On the Complexity of Planning in Transportation Domains

Malte Helmert

Institut für Informatik, Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee, Gebäude 052, 79110 Freiburg, Germany
helmert@informatik.uni-freiburg.de

Abstract. The efficiency of AI planning systems is usually evaluated empirically. The planning domains used in the competitions of the 1998 and 2000 AIPS conferences are of particular importance in this context. Many of these domains share a common theme of transporting *portables*, making use of *mobiles* traversing a map of *locations* and *roads*.
In this contribution, we embed these benchmarks into a well-structured hierarchy of *transportation problems* and study the computational complexity of optimal and non-optimal planning in this domain family. We identify the key features that make transportation tasks hard and try to shed some light on the recent success of planning systems based on heuristic local search, as observed in the AIPS 2000 competition.

1 Introduction

Apart from generally applicable hardness results [4], there is hardly any theoretical work on the time and space efficiency of common planning algorithms, so empirical methods have become the standard for performance evaluations in the planning community. Running time on problems from classical planning domains such as LOGISTICS and BLOCKSWORLD has often been (and still is) used for comparing the relative merits of planning systems. However, this kind of comparison is always difficult. If no planning system performs well in a given domain, does that mean that they are all poor, or is that domain intrinsically hard? If they all perform well, is this because of their strength or because of the simplicity of the task?

On a related issue, should planning systems be preferred that generate shorter plans but need more time? While there is no general answer to that question, theoretical results can contribute to the discussion, e. g. in cases where generating plans is easy but generating optimal plans is infeasible.

For addressing these issues, domain-specific complexity results for planning tasks appear to be useful. Pondering which domains to analyze, the ones that immediately spring to mind are the competition benchmarks from AIPS 1998 and AIPS 2000, considering their general importance for the planning community and the wealth of empirical performance data available.

While it would be possible to investigate each competition domain in isolation, it seems more worthwhile to identify commonly reoccurring concepts and prove more general results that apply to domain families rather than individual domains. Not only does this help present the results in a more structured way, it also allows to shed some light on the *sources of hardness* in these benchmarks.

Because of space limitations, we only discuss the *transportation* domain family here, covering eight of the thirteen competition domains, namely GRID, GRIPPER, LOGISTICS, MYSTERY, MYSTERY', and three versions of MICONIC-10. A similar discussion of the other domains (ASSEMBLY, BLOCKSWORLD, FREECELL, MOVIE, and SCHEDULE) and the corresponding domain families as well as a more thorough discussion of the results presented here can be found elsewhere [8].

In the following section, we will introduce and analyze some new transportation problems generalizing most of the competition benchmarks. Section 3 applies the results of this analysis to the competition domains and covers some additional aspects of the GRID and MICONIC-10 domains. The implications of those results are discussed in Section 4, followed by some comments on related work in Section 5 and possible directions for future research in Section 6.

2 A Hierarchy of Transportation Problems

In this section, we will define and analyze a hierarchy of transportation problems that combines the key features of the important transportation benchmark domains.

Definition 1. TRANSPORT *task*

A TRANSPORT **task** *is a 9-tuple* $(V, E, M, P, fuel_0, l_0, l_G, cap, road)$, *where*

- (V, E) *is the* **roadmap** *graph; its vertices are called* **locations**, *its edges are called* **roads**,
- M *is a finite set of* **mobiles**,
- P *is a finite set of* **portables** *(V, M, and P must be disjoint)*,
- $fuel_0 : V \to \mathbb{N}$ *is the* **fuel** *function,*
- $l_0 : (M \cup P) \to V$ *is the* **initial location** *function,*
- $l_G : P \to V$ *is the* **goal location** *function,*
- $cap : M \to \mathbb{N}$ *is the* **capacity** *function, and finally*
- $road : M \to \mathbb{P}(E)$ *is the* **movement constraints** *function.*

This should not require much explanation. The goal location function is only defined for portables because we do not care about the final locations of mobiles. We do require that goal locations are specified for *all* portables, unlike most planning domains. This is because portables with unspecified goals could safely be ignored, not contributing to the hardness of the task.

The fuel function bounds the number of times a given location can be left by a mobile. Fuel is associated with locations rather than mobiles because this is the way it is handled in the MYSTERY-like domains. The carrying capacity function bounds the number of portables a given mobile can carry at the same time. The movement constraints function specifies which roads a given mobile is allowed to use.

