Improving Hierarchical Planning Performance by the Use of Landmarks

Mohamed Elkawkagy, Pascal Bercher,

Bernd Schattenberg, and Susanne Biundo

Institute of Articial Intelligence

Ulm University

Germany

AAAI-12, 26th July, 2012, Toronto, Canada

Agenda

- Introduction
- Motivation
- Landmark-Aware Strategies
- Evaluation
- Conclusion

HTN Planning- Problem Formalization

$\mathbf{\Pi} = < \mathbf{S_{init}}, \mathbf{P_{init}}, \mathbf{D} >$ is an HTN planning problem with

- ❖ $\mathbf{S_{init}}$ is the initial state
- ❖ $\mathbf{P_{init}}$ is the initial plan that needs to be decomposed
- ❖ $\mathbf{D} = < \mathbf{T}, \mathbf{M} >$ is the domain model with
  - ▪ **T** is a set of task schemata of the form

    $\mathbf{t(\tau)} = < \mathbf{pre}, \mathbf{eff} >$, τ is the parameter list of t
  - ▪ **M** is a set of decomposition methods of the form

    $\mathbf{m} = < \mathbf{t(\tau)}, \mathbf{P} >$, **P** is a partial plan,

    - o partial plan $\mathbf{P} = < \mathbf{S}, \mathbf{C} >$

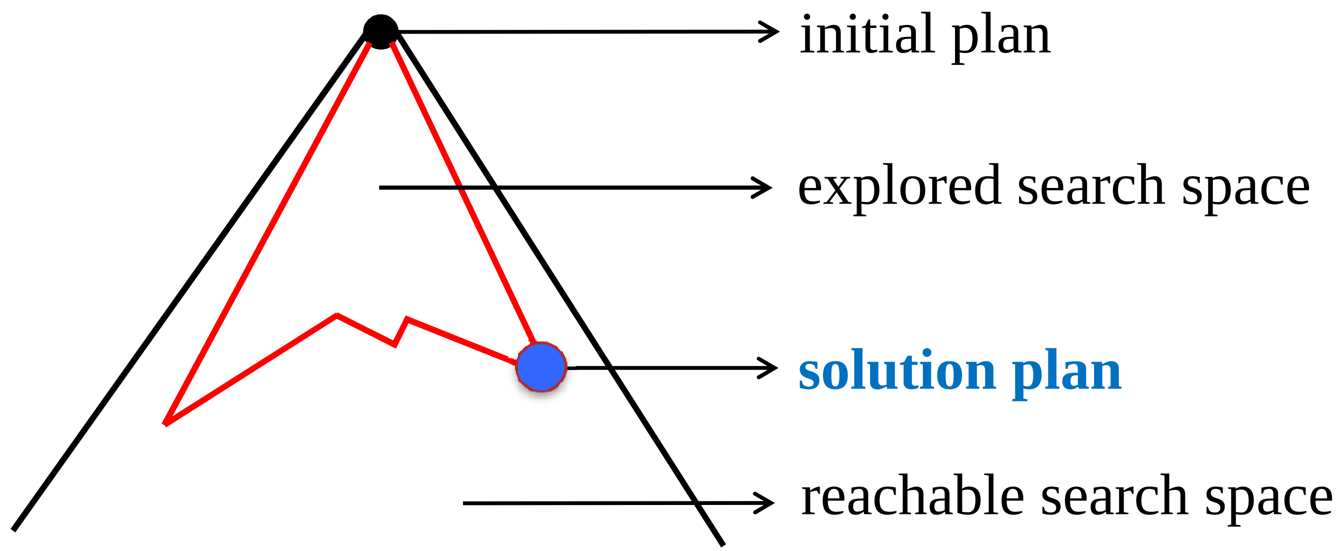
HTN Planning - Solution Formalization

A Solution of a planning problem $\mathbf{\Pi = < \mathit{S_{init}}, \mathit{P_{init}}, \mathit{D} >}$ is obtained by refining the initial plan $\boldsymbol{P_{init}}$ stepwise into a plan $\mathbf{P}_{\boldsymbol{sol}}$, s.t.

