Intention Interleaving Via Classical Replanning

Mengwei Xu, Kim Bauters, Kevin McAreavey, Weiru Liu

University of BRISTOL

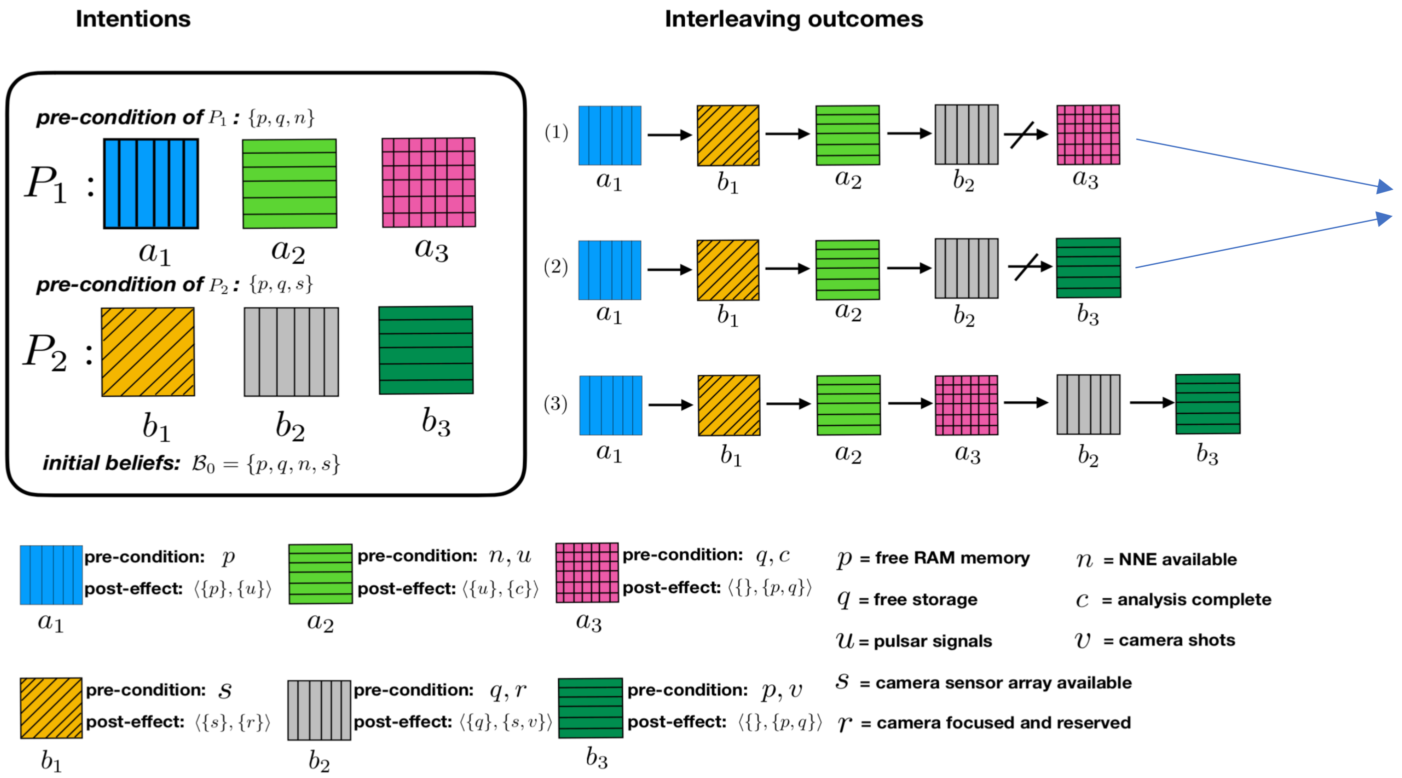
Extending Belief-Desire-Intention (BDI) Agents to Managing Intention Interleaving

- Intention Resolution: to avoid negative interference
  - **Guarantee the achievability of intentions when interleaving the steps in different intentions**

- Intention Merging: to facilitate positive interference
  - **Perform one task once for at least two goals, i.e. “kill two birds with one stone”**