We will now define some special cases of transportation tasks.

Definition 2. *Special cases of* TRANSPORT *tasks*

For $i, j \in \{1, \infty, *\}$ *and* $k \in \{1, +, *\}$, $\mathcal{I}_{ijk}$ *is defined as the set of all* TRANSPORT *tasks* $I = (V, E, M, P, fuel_0, l_0, l_G, cap, road)$ *satisfying:*

- *For* $i = 1$, $cap(m) = 1$ *for all mobiles* m *(one mobile can carry one portable).*
- *For* $i = \infty$, $cap(m) = |P|$ *for all mobiles* m *(unlimited capacity).*
- *For* $j = 1$, $fuel_0(v) = 1$ *for all locations* v *(one fuel unit per location).*
- *For* $j = \infty$, $fuel_0(v) = \infty$[1] *for all locations* v *(unlimited fuel).*
- *For* $k = +$, $road(m) = E$ *for all mobiles* m *(no movement restrictions).*
- *For* $k = 1$, $road(m) = E$ *for all mobiles* m *and* $|M| = 1$ *(no movement restrictions, only one mobile).*

According to this definition, the most general task set, containing all TRANSPORT tasks, is $\mathcal{I}_{***}$, and the most specific ones, having no proper specializations in the hierarchy, are $\mathcal{I}_{111}$, $\mathcal{I}_{1\infty1}$, $\mathcal{I}_{\infty11}$, and $\mathcal{I}_{\infty\infty1}$.

Definition 3. TRANSPORT *state transition graph*

The **state transition graph** $\mathcal{T}(I)$ *of a* TRANSPORT *task* $I = (V, E, M, P, fuel_0, l_0, l_G, cap, road)$ *is the digraph* $(V_\mathcal{T}, A_\mathcal{T})$ *with* $V_\mathcal{T} = (M \cup P \to V \cup M) \times (V \to \{0, \ldots, \max fuel_0(V)\})$[2] *and* $((l, fuel), (l', fuel')) \in A_\mathcal{T}$ *if and only if:*

$$
\begin{aligned}
&(\exists m \in M, v, v' \in V: && l(m) = v \wedge \{v, v'\} \in road(m) \wedge fuel(v) > 0 \\
& && \wedge\ l' = l \oplus (m, v')^{3} \wedge fuel' = fuel \oplus (v, fuel(v) - 1)) \\
&\vee\ (\exists m \in M, p \in P: && l(m) = l(p) \wedge |\{\ p \in P \mid l(p) = m\ \}| < cap(m) \\
& && \wedge\ l' = l \oplus (p, m) \wedge fuel' = fuel) \\
&\vee\ (\exists m \in M, p \in P: && l(p) = m \wedge l' = l \oplus (p, l(m)) \wedge fuel' = fuel)
\end{aligned}
$$

This definition captures the intuition of legal state transitions in the specified transportation task. The first disjunct specifies transitions related to *movements* of a mobile, the second relates to a mobile *picking up* a portable, and the third to a mobile *dropping* a portable. In the following, we will only use these intuitive terms when talking about state transitions.

We can now define the decision problems we are interested in:

Definition 4. PLANEX-TRANSPORT$_{ijk}$
Given: TRANSPORT *task* $I = (V, E, M, P, fuel_0, l_0, l_G, cap, road) \in \mathcal{I}_{ijk}$.
Question: *In* $\mathcal{T}(I)$, *is there any directed path from* $(l_0, fuel_0)$ *to* $(l_G, fuel')$ *for some* $fuel' \in V \to \mathbb{N}$?

Definition 5. PLANLEN-TRANSPORT$_{ijk}$
Given: TRANSPORT *task* $I = (V, E, M, P, fuel_0, l_0, l_G, cap, road) \in \mathcal{I}_{ijk}$, $K \in \mathbb{N}$.
Question: *In* $\mathcal{T}(I)$, *is there a directed path of length at most* K *from* $(l_0, fuel_0)$ *to* $(l_G, fuel')$ *for some* $fuel' \in V \to \mathbb{N}$?

---

[1] Of course, ∞ is not a natural number. However, as we shall see shortly in the proof of Theorem 1, we can assume that there is "enough" fuel at each location, justifying this definition.