- $\mathbf{P}_{\boldsymbol{sol}}$ contains only primitive tasks,
- $\mathbf{P}_{\boldsymbol{sol}}$ is obtained from $\mathbf{P_{init}}$ by decomposing non-primitive tasks,
- $\mathbf{P}_{\boldsymbol{sol}}$ is executable in $\boldsymbol{S_{init}}$

Motivation

❖Hierarchical planning is based on search techniques in the space of plans.

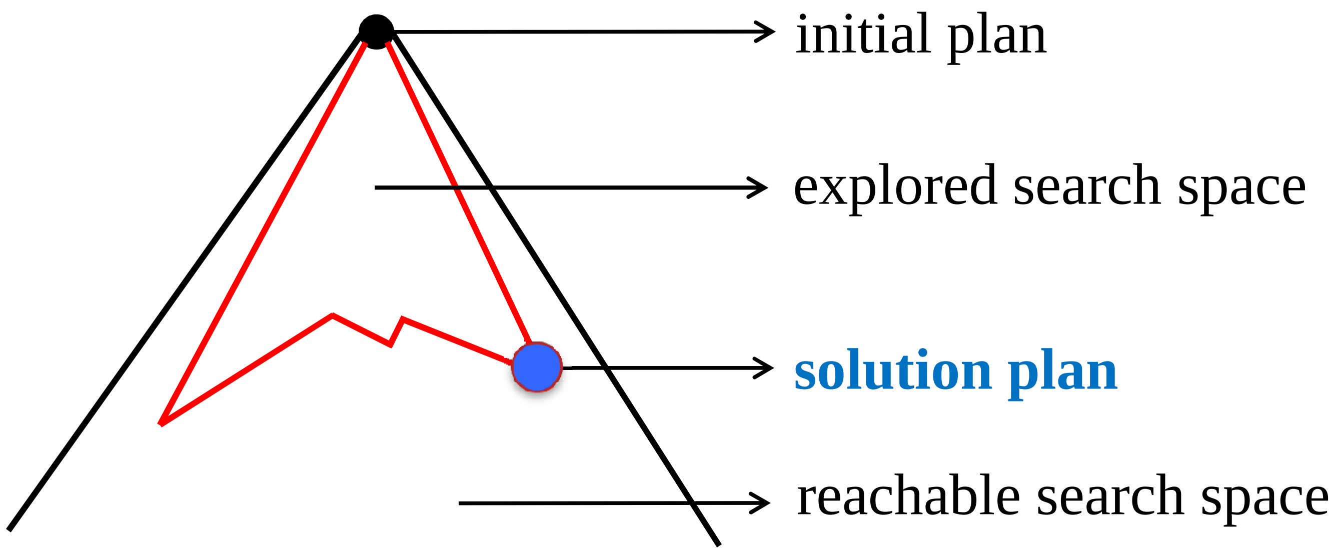


❖Well-known HTN planners and their strategies:

- UMCP*[Erol et al., 1995]*: (make primitive, then resolve conflicts)
- EMS*[McCluskey, 2000]*: (expand, then make sound)
- SHOP Family*[Nau et al., 1999, Nau et al., 2003]*: (progression)

Motivation

❖Hierarchical planning is based on search techniques in the space of plans.



❖ Development of novel domain-independent search strategies for HTN-based planning :

- Strategies estimate future branching factor based on landmark information.

Search Procedure

Algorithm: Refinement Planning Algorithm

Input : The sequence $\texttt{Fringe} = \langle P_{\text{init}} \rangle$.
Output : A solution or $\texttt{Fail}$.

while $\texttt{Fringe} = \langle P_1 \ldots P_n \rangle \neq \varepsilon$ **do**
  $F \leftarrow f^{\text{FlawDet}}(P_1)$
  if $F = \emptyset$ **then return** P_1
  $\langle \texttt{m}_1 \ldots \texttt{m}_k \rangle \leftarrow f^{\text{ModOrd}}\left(\bigcup_{\texttt{f} \in F} f^{\text{ModGen}}(\texttt{f})\right)$
  $\texttt{succ} \leftarrow \langle \text{app}(\texttt{m}_1, P_1) \ldots \text{app}(\texttt{m}_k, P_1) \rangle$
  $\texttt{Fringe} \leftarrow f^{\text{PlanOrd}}(\texttt{succ} \circ \langle P_2 \ldots P_n \rangle)$
return $\texttt{fail}$

orders plan modifications according to a given strategy

Orders the updated fringe

Search Procedure

Algorithm: Refinement Planning Algorithm

Input : The sequence $\texttt{Fringe} = \langle P_{\text{init}} \rangle$.
Output : A solution or $\texttt{Fail}$.