Motivation to Manage Intention Interleaving

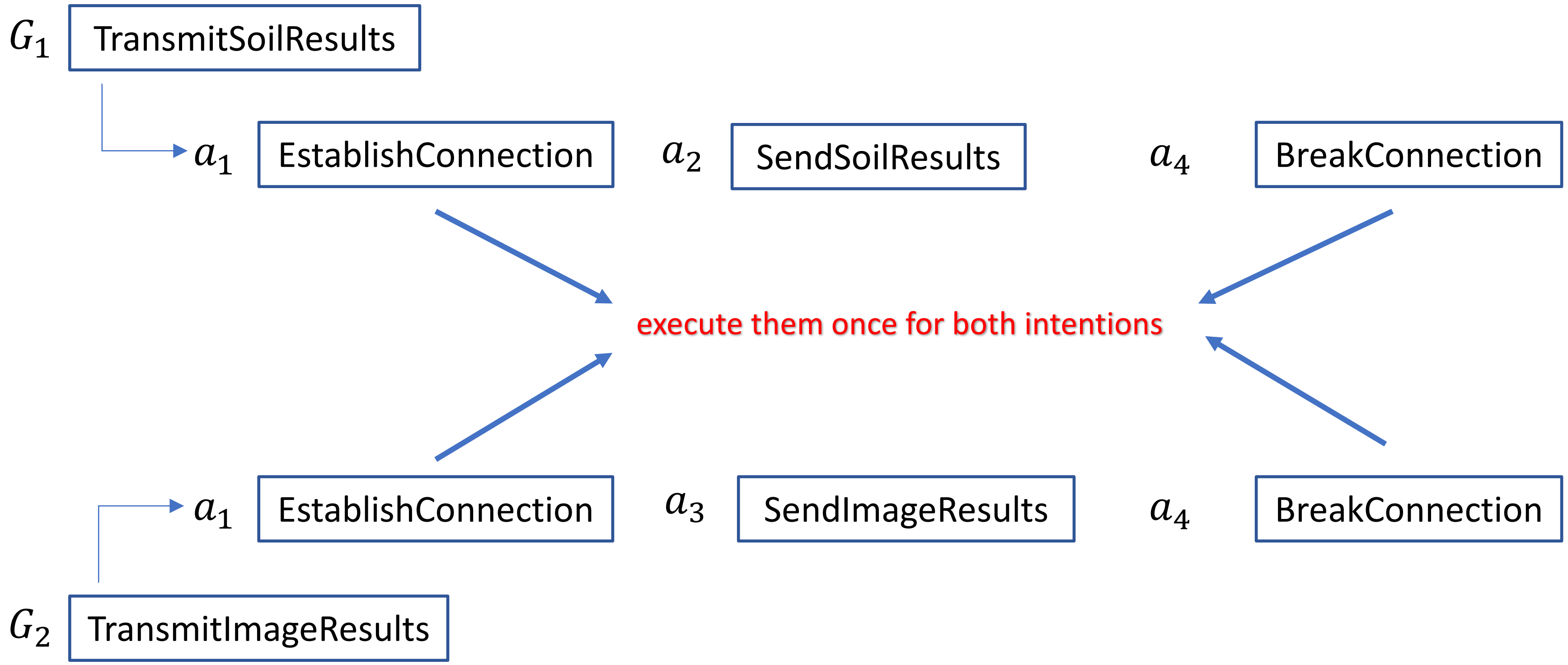
- Intention Resolution



Careless interleaving could result in that neither of its intention can be completed.

