm the *arbiter-0 algorithm*. In this section we will determine worst-case performance ratios of the arbiter-0 algorithm.

Analysing the local problems into more detail, it is easily seen that only the pre-transportation and plane-transportation problems need the local approximation algorithm, because the post-transportation problems can be solved exactly. This is easy to see, since during post-transportation packages that has been brought to the airport by an airplane only have to be distributed among a number of locations within the city. Hence, the corresponding order-graph for these local problem is acyclic, and a shortest route can be found in polynomial time. Accordingly, there are two reasons why the quality of a plan found by the arbiter-0 algorithm is worse than the plan quality of an optimal plan:

1. the same as for the arbiter-0 policy, an overhead is caused because in the post-transportation phase move actions can be saved by combining this phase with the pre-transportation phase, and
2. in the pre-transportation phase and the plane-transportation phase an overhead is caused by the use of the local approximation algorithm; an overhead that does not occur when using optimal local problem solvers.

Combining the two reasons gives the following performance ratio; due to lack of space the proof is omitted.

Theorem 2. *The arbiter-0 algorithm approximates the optimal solution within*

$$\frac{c(p)}{c(p^\star)} \leq 1 + \frac{2 \cdot c_m^t + c_m^p}{6c_l^t + 4c_l^p + c_m^t + c_m^p}.$$

In the totally uniform cost model, we obtain $\epsilon = 1.25$. Note that in this cost model, the overhead due to the policy is approximately 5% as the performance ratio for solving the local planning problems is 1.2. If the plane move costs dominate the other costs completely, the approximation ratio will converge to 2 as expected, since in that case the overhead due to approximating optimal plane moves will dominate the plan costs.

Example 1. We present a tight example. Consider the sequence of problem instance for increasing n with $n + 1$ cities, and $n + 1$ locations in city 1 and one location in the remaining n cities. Assume that the set $\{(l_{i,1}, l_{1,i} \mid i = 2, \ldots, n{+}1\}$ is a subset of the set of orders O. Further, assume that both the local planning problem within city 1 and the local planning problem for the planes are worst approximable. This can be obtained by adding the correct orders within city 1 and between airports. It is not difficult to see that the plan length of this sequence of problem instances converges to the ratio of Theorem 2 for $n \to \infty$.

6 Performance Guarantees for Coordination Strategies

The simple policy we have discussed before can be easily used to derive some lowerbounds on the performance ratio of *coordination strategies* agents can use to improve their solution of complex multimodal problems. In this section we present a simple example of such a strategy.

Suppose that, initially, the agents have agreed upon using the arbiter-0 task allocation strategy. Note that when using this strategy, on the one hand, truck agents are committed to first executing the local- and pre-transportation phase before they are allowed to execute the post-transportation phase, whereas, on the other hand, the plane agents have no commitments to the truck agents at all. However, loosening the commitments for the trucks might result in a better plan, because local-transportation, pre-transportation and post-transportation might be combined. Each truck agent a could compute the performance gain of its own local plan as follows. First, a computes an efficient plan under the arbiter-0 policy, i.e., a plan that has a separate pre-transportation phase and post-transportation. Such a plan is denoted by p_a^0. Second, agent a computes an efficient plan under the assumption that all orders that require post-transportation by agent a are already delivered at the airport. Such a plan is denoted by p_a^1.

Now, if $\delta_a = c(p_a^0) - c(p_a^1) > 0$, agent a might try to negotiate with the plane agent to achieve that the plane, before taking any orders from the airport of agent a, will deliver all the post-orders to a. If such a negotiation will succeed, we say that the plane agent is *committed* to a. We will assume that once a commitment to an agent has been made it will not be broken during subsequent negotiations with other agents.

In determining whether previous commitments are in danger, the *dependencies* between agents play a central role. Given two distinct agents a and b, b is said to be *directly dependent* on a if there exists an order o such that *(i)* o needs pre-transportation by agent a and *(ii)* o also needs post-transportation by agent b. We say that b is said to be *dependent* on a if there exists a sequence $a_0, \ldots, a_n$ of agents with $a = a_0$ and $b = a_n$ such that for each $0 \leq i < n$, a_i is directly dependent on a_{i+1}. Two distinct agents a and b are said to be *mutually dependent* if a depends on b and b depends on a.

A *negotiation strategy* iteratively selects some agent with which the plane agent tries to negotiate. Initially, the plane agent has no commitments to the truck agents. A negotiation with agent a will succeed if

1. $\delta_a > 0$ and the possible additional costs the plane has to make in order to delay its visit to the airport of agent a are less than δ_a, and
2. the plane agent is not already committed to another truck agent b such that a and b are mutually dependent.

