Hybrid Multi-agent Planning

Mohamed Elkawkagy and Susanne Biundo

Institute of Articial Intelligence

October 6th, 2011
Berlin, Germany

ulm university universität ulm

HTN Planning - Problem Formalization

- $\Pi = < S_{init} , P_{init} , D >$ is an HTN planning problem with
  - S_{init} is the initial state
  - P_{init} is the initial partial plan that needs to be decomposed
  - $D = <$ T , M $>$ is the domain model with
    - T is a set of task schemata of the form

      t(τ) = < pre , eff > , τ is the parameter list of t
    - M is a set of decomposition methods of the form

      m = < t(τ), P >, P is a partial plan

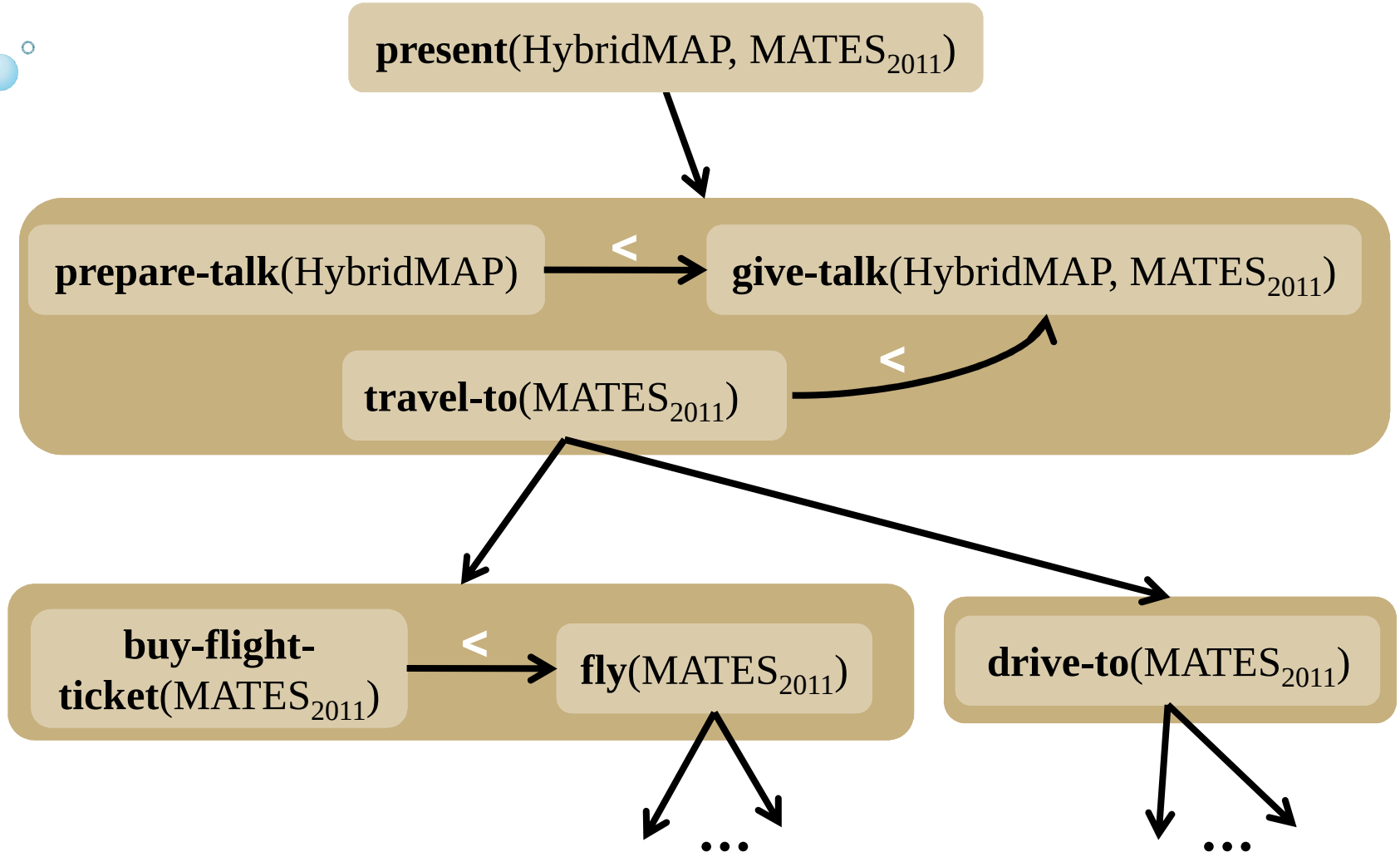
HTN Planning - Solution Formalization

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