while $\texttt{Fringe} = \langle P_1 \ldots P_n \rangle \neq \varepsilon$ **do**
  $F \leftarrow f^{\text{FlawDet}}(P_1)$
  if $F = \emptyset$ **then return** P_1
  $\langle \texttt{m}_1 \ldots \texttt{m}_k \rangle \leftarrow f^{\text{ModOrd}}\left(\bigcup_{\texttt{f} \in F} f^{\text{ModGen}}(\texttt{f})\right)$
  $\texttt{succ} \leftarrow \langle \text{app}(\texttt{m}_1, P_1) \ldots \text{app}(\texttt{m}_k, P_1) \rangle$
  $\texttt{Fringe} \leftarrow f^{\text{PlanOrd}}(\texttt{succ} \circ \langle P_2 \ldots P_n \rangle)$
return $\texttt{fail}$

Use landmarks to reduce branching factor by deciding which non-primitive task to decompose first.

Landmarks

❖ **Landmarks** are tasks that occur in the plan sequences leading from the problem s initial plan to its solution *[Elkawkagy et al, 2010]*.

Table: Landmark Table

t	$M(t)$	$O(t)$
$t_1(c_1)$	$\{t_2(c_1)\}$	$\{\{t_3(c_2), t_3(c_1)\}, \{t_1(c_1)\}\}$
$t_3(c_2)$	$\{t_4(c_2)\}$	$\{\emptyset, \{t_5(c_2), t_2(c_2)\}\}$
$t_3(c_1)$	$\{t_4(c_1)\}$	$\{\emptyset, \{t_5(c_1), t_2(c_1)\}\}$

Mandatory Tasks:

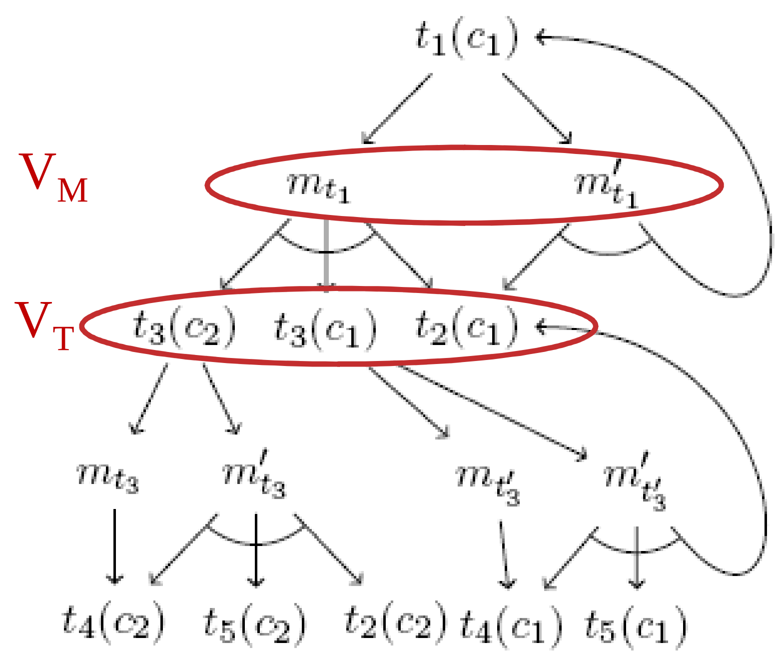
Intersection of method's plans

Optional Tasks:

Remaining tasks

Figure: Task Decomposition Graph

TDG = $\langle V_T, V_M, E\rangle$



Landmark-Aware Strategies – lm_2

lm_2

Given a partial plan P, let $t_i, t_j \in P$ be abstract tasks.

Let LT be a landmark table and

$$\langle t_i, M(t_i), O(t_i) \rangle \in LT$$

$$\langle t_j, M(t_j), O(t_j) \rangle \in LT$$

$\mathbf{f}^{\mathbf{ModOrd}}_{\mathbf{lm_2}}$ decomposes task t_i before task t_j if and only if

$$\sum_{o \in O(t_i)} |\mathbf{o}|_{LT} \prec \sum_{o \in O(t_j)} |\mathbf{o}|_{LT}$$

Landmark-Aware Strategies – lm$_1$

lm_1

Given a partial plan P, let t_i, $t_j \in$ P be abstract tasks.