[2] States specify the location of mobiles and portables and the current fuel function.

[3] We use the notation $f \oplus (a', b')$ for *functional overloading*, i.e. the function f' with $f'(a') = b'$ and $f'(a) = f(a)$ for $a \neq a'$.

Theorem 1. PLANLEN-TRANSPORT$_{***}$ $\in$ **NP**

Proof. If we can show that any solvable TRANSPORT task I has a solution of length $p(||I||)$ for some fixed polynomial p, then a simple guess and check algorithm can solve the problem non-deterministically.

This is true because each portable only needs to be at each location at most once, bounding the number of pickup and drop actions, and in between two pickup or drop actions, no mobile should visit a given location twice. □

Corollary 1. PLANEX-TRANSPORT$_{ijk}$ $\leq_p$ PLANLEN-TRANSPORT$_{ijk}$ *for arbitrary values of* i, j, k *(and hence* PLANEX-TRANSPORT$_{***}$ $\in$ **NP***)*

Proof. A TRANSPORT task I has a solution if and only if it has a solution of length $p(||I||)$, for the polynomial p from the preceding theorem. Therefore the mapping of I to $(I, p(||I||))$ is a polynomial reduction. □

2.1 Plan Existence

Theorem 2. PLANEX-TRANSPORT$_{*\infty*}$ $\in$ **P**

Proof. Using breadth-first search on $road(m)$ starting at $l_0(m)$ for each mobile m with non-zero capacity, we can determine which roads can ever be used by any loaded mobile. The task can be solved if and only if for each portable p, $l_G(p)$ can be reached from $l_0(p)$ using these roads. This can easily be decided in polynomial time, and in fact the actual plans can easily be generated. □

This shows that the plan existence problems can be solved in polynomial time if no fuel constraints are present. We will now show that they are **NP**-complete otherwise, by proving **NP**-hardness of PLANEX-TRANSPORT$_{111}$ and PLANEX-TRANSPORT$_{\infty 11}$.

Theorem 3. PLANEX-TRANSPORT$_{111}$ *is* **NP***-complete*

Proof. Membership in **NP** is already known. We prove **NP**-hardness by a reduction from the **NP**-complete problem of finding a Hamiltonian path with a fixed start vertex [6, Problem GT39]. Let (V, E) be a graph and $v_1 \in V$. Then (V, E) contains a Hamiltonian path starting at v_1 if and only if there is a solution for the TRANSPORT task $I \in \mathcal{I}_{111}$ defined as follows: For each $v \in V$, there are two distinct locations v (called an *entrance*) and v^* (called an *exit*), with one unit of fuel each. At each entrance, there is a portable to be moved to the corresponding exit. There is only one mobile, of capacity one, starting at the entrance v_1. There are roads from v to v^* for $v \in V$ and from u^* to v for $\{u, v\} \in E$.

Now, if there is a Hamiltonian path in (V, E) starting at v_1, say $[v_1, \ldots, v_n]$, then there is a solution for the planning task where the movement path of the mobile is $[v_1, v_1^*, \ldots, v_n, v_n^*]$ and portables are picked up and dropped in the obvious way.

Now consider there is a solution to the planning task. Whenever a portable is picked up (at an entrance), the only reasonable thing to do is to move to its destination (the corresponding exit) and drop it, because there is no use in deferring that movement when the carrying capacity is exhausted. The mobile must then proceed to the next entrance, which is only possible in the ways defined by the edges in the original graph. Thus, the plan corresponds to a path in the original graph that visits every vertex. It must be Hamiltonian, because if an entrance were ever visited twice, it could never be left again because of fuel constraints. □

Although the same reduction could be used in the infinite capacity case, we give another proof for this case showing that it is already **NP**-complete even if the roadmap is restricted to be a planar graph.

Theorem 4. PLANEX-TRANSPORT$_{\infty 11}$ *is* **NP***-complete*
PLANEX-TRANSPORT$_{\infty 11}$ *is* **NP***-complete, even if the roadmap is restricted to be a planar graph.*

Proof. Membership in **NP** is already known. For hardness, we reduce from the Hamiltonian Path problem with a fixed start vertex in a planar graph [8]. Let (V, E) be the graph and $v_1 \in V$. Then (V, E) contains a Hamiltonian path starting at v_1 if and only if there is a solution for the TRANSPORT task $I \in \mathcal{I}_{\infty 11}$ defined as follows: The roadmap of the planning task is (V, E), each location provides one unit of fuel, and there is one portable to be delivered to each location from v_1, the initial location of the only mobile (of unlimited capacity).