The partial plan P_{sol} is a solution to if and only if

- P_{sol} contains only primitive tasks
- P_{sol} is obtained from P_{init} by decomposing non-primitive tasks
- Every linearization of P_{sol} is executable in s_{init}

HTN Planning - Example

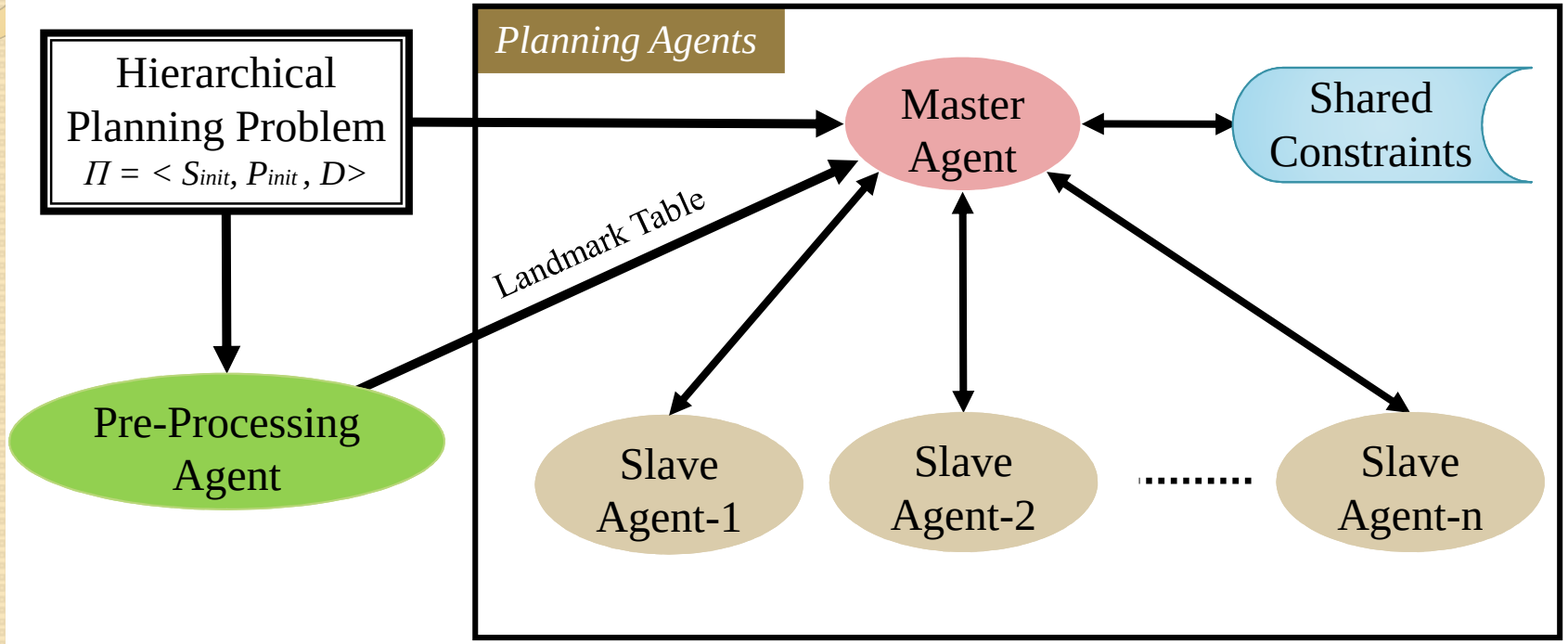


Motivation

- Development of novel technique to reduce the search effort and increase the performance of HTN planning systems.
  - Our technique integrates landmark preprocessing technique in the context of hierarchical planning with multi-agent planning ***and*** serves to decompose the original planning problem into a set of sub-problems each of which can then be solved separately using a multi-agent based planning approach.