Motivation to Manage Intention Interleaving

➢ Intention Merging



Belief-Desire-Intention: Literature

Logics

[Cohen & Levesque, 1990]

[Rao & Georgeff, 1991]

[Shoham, 2009]

Programming Languages

AgentSpeak [Rao, 1996]

CAN [Winikoff et al., 2002]

CANPLAN [Sardina et al., 2011]

Software Platforms

Jason [Bordini et al., 2007]

Jack [Winikoff, 2005]

Jadex [Pokahr et al., 2013]

BDI Agent $(\mathcal{B}, \Lambda, \Pi)$

Initial belief base

Belief base specifying agent's initial beliefs

Action library

Set of STRIPS-style action descriptions

Plan library

Set of plan rules

BDI Agent $(\mathcal{B}, \Lambda, \Pi)$

Initial belief base

Belief base specifying agent's initial beliefs

Belief base $\mathcal{B} \subseteq \mathcal{L}$

Set of formulas from **logical language $\mathcal{L}$**

$\mathcal{B}$ must support:

- $\mathcal{B} \vDash \varphi$ (Entailment)
- $\mathcal{B} \cup \{\varphi\}$ (Addition)
- $\mathcal{B} \setminus \{\varphi\}$ (Deletion)

<u>Assume $\mathcal{B}$ is a set of atoms</u>

CAN: Agent $(\mathcal{B}, \Lambda, \Pi)$

Action library

Set of STRIPS-style action descriptions

Action description $\mathrm{act} : \varphi \leftarrow \mathcal{B}^- \, ; \, \mathcal{B}^+$

Primitive action symbol

Precondition $\varphi \in \mathcal{L}$

Set of "delete" atoms $\mathcal{B}^- \subseteq \mathcal{L}$

Set of "add" atoms $\mathcal{B}^+ \subseteq \mathcal{L}$

BDI Agent $(\mathcal{B}, \Lambda, \Pi)$

Plan library

Set of plan rules

$\boldsymbol{Head}(\boldsymbol{P})\text{: } G$

e.g. new goal

$\boldsymbol{Context}(\boldsymbol{P})\text{: } \varphi \in \mathcal{L}$

Formula from $\mathcal{L}$

body$(\boldsymbol{P})\text{: } h_1; \cdots; h_n$

e.g. a sequence of actions or goals

Plan rule $\mathrm{P} = G\text{: }\varphi \leftarrow h_1; \cdots; h_n$

BDI Operational Mechanism Sketch

Goal G $\xrightarrow{\textit{select}}$ **Relevant Plans** $\begin{array}{l} G : \varphi_1 \leftarrow P_1 \\ G : \varphi_2 \leftarrow P_2 \\ G : \varphi_3 \leftarrow P_3 \\ \quad\quad \vdots \\ G : \varphi_n \leftarrow P_n \end{array}$ $\xrightarrow{\textit{select}}$ **Applicable Plans** $\begin{array}{l} G : \varphi_{11} \leftarrow P_{11} \\ G : \varphi_{21} \leftarrow P_{21} \\ G : \varphi_{31} \leftarrow P_{31} \\ \quad\quad \vdots \\ G : \varphi_{n1} \leftarrow P_{n1} \end{array}$

$\subseteq \Pi$

where $\boldsymbol{\mathcal{B}} \vDash \varphi_{\boldsymbol{j1}}, j \in \{1, \cdots, n\}$