Clearly, this problem is solvable if and only if there is a Hamiltonian Path in (V, E) starting at v_1. □

This concludes our analysis of the PLANEX-TRANSPORT$_{ijk}$ decision problems. They can be solved in polynomial time if $j = \infty$ and are **NP**-complete otherwise.

2.2 Bounded Plan Existence

Theorems 1, 3 and 4 and Corollary 1 imply **NP**-completeness for PLANLEN-TRANSPORT$_{ijk}$ for $j \neq \infty$. In this subsection, we will show that the same result holds in the unrestricted fuel case, even in some very limited special cases.

In fact, the proofs of Theorems 3 and 4 can be adjusted to prove **NP**-completeness of PLANLEN-TRANSPORT$_{1\infty 1}$ and PLANLEN-TRANSPORT$_{\infty\infty 1}$ by replacing the fuel restrictions with plan length bounds of $4|V| - 1$ and $3|V| - 3$, respectively. However, these results require allowing for arbitrary (or arbitrary planar) roadmaps, and thus do not apply to planning domains such as LOGISTICS or GRID. For that reason, we will prove some stronger results now.

The first result in this section applies to *grid* roadmaps, i. e. graphs with vertex set $\{0, \ldots, w\} \times \{0, \ldots, h\}$ for some $w, h \in \mathbb{N}$ (called *width* and *height* of the grid, respectively), where vertices (a, b) and (a', b') are connected by an edge if and only if $|a - a'| + |b - b'| = 1$. Note that grids are always planar graphs.

Theorem 5. PLANLEN-TRANSPORT$_{1\infty 1}$ *is* **NP***-complete*
PLANLEN-TRANSPORT$_{1\infty 1}$ *is* **NP***-complete, even if the roadmap is restricted to be a grid.*

Proof. Membership in **NP** is already known. For hardness, we reduce from the $\mathcal{L}_1$ metric TSP, which is **NP**-complete in the strong sense [6, Problem ND23].[4]

Omitting the technical details which can be found elsewhere [8], the key idea is to have one portable for each site in the TSP instance, which needs to be moved to an adjacent location. The mobile starts at the northmost (with the highest y coordinate) TSP site and has to visit each site in order to deliver all portables, and the number of movements needed for that is equal to the length of the shortest non-closed TSP tour (i. e. a tour not returning to the initial location). The tour can be closed by putting an additional portable that needs to be moved "far up north".

To enforce that the length of the shortest plan is dominated by the movement between sites rather than movement between the initial and (adjacent) goal locations of portables the coordinates of the sites are scaled by a factor of $2n$ (n being the number of sites). □

The same reduction can be used in the unrestricted capacity case [8]. Additionally, in this setting the following result holds.

Theorem 6. PLANLEN-TRANSPORT$_{\infty\infty 1}$ *is* **NP***-complete*

PLANLEN-TRANSPORT$_{\infty\infty 1}$ *is* **NP***-complete, even if the roadmap is restricted to be a complete graph.*

Proof. Membership in **NP** is already known. For hardness, we reduce from the Feedback Vertex Set problem [6, Problem GT7]. Let (V, A) be a digraph and $K \in \mathbb{N}$. Then (V, A) contains a feedback vertex set of size at most K if and only if there is a solution of length at most $3|V| + 2|A| + K$ for the TRANSPORT task $I \in \mathcal{I}_{\infty\infty 1}$ where the roadmap is a complete graph with locations V and an additional location v_0, which is the initial location of the only mobile, there are no capacity or fuel constraints, there is one portable to be moved from v_0 to each $v \in V$ and one portable to be moved from u to v for each $(u, v) \in A$.

To see this, observe that for each feedback vertex set $V' \subseteq V$, the planning task can be solved by moving the mobile to the vertices from V' in any order, then to the vertices from $V \setminus V'$ in an order which is consistent with the arcs in the subgraph induced by $V \setminus V'$ (which must be acyclic because V' is a feedback vertex set), and finally to the vertices from V' again, in any order, picking up and dropping portables in the obvious way. This requires $|A| + |V|$ pickup and drop actions each and $|V| + |V'|$ movements, totaling a number of actions bounded by $3|V| + 2|A| + K$ if $|V'| \leq K$.