Hybrid Multi-agent Planning

- Hybrid multi-agent planning architecture



Pre-Processing Agent
(Task Decomposition Tree & Landmark Table)

Figure: Task Decomposition Graph

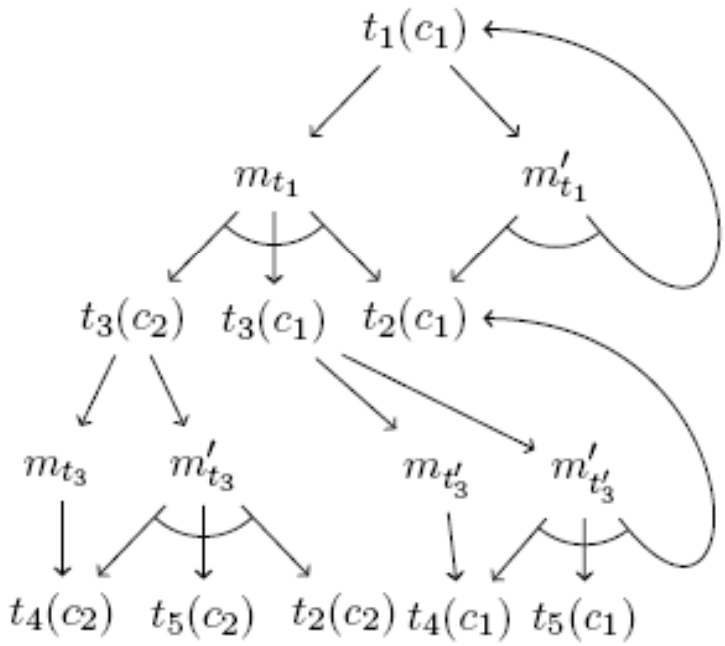


Table: Landmark Table

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:

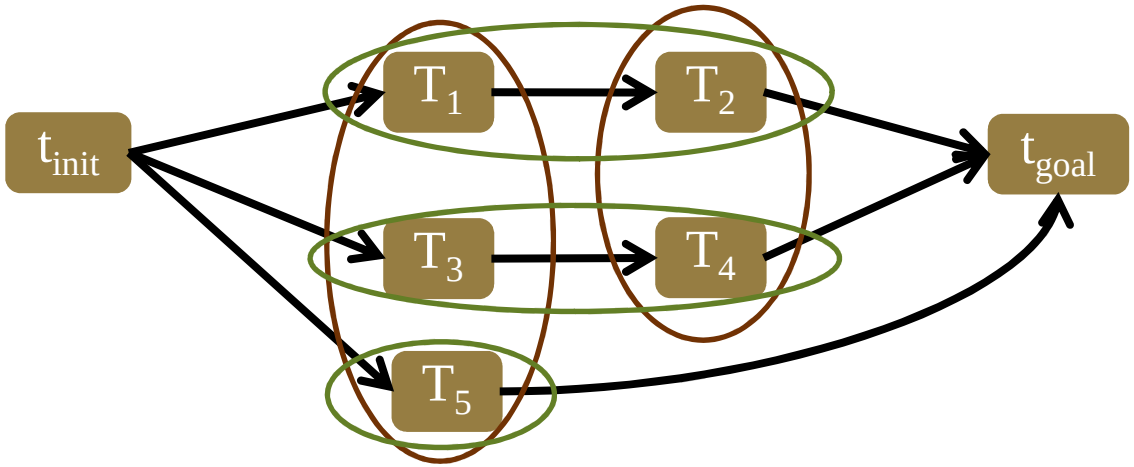
Intersection of method's plans

Optional Tasks:

Remaining tasks

Master Agent – Splitting Process

- Clustering algorithms
  - Dependent clustering algorithm : A set of dependent clusters
  - Independent clustering algorithm : A set of independent clusters



Dependent clustering algorithm

Cluster-1 : $\langle \{T_1, T_3, T_5\}, \{\} \rangle$

Cluster-2 : $\langle \{T_2, T_4\}, \{\} \rangle$

SC : $\{ T_1 < T_2 , T_3 < T_4 \}$

Independent clustering algorithm

Cluster-1 : $\langle \{T_1, T_2\}, \{T_1 < T_2\} \rangle$

Cluster-2 : $\langle \{T_3, T_4\}, \{T_3 < T_4\} \rangle$

Cluster-3 : $\langle \{T_5\}, \{\} \rangle$

SC : $\{\}$

Master Agent

- Initiates a set of slave agents based on the number of clusters.
- Distributes these clusters among slave agents :
  - Identical and non-cooperating.
  - Perform Hierarchical planning *(Refinement Planning Algorithm).*
  - Failure : act as master agent.
- Collects and merges solution plans.

Slave Agents - Search Procedure

Algorithm 1: Refinement Planning Algorithm

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

1 **while** $\texttt{Fringe} = \langle P_1 \dots P_n \rangle \neq \varepsilon$ **do**
2 $\quad F \leftarrow f^{\text{FlawDet}}(P_1)$
3 $\quad$ **if** $F = \emptyset$ **then return** P_1
4 $\quad \langle \mathtt{m}_1 \dots \mathtt{m}'_n \rangle \leftarrow f^{\text{ModOrd}}(\bigcup_{\mathtt{f} \in F} f^{\text{ModGen}}(\mathtt{f}))$
5 $\quad \texttt{succ} \leftarrow \langle apply(\mathtt{m}_1, P_1) \dots \ apply(\mathtt{m}'_n, P_1) \rangle$
6 $\quad \texttt{Fringe} \leftarrow f^{\text{PlanOrd}}(\texttt{succ} \circ \langle P_2 \dots P_n \rangle)$
7 **return** `fail`

Master Agent – Merging Process

➢Relies on the notion of ***Fragments***

- ❖ Individual plans : $P_\Gamma = \{p_{\gamma 1}, p_{\gamma 2}, \cdots, p_{\gamma n}\}$
- ❖ Ordering constraints in the Shared Constraints set (SC)
- ❖ $F = < P_\gamma, O_\gamma >$
  - ✓ P_γ represents a set of individual plans $(P_\gamma \subseteq P_\Gamma)$
  - ✓ O_γ imposes a partial order on P_γ
- ❖ Zero-Fragment and Related-Fragments
  - ✓ Zero-Fragment : includes those individual plans that are independent.
  - ✓ Related-Fragment : includes those individual plans that are dependent.