If both conditions are satisfied, we are guaranteed that the resulting plan costs can be reduced by a commitment to agent a.

Clearly, such negotiations can be performed iteratively, until no new commitments can be achieved. Since at each iteration plan quality can only improve, it immediately follows that the performance ratios of Theorem 1 and Theorem 2 are also upperbounds for the performance when the policy is combined with the above negotiation phase. In fact, for a particular set of instances we have the following result.

Proposition 5. *Suppose that the plane order graph* (L, O_{plane}) *is acyclic. Then, in the uniform cost model, any fair coordination strategy will return a plan* p *such that* $\frac{c(p)}{c(p^*)} \leq 1.2$*. Furthermore, under the assumption that the local problem solvers are optimal, any fair negotiation strategy will find an optimal plan.*

7 Experimental Results

The multimodal planning problem with a totally uniform cost model was used at the AIPS-2000 conference[7] as a benchmark set to test the performance of general purpose planning systems using STRIPS-style problem descriptions. We have implemented a polynomial 1.25-approximate algorithm based on the arbiter-0 policy and using a greedy algorithm for the local subproblems. Using our algorithm we have solved all 120 planning problems from the AIPS benchmark set ranging from 41 to 100 orders. The results, compared to the planning systems that participated in the planning competition, are summarized in Table 1. For each planning system that had a good performance at the planning competition[8] and for the problem instances in track 2 numbered 41-0 to 100-1, a total of 120 instances, we have determined the *minimum*, *average* and *maximum* of the plan cost found by the different planning systems divided by the plan cost found by our algorithm. Hence, a number > 1 means that our planner outperforms the planner compared with. Furthermore, we have compared the run-times of these planners with our planner, resulting in a speed-up ratio. Since the all plan cost ratios and all CPU time ratios are greater than 1, it can be concluded that our algorithm outperforms all planning systems at AIPS-2000 both in CPU time and in plan quality. Regarding SHOP and TALplanner, it seems that the plan cost ratio is independent of the number of orders. For System-R, however, we observed that the performance ratio becomes greater as the size of the problem instance increases. In running time, the TALplanner system, which seems to

[7] See http://www.cs.toronto.edu/aips2000/ for the results of the planning competition.

[8] These planners were System-R, SHOP and TALplanner

	System-R	SHOP	TALplanner
	plan cost ratio		
min	3.000	1.007	1.014
avg	4.397	1.028	1.057
max	5.958	1.052	1.085
	CPU time ratio		
min	878.50	305.25	4.6000
avg	4778.0	2056.0	9.4526
max	12307	4935.0	22.400

Table 1. The performance of the arbiter-0 algorithm compared with the best systems from the AIPS-2000 planning competition.

have a polynomial running time as well, turned out to be most competitive. System-R and SHOP require an exponential running time, and therefore it is expected that the speedup factor of our algorithm with respect to these systems is greater for larger problem instances.

References

[ENSS95] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multi-cuts in directed graphs. In *Proceedings of the 4th International Conference on Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science 920, pages 14–28, Berlin, 1995. Springer-Verlag.

[FPR99] Paola Festa, Panos M. Pardalos, and Mauricio G. C. Resende. Feedback set problems. Technical report, ATT TR 99.2.2, 1999.

[Kan92] V. Kann. *On the Approximability of NP-complete Optimization Problems*. PhD thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.

[Kra97] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2):79–98, 1997.

[Sey95] P.D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15:281–288, 1995.

[VWZ01] Jeroen Valk, Cees Witteveen, and Jonne Zutt. Task allocation policies for logistic problems. In *Proceedings of the First Seminar "Joint TNO TRAIL Transport Research T3"*, March 2001.

[WJ99] M. Wooldridge and N.R. Jennings. The cooperative problem solving process. *Journal of Logic and Computation*, 9(4), 1999.

Acknowledgements

J.M. Valk is supported by the TNO-TRAIL project *IT-architecture and coordination in transport chains*. J. Zutt is supported by the TNO-TRAIL project *Foutdetectie en recovery in multimodale verkeersnetwerken met autonome mobiele actoren* and by the *Freight Transport Automation and Multimodality (FTAM)* research program. Both programs are carried out within the TRAIL research school for Transport, Infrastructure and Logistics. The authors thank Jan Rogier (TNO-TPD) and Hans Tonino (TUD-ITS) for their comments on this paper.