On the other hand, any plan must contain at least one pickup and drop action for each portable and visit each location at least once, totaling $3|V|+2|A|$ actions, so if a plan does not exceed the given length bound, there cannot be more than K locations that are visited more than once. These locations must form a feedback vertex set. □

[4] Our transformation is only polynomial if numbers in the original TSP instance are encoded in unary, but this is a valid assumption for decision problems that are **NP**-complete in the strong sense.

3 Competition Domains from AIPS 1998/2000

Having completed the analysis of the TRANSPORT domain, we can now apply these results to the transportation domains from the planning competition.[5]

The MYSTERY domain [15] is equal to our $\mathcal{I}_{**+}$ task set. Thus, plan existence and bounded plan existence are **NP**-complete in this domain, even in the case of planar roadmaps, according to Theorems 1 and 4 and Corollary 1. This still holds if there is only one mobile and all portables start at the same location as the mobile.

The MYSTERY' domain [15] adds operators to move fuel between locations to the original MYSTERY domain. However, these can only be applied if at least two units of fuel are present at a given location, so for tasks from $\mathcal{I}_{*1+}$, there is no difference between the two domains and consequently the same hardness results apply for MYSTERY'. Membership in **NP** for the decision problems related to MYSTERY' follows from a polynomial plan length argument, as for the number of move, pickup and drop actions the same bounds as for TRANSPORT tasks apply, and there is no need to have more actions that move fuel than movements of mobiles.

LOGISTICS tasks [15] are special cases of $\mathcal{I}_{\infty\infty*}$ tasks and generalizations of $\mathcal{I}_{\infty\infty1}$ tasks with complete graph roadmaps. Thus, according to Theorems 1, 2, and 6, plans can be found in polynomial time in this domain, but the bounded plan existence problem is **NP**-complete, even if there is only one mobile (either truck or airplane).

The GRIPPER domain [15] is a specialization of $\mathcal{I}_{*\infty1}$ and thus allows for generating plans in polynomial time. Of course, this domain is so simple that even optimal plans can be generated in polynomial time.

For tasks without doors, the GRID domain [15] is very similar to $\mathcal{I}_{1\infty1}$ with grid roadmaps[6], thus the bounded plan existence problem in this domain is **NP**-hard, even in the absence of doors. It is actually in **NP** and thus **NP**-complete (with or without doors), again by a polynomial plan length argument, as it is not hard to bound the number of actions between two *unlock* actions in a reasonable plan, and no location can be unlocked more than once.

If optimality is not required, plans can be generated in the GRID domain in polynomial time by a simple strategy unlocking door after door as long as this is possible and then moving the keys to their goal destinations if reachable. [8]

Concluding our discussion of GRID, we want to briefly mention another proof of **NP**-hardness for the bounded plan existence problem without going into detail (cf. [8]). This reduction does not emphasize the route planning aspect of the domain and instead makes use of doors and is illustrated in Figure 1.

---

[5] Since a "benchmark domain" is not defined by the PDDL domain file alone (consider the LOGISTICS domain, where it is implicitly assumed that in well-formed problems the sets of portables, trucks and airplanes are disjoint), we refer to the literature for informal [1, 14, 15] and formal definitions [8] of these planning tasks.

[6] The only difference is that in GRID, the portable in hand can be swapped with a portable at the current location in just one action, but this does not make a difference for the proof of Theorem 5.

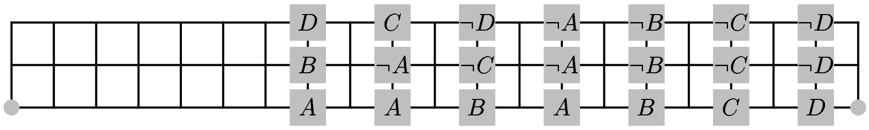

Fig. 1. GRID instance corresponding to $(A \vee B \vee D) \wedge (A \vee \neg A \vee C) \wedge (B \vee \neg C \vee \neg D)$. Locations with doors are marked with squares. The bottom left location contains the mobile and one key for each literal, opening the corresponding doors. The bottom right location contains an additional key. All keys must be moved to the bottom left location.