Let LT be a landmark table and

$$\langle\, t_i, M(t_i), O(t_i) \,\rangle \in LT$$

$$\langle\, t_j, M(t_j), O(t_j) \,\rangle \in LT$$

$\mathbf{f}^{\mathbf{ModOrd}}_{\mathbf{lm_1}}$ decomposes task t_i before task t_j if and only if

$$\sum_{\mathbf{o} \in \mathbf{O}(\mathbf{t}_i)} |\mathbf{o}|_{\mathbf{LT}} \prec \sum_{\mathbf{o} \in \mathbf{O}(\mathbf{t}_j)} |\mathbf{o}|_{\mathbf{LT}}$$

for $|\mathbf{o}|_{\mathbf{LT}} := \left| \left\{ \mathbf{t} \in \mathbf{o} \mid \langle \mathbf{t}, \mathbf{M}(\mathbf{t}), \mathbf{O}(\mathbf{t}) \in \mathbf{LT} \rangle \right\} \right|$

Landmark-Aware Strategies

Example

Let a plan P contain two abstract tasks $\mathbf{t_1(c_1)}$ **and** $\mathbf{t_3(c_2)}$.

Let

$$\langle t_1(c_1), \{t_2(c_1)\}, \{\{t_3(c_2), t_3(c_1)\}, \{t_1(c_1)\}\}\rangle \in LT$$

$$\langle t_3(c_1), \{t_4(c_1)\}, \{\{t_5(c_1), t_2(c_1)\}\}\rangle \in LT$$

Primitive tasks

$$f^{ModOrd}_{lm_1}: \sum_{o \in O(t_3)} |o|_{LT} \prec \sum_{o \in O(t_1)} |o|_{LT} \Leftrightarrow 0 \prec 2 + 1$$

Decompose t_3

$$f^{ModOrd}_{lm_2}: \sum_{o \in O(t_3)} |o| \prec \sum_{o \in O(t_1)} |o| \Leftrightarrow 2 \prec 2 + 1$$

Decompose t_3

Landmark-Aware Strategies – lm_1^*, lm_2^*

Closure of the optional set

The closure of the optional set for a given task **t** and landmark table **LT** is the transitive closure of the optional sets on a recursive traversal of the table entries, begining in task **t**.

$$\mathbf{O}^*(\mathbf{t}) = \mathbf{O}(\mathbf{t}) \cup \bigcup_{o \in O(t)} \left(\bigcup_{\bar{t} \in o} \mathbf{O}^*(\bar{t}) \right)$$

$$\text{with } \langle \mathbf{t}, \mathbf{M}(\mathbf{t}), \mathbf{O}(\mathbf{t}) \rangle \in \mathbf{LT}$$

t	M(t)	O(t)	$O^*(t)$
$t_1(c_1)$	$\{t_2(c_1)\}$	$\{\{t_3(c_2), t_3(c_1)\}, \{t_1(c_1)\}\}$	$O(t_1(c_1)) \cup O(t_3(c_2)) \cup O(t_3(c_1))$
$t_3(c_2)$	$\{t_4(c_2)\}$	$\{\{t_5(c_2), t_2(c_2)\}\}$	$O(t_3(c_2))$
$t_3(c_1)$	$\{t_4(c_1)\}$	$\{\{t_5(c_1), t_2(c_1)\}\}$	$O(t_3(c_1)) \cup O(t_2(c_1))$

Evaluation Benchmark

❖ We ran our evaluations over four distinguished benchmark domains:

Domain Name	Methods	Abstract tasks	Primitive tasks
UM-Translog	51	21	48
SmartPhone	94	50	87
Satellite	8	3	5
WoodWorking	14	6	13

❖ ***UM-Translog Domain*** describes scenarios of transporting various types of goods by various means (trucks, trains,…) via appropriate infrastructure (roads, transport centers,…).

❖ ***SmartPhone*** manges the operation of a smart phone by a human user such as sending messages and creating contacts or appointments.