repeat for the subgoals

A tree structure representing all possible ways of achieving a goal G

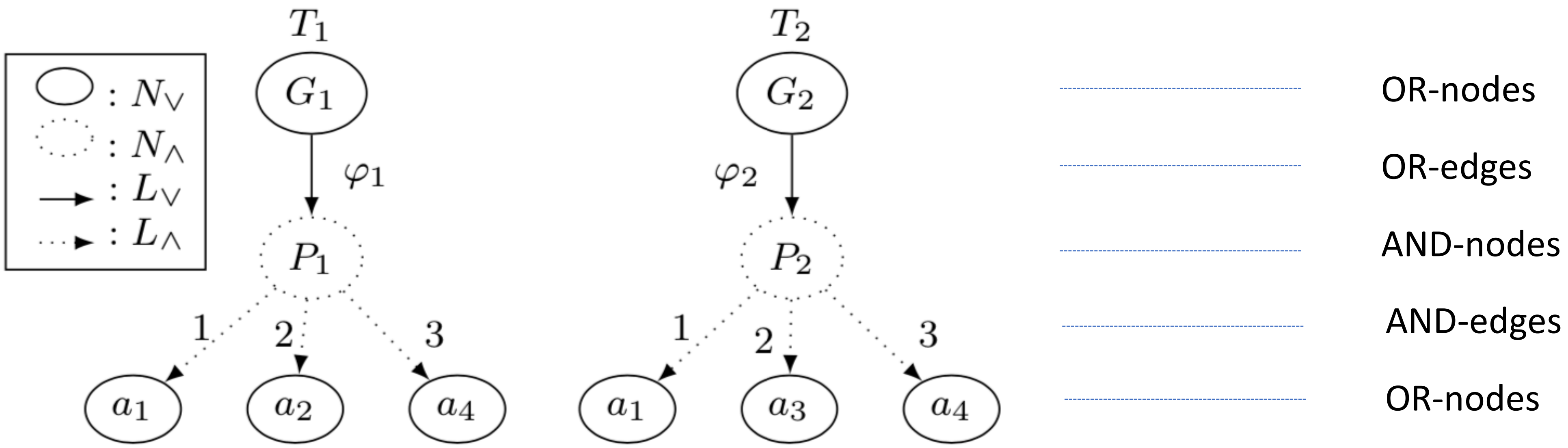
Our Intention Interleaving Framework in BDI

1. Intention Formalisation
   - Model an intention as an AND/OR graph
   - Define the execution trace for multiple intentions
   - Define the conflict-free and maximal-merged execution trace for multiple intentions
2. Intention Interleaving Planning Preparation
   - Indexing nodes
   - Defined terminal, initial node sets, and progression links of intentions
   - Computing overlapping programs between multiple intentions
3. Intention Interleaving Planning Formalism
   - Formalise FPP problem of interleaving intentions
   - Correctness Proof
4. Implementation
5. Evaluation

AND/OR Graphs for Intentions

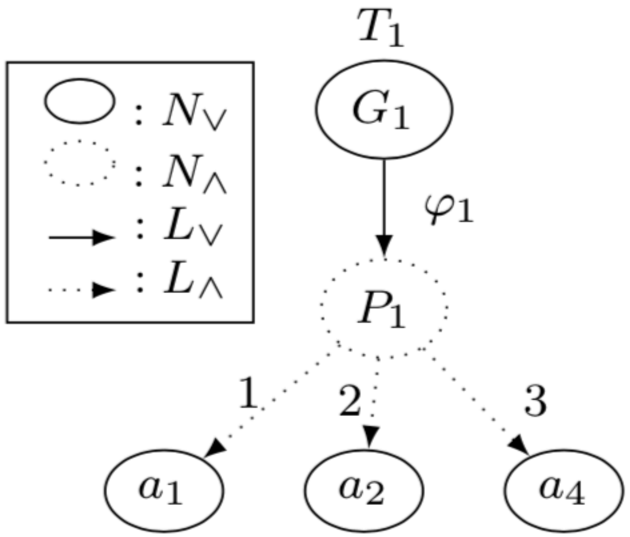
$P_1 = G_1 \colon \varphi_1 \leftarrow a_1; a_2; a_4$

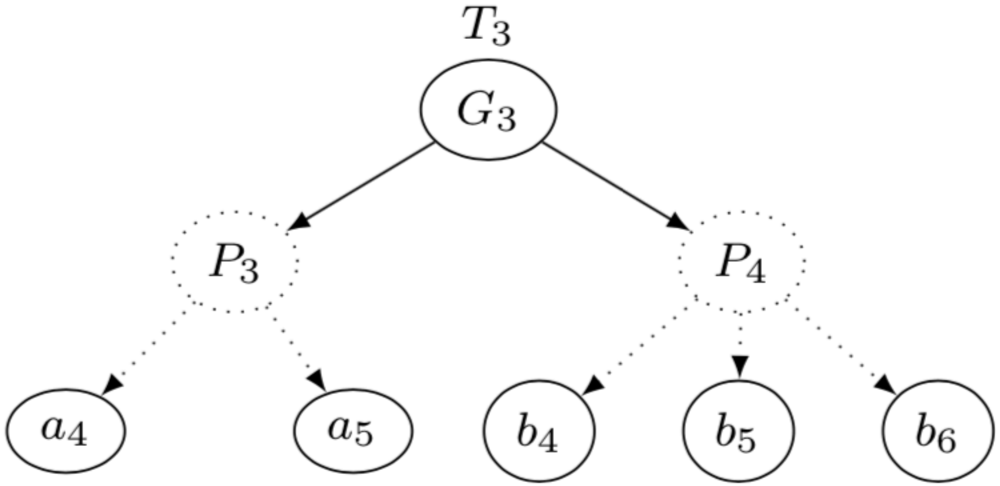
$P_2 = G_2 \colon \varphi_2 \leftarrow a_1; a_3; a_4$