3.1 MICONIC-10

For the remaining competition domain, the MICONIC-10 elevator domain [12], things are a bit more complicated. There are actually three different domains under that name that were part of the AIPS 2000 competition. The first, called MICONIC-10 STRIPS, defines tasks very similar to LOGISTICS with one mobile, or $\mathcal{I}_{\infty\infty 1}$ with complete graphs. The only difference is that portables (passengers) can only be dropped at their destination locations and can never be picked up again (reboard the elevator). Theorems 1, 2, and 6 apply, and thus plans can be found in polynomial time, but deciding existence of a bounded length plan is an **NP**-complete problem.

The same is true for the second version of MICONIC-10, called *simple ADL* . In this version, all boarding and leaving at a given floor (picking up or dropping) is automatically handled by a single *stop* action with conditional effects. It causes all passengers inside the elevator with that goal destination to leave and all passengers waiting outside to board. This only requires a minor change to the proof of Theorem 6, changing the plan length bound to $2(|V|+K)+1$ for $|V|+K$ movements of the elevator and $|V|+K+1$ *stop* actions.

The "real" MICONIC-10 domain additionally introduces special passengers which impose movement restrictions on the elevator. Most importantly, the elevator may only stop at floors to which all passengers inside the elevator have access, and there are "attended" passengers who require the presence of at least one "attendant" passenger as long as they are inside the cabin (if the last attendant leaves the elevator, a new one must board). There are also VIP passengers who must be served with priority.

The decision problems related to that domain are still in **NP** because the number of stops can be bounded by twice the number of passengers to be served (one stop at their initial, another at their goal floor), and this in turn bounds the number of movements. However, as it turns out, plan existence is already **NP**-hard in this domain. Due to space restrictions, we will not give a formal definition of the decision problem at hand, which can be found in [8]. It should be possible to understand the following proof without those details, though.

Theorem 7. PLANEX-MICONIC-10 *is* **NP**-*complete*

Proof. Membership in **NP** has been shown. For **NP**-hardness, we reduce from the problem of finding a Hamiltonian path with a fixed start vertex v_1 in a digraph (V, A) [6, Problem GT39].

The corresponding MICONIC-10 task has the following floors: an *init floor* f_0, *final floor* f_∞, for each vertex u a *vertex start floor* f_u and *vertex end floor* f_u^*, and for each arc (u, v) an *arc floor* $f_{u,v}$. F is the set of all these floors and for each vertex u, F_u is the set containing f_u, f_u^*, and the arc floors for outgoing arcs of u. These are the passengers to be served:

Passenger	From	To	Access to ...	Special
p_0	f_0	f_{v_1}	$\{f_0, f_{v_1}\}$	VIP, attendant
$\forall u \in V$: p_u	f_0	f_u	$F \setminus \{f_\infty\}$	attended
$\forall u \in V$: p_u^*	f_u	f_u^*	$F_u \cup \{f_\infty\}$	attendant
$\forall u \in V$: p_u^∞	f_u^*	f_∞	$F \setminus \{f_u\}$	none
$\forall (u,v) \in A$: $p_{u,v}$	$f_{u,v}$	f_v	$\{f_{u,v}, f_u^*, f_v\}$	attendant

Assume that it is possible to solve the task. Because p_0 is a VIP, the first stops must be at f_0 and f_{v_1}, picking up all the attended passengers and $p_{v_1}^*$. Because of the movement restrictions of that passenger, the journey can only proceed to floors from F_{v_1}, and $f_{v_1}^*$ is not an option because going there would lead to the only attendant leaving. Thus, the elevator must go to f_{v_1,v_2} (for some vertex v_2 that is adjacent to v_1) and can then only proceed to $f_{v_1}^*$ and then f_{v_2}, picking up $p_{v_1}^\infty$.

We are now in a similar situation as upon arrival at f_{v_1}, and again, the elevator will eventually go to some floor f_{v_3}, then f_{v_4}, following the arcs of the digraph (V, A) in a path $[v_1, \dots, v_n]$ until all vertices have been visited once. No vertex can be visited twice because of the passengers of type p_u^∞. So plan existence implies a Hamiltonian path starting at v_1 in the digraph.