Evaluation Benchmark

❖ We ran our evaluations over four distinguished benchmark domains:

Domain Name	Methods	Abstract tasks	Primitive tasks
UM-Translog	51	21	48
SmartPhone	94	50	87
Satellite	8	3	5
WoodWorking	14	6	13

❖ ***Satellite Domain*** manages scientific stellar observations by earth orbiting instrument platforms.

❖ ***WoodWorking*** specifies the processing of raw wood into smooth and varnished product parts.

Evaluation - Comparison

- HTN strategies like lcf (least commitment first) and HotSpot/HotZone (modifications that don't effect many plan fragments).
- Planning systems mentioned in motivation: our generic refinement algorithm can simulate behavior of (almost) any system when using the according modification $\mathbf{f}^{\mathbf{ModOrd}}$ and plan ordering $\mathbf{f}^{\mathbf{PlanOrd}}$ functions.
- Org columns show the time needed to solve the *original* (unreduced) problem.
- Red columns show the time in seconds needed to solve the problem if our domain *reduction* is used.

Evaluation

UM-Translog domain	#1		#3		#5	
f_{ModOrd}	org	red	org	red	org	red
UMCP	952	244	994	229	187	122
EMS	2056	1048	2199	1806	969	259
SHOP	1735	353	1911	274	5874	4012
Lm_1	**243**	**180**	**447**	**184**	**121**	**111**
Lm_1^*	1772	212	**370**	**205**	657	**109**
Lm_2	3311	255	1670	248	716	162
Lm_2^*	846	226	991	238	583	340
Lcf	1878	225	3020	209	187	118
Da-HotSpot	2414	1958	-	2030	589	352
Du-HotSpot	1319	775	987	1090	692	224
HotZone	**473**	**196**	498	224	**149**	121

SmartPhone domain	#1		#2		#5	
f_{ModOrd}	org	red	org	red	org	red
UMCP	80	**30**	256	**115**	-	-
EMS	107	52	235	148	-	-
SHOP	95	73	-	-	-	-
Lm_1	**50**	**30**	**134**	**53**	-	**465**
Lm_1^*	65	50	392	173	-	-
Lm_2	60	50	**181**	**53**	-	**680**
Lm_2^*	98	76	1632	327	-	697
Lcf	63	40	-	159	**8455**	6827
Da-HotSpot	**45**	43	-	203	**1747**	1041
Du-HotSpot	52	46	638	166	-	3421
HotZone	65	**33**	490	212	-	-

- **red** == best result / **blue** == second-best result
- Dashes indicate that no solution was found within a limit of 150 minutes.

Evaluation

WoodWorking domain

f_{ModOrd}	#1		#3		#5	
	org	**red**	**org**	**red**	**org**	**red**
UMCP	228	133	259	125	892	218
EMS	415	298	-	2457	-	512
SHOP	-	-	-	-	-	3578
Lm_1	**96**	**55**	**171**	159	**564**	**197**
Lm_1^*	**82**	**50**	614	**98**	2109	1245
Lm_2	881	433	-	362	-	-
Lm_2^*	1359	403	-	367	-	893
Lcf	2067	350	-	-	-	-
Da-HotSpot	113	85	**355**	**110**	**573**	**201**
Du-HotSpot	-	-	-	-	-	-
HotZone	-	-	-	-	-	-

Satellite domain

f_{ModOrd}	#1		#2		#3	
	org	**red**	**org**	**red**	**org**	**red**
UMCP	**51**	**41**	2035	1336	4150	1894
EMS	62	53	2608	2856	-	-
SHOP	113	111	**270**	**264**	-	-
Lm_1	209	208	**767**	652	**458**	**400**
Lm_1^*	**54**	**43**	1024	969	**2617**	2569
Lm_2	151	140	-	5804	2816	251
Lm_2^*	191	99	-	-	2636	2553
Lcf	154	77	1551	1338	-	4069
Da-HotSpot	85	78	2136	1131	-	1131
Du-HotSpot	270	150	-	-	-	-
HotZone	142	62	-	4764	-	-

❖ We solved problems for which some established strategies do not find solutions within the given resource bounds.

Conclusion

- ❖ Partitioning of method's tasks in mandatory and optional sets.
- ❖ Landmark strategies use size of (closure of) optional sets to estimate branching factor.
- ❖ Landmark-aware strategies outperform the established ones on almost the evaluated problems .