Execution Trace for An Intention

To identifies every unique way in which a given intention can be achieved





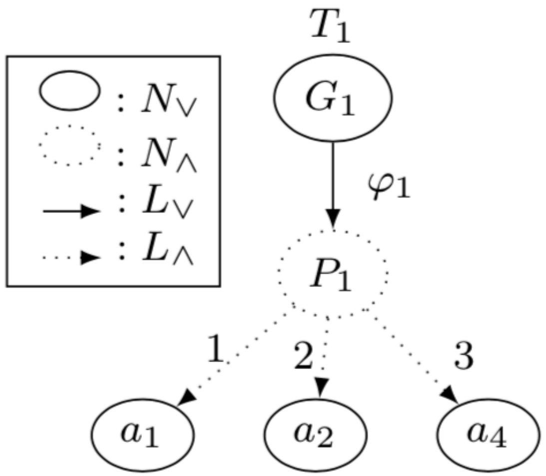
Execution trace for T_1: $\tau(T_1) = G_1; P_1; a_1 ; a_2 ; a_4$

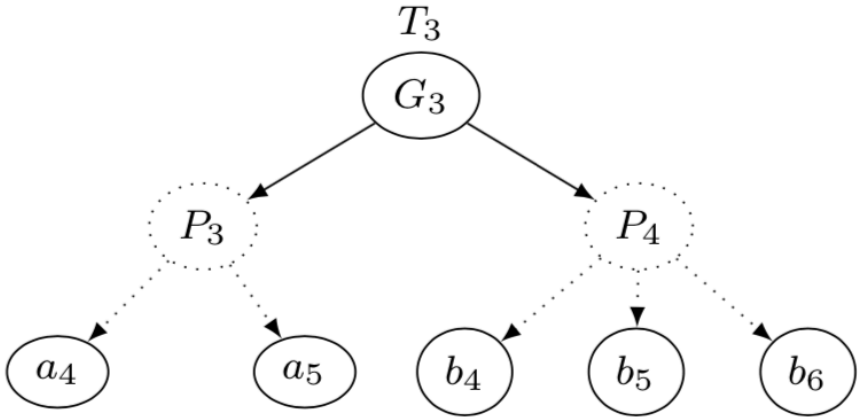
Execution trace for T_3: $\tau_1(T_3) = G_3; P_3; a_4; a_5$

$\tau_2(T_3) = G_3; P_4; b_4; b_5; b_6$

Execution Trace for Multiple Intentions

The construction of an execution trace of a set of intentions is to interleave elements in the execution traces of different intentions





Potential execution trace for T_1 and T_3: $\sigma = \boldsymbol{G_1}; \boldsymbol{P_1}; G_3; P_3; \boldsymbol{a_1}; a_4\ ; \boldsymbol{a_2}\ ; \boldsymbol{a_4}\ ; a_5$

by interleaving $\tau(T_1) = \boldsymbol{G_1}; \boldsymbol{P_1}; \boldsymbol{a_1}\ ; \boldsymbol{a_2}\ ; \boldsymbol{a_4}$ and $\tau_1(T_3) = G_3; P_3; a_4; a_5$

Execution Trace for Intentions (Cont.)

Conflict-free Execution Trace:

To model the successful interleaving which