On the other hand, if a Hamiltonian path exists, there is a sequence of elevator movements that leads to all attended passengers having arrived at their final destination and the elevator being at some floor f_u for $u \in V$. No longer requiring attendants, it can then immediately proceed to f_u^*, then f_∞ and finally serve the remaining passengers of type $f_{u,v}$ (for arcs (u, v) not part of the Hamiltonian path), one after the other, completing the plan. □

4 Discussion

Let us briefly summarize the results of our analysis. For fairly general transportation tasks, we have shown **NP**-completeness of non-optimal planning in the restricted fuel case and **NP**-completeness of optimal planning in all cases. Just finding some plan in tasks where fuel is abundant was shown to be a polynomial problem.

This is detailed in Figure 2. For some domains, even some severe restrictions are still sufficient to get **NP**-hardness. Specifically, all **NP**-hardness results in the multi-agent competition domains still hold if there is only one agent, and the **NP**-hardness result for GRID still holds if there are no doors at all. For convenience, we repeat the results for the competition domains:

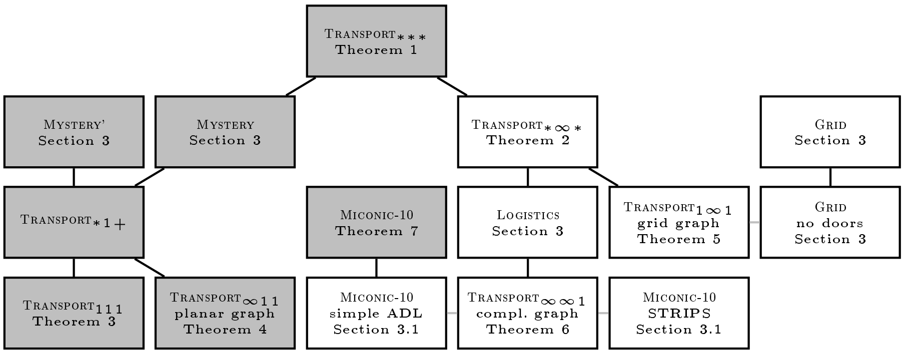

Fig. 2. The transportation domains hierarchy. Black lines indicate special cases, gray lines strong similarities of domains. Deciding plan existence is **NP**-complete for domains with gray boxes, plans can be generated in polynomial time for domains in white boxes. The bounded plan length problem is **NP**-complete for all domains in the figure. For the Gripper domain (not shown), both problems are polynomial.

Domain name	PlanEx	PlanLen
Grid	polynomial	**NP**-complete
Gripper	polynomial	polynomial
Logistics	polynomial	**NP**-complete
Miconic-10 (STRIPS or simple ADL)	polynomial	**NP**-complete
Miconic-10 (full ADL)	**NP**-complete	**NP**-complete
Mystery, Mystery'	**NP**-complete	**NP**-complete

It is interesting to observe that in those domains where heuristic local search planners such as **FF** [10] excel, the table lists different results for plan existence and bounded plan existence. Because all hardness proofs only use a single agent, they carry over to *optimal parallel planning*, which implies that in these domains planners like **Graphplan** [3] or **IPP** [11] try to solve provably hard subproblems that local search planners do not have to care about. When optimal plans are not required, local search has a conceptual advantage here, and we cannot hope for similar performance from any planner striving for optimality.

Greedy local search is less appropriate, however, if additional constraints can lead to dead ends in the search space. We have faced this problem when dealing with fuel constraints and in the full Miconic-10 domain, where it may be unwise to have people board the elevator who restrict its movement too much. In fact, the competition domains with **NP**-hard plan existence problems are exactly the ones for which current planners based on heuristic local search encounter unrecognized dead ends.[7] [9]

While the observation that non-optimal planning is often easier than optimal planning is by no means surprising or new, we consider it important to point out. While there has been significant recent progress on non-optimal planning, optimal planners tend to get less attention than they deserve, maybe due to the

[7] This is also true for the non-transportation benchmarks. [8, 9]

fact that they are often compared to their non-optimal counterparts in terms of the size of problems they can handle. This kind of comparison is hardly fair.

We also observe that all discussed decision problems are in **NP**. We do not consider this a weakness of the benchmark set, as in STRIPS/ADL planning, **NP** membership is guaranteed as soon as there are polynomial bounds on plan lengths, which is a reasonable restriction from a plan execution point of view.

5 Related Work

Other work in the AI planning literature concerned with computational complexity results mostly focuses on domain-independent planning, analyzing different variants of the planning problem and special cases thereof [2, 4, 5]. This work mainly covers purely *syntactical* restrictions of general planning, such as limiting the number of operator preconditions or effects, but also discusses the complexity of STRIPS-style planning in (arbitrary) fixed domains [5].

There are very few articles in the planning literature which are concerned with the same kind of domain-dependent planning complexity results as this work. The existing literature concentrates on the complexity of BLOCKSWORLD, including results for generalizations of the classical domain, e. g. allowing for blocks of different size. The most comprehensive reference for this line of research is an article by Gupta and Nau [7]. There is also a very interesting discussion of the important distinction between optimal, near-optimal and non-optimal planning in BLOCKSWORLD. [16]

The usefulness of the idea of partitioning planning domains into families like *transportation* and most of the corresponding terminology is borrowed from work by Long and Fox [13], although in that paper the focus is on the automatic *detection* of transportation domains and the exploitation of some of their features by a planning algorithm, not on complexity aspects.

6 Outlook

While some questions were answered in the preceding sections, open issues remain. In some domains it would be interesting to investigate some more special cases to come up with more fine-grained results. For example, in the full MICONIC-10 domain, plan existence is **NP**-complete, but it is polynomial without special passengers and access restrictions. What is the complexity if only *some* of these enhancements are made?

Where plan existence is **NP**-complete, detecting the *phase transition* between (usually easy) under-constrained and (usually easy) over-constrained instances would be interesting, increasing the benefit of these domains for benchmarking.

Finally, in addition to discussing "optimal" and "non-optimal" planning, near-optimal planning is an interesting topic for domains where plan existence and bounded plan existence are of different complexity [17]. Giving performance guarantees is certainly easy in LOGISTICS and the restricted MICONIC-10 domains, but what about GRID?

References

1. Fahiem Bacchus and Dana S. Nau. The AIPS-2000 planning competition. *AI Magazine*, 2001. To appear.
2. Christer Bäckström and Bernhard Nebel. Complexity results for SAS^+ planning. *Computational Intelligence*, 11(4):625–655, 1995.
3. Avrim Blum and Merrick Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90(1–2):281–300, 1997.
4. Tom Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1–2):165–204, 1994.
5. Kutluhan Erol, Dana S. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 76(1–2):65–88, 1995.
6. Michael R. Garey and David S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness.* Freeman, 1979.
7. Naresh Gupta and Dana S. Nau. On the complexity of blocks-world planning. *Artificial Intelligence*, 56(2–3):223–254, 1992.
8. Malte Helmert. On the complexity of planning in transportation and manipulation domains. Master's thesis, Albert-Ludwigs-Universität Freiburg, 2001. Postscript available at http://www.informatik.uni-freiburg.de/~ki/theses.html.
9. Jörg Hoffmann. Local search topology in planning benchmarks: An empirical analysis. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, 2001. Accepted for publication.
10. Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.
11. Jana Köhler, Bernhard Nebel, Jörg Hoffmann, and Yannis Dimopoulos. Extending planning graphs to an ADL subset. In S. Steel and R. Alami, editors, *Recent Advances in AI Planning. 4th European Conference on Planning (ECP'97)*, volume 1348 of *Lecture Notes in Artificial Intelligence*, pages 273–285, New York, 1997. Springer-Verlag.
12. Jana Köhler and Kilian Schuster. Elevator control as a planning problem. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, pages 331–338, 2000.
13. Derek Long and Maria Fox. Automatic synthesis and use of generic types in planning. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS 2000)*, pages 196–205, 2000.
14. Derek Long, Henry Kautz, Bart Selman, Blai Bonet, Hector Geffner, Jana Köhler, Michael Brenner, Jörg Hoffmann, Frank Rittinger, Corin R. Anderson, Daniel S. Weld, David E. Smith, and Maria Fox. The AIPS-98 planning competition. *AI Magazine*, 21(2):13–33, 2000.
15. Drew McDermott. The 1998 AI Planning Systems competition. *AI Magazine*, 21(2):35–55, 2000.
16. Bart Selman. Near-optimal plans, tractability, and reactivity. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR'94)*, pages 521–529. Morgan Kaufmann, 1994.
17. John Slaney and Sylvie Thiébaux. Blocks world revisited. *Artificial Intelligence*, 125:119–153, 2001.