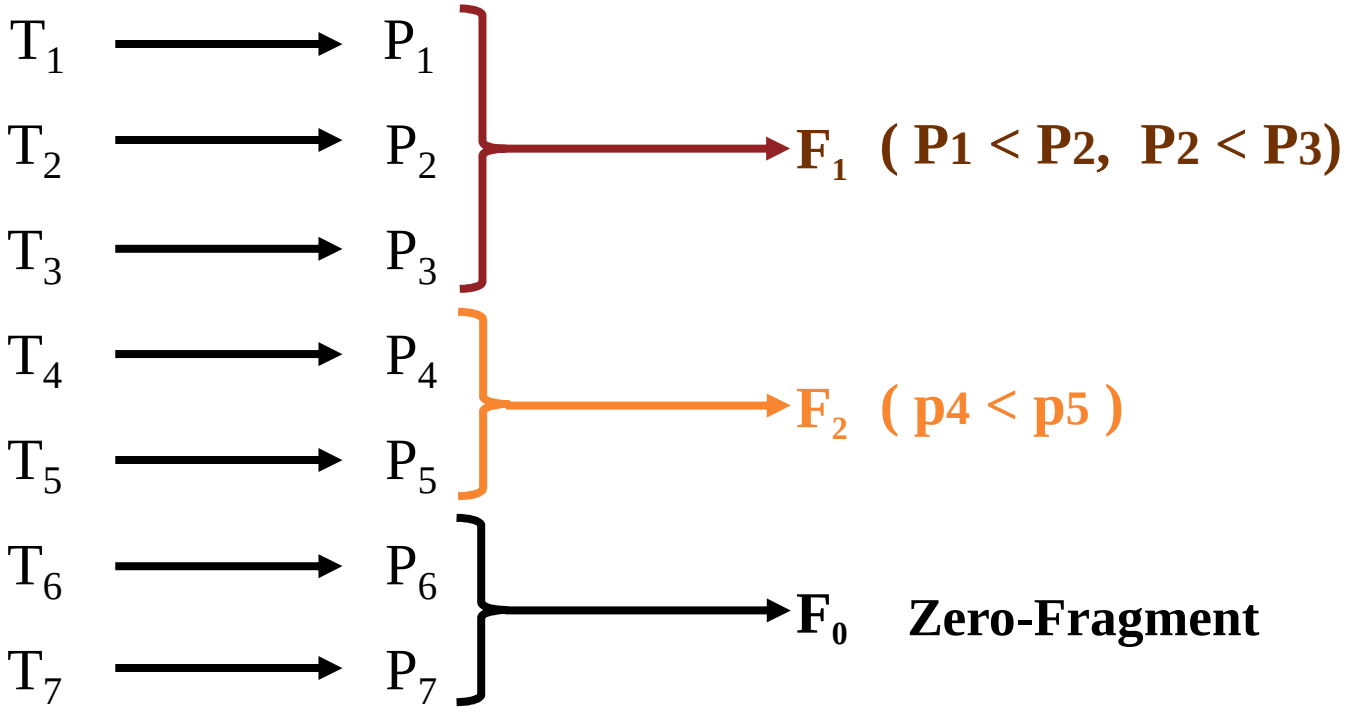
Master Agent – Merging Process

Example:

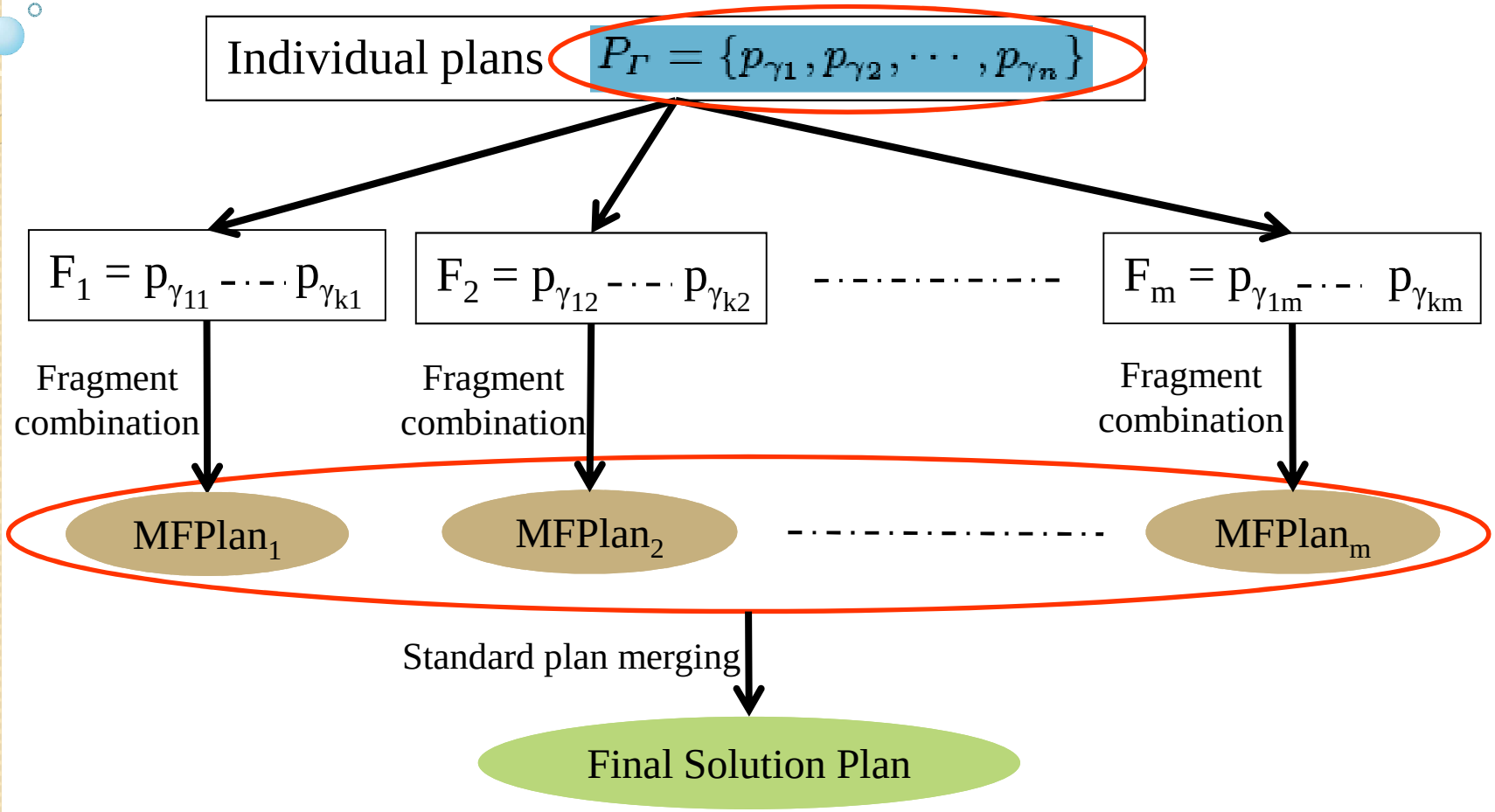
- Suppose Shared Constraints include the following ordering constraints:

$$T_1 < T_2 , \; T_2 < T_3, \; T_4 < T_5$$

and the set of individual plans for these tasks are:



Master Agent – Merging Process



Evaluation Benchmark

➢ We ran our evaluations over two distinguished benchmark domains:

- **UM-Translog Domain** describes scenarios of transporting various types of goods by various means (trucks, trains,…) via appropriate infrastructure (roads, transport centers,…).
- **Satellite Domain** manages scientific stellar observations by earth-orbiting instrument platforms .

Domain Name	Methods	Abstract tasks	Primitive tasks
UM-Translog	51	21	48
Satellite	8	3	5

Evaluation Benchmark

- **Evaluation factors**
  - *Search Space Size (SSS) :* The number of plans that are visited before obtaining the first solution.
  - *CPU time :* The total running time of the planning system in seconds.
- *Search space limit* **:** 5,000 plans
- *Time limit* **:** 9,000 seconds

Evaluation

- The average improvement of HMAP versus un-pruned and pruned planners.

Planner	*HMAP*	
	Dependent clustering	***Independent clustering***
UM-Translog		
Un-pruned	**72 %**	**79%**
Pruned	**48 %**	**50%**
Satellite		
Un-pruned	**74 %**	**78 %**
Pruned	**66 %**	**69 %**

- The average performance improvements will increase dramatically with the number of tasks in the initial plan.
- Our experiments indicate that when there is a causal interaction between tasks in the plan, the *independent clustering technique* is more efficient than the *dependent clustering technique*.

Summary

- Integrates the hierarchical landmark pre-processing technique with MAP.
- Enables to break up the planning problem into a set of clusters using two different techniques: Dependent and Independent.
- The set of slave agents work independently.
- The individually constructed plans are merged successfully in order to generate a global plan without additional refinement in individual plan.
- Our evaluations over a number of representative hierarchical planning domains and problems in which the HMAP approach competed with a planner “with and without” preprocessing.
- Results give evidence for the practical relevance of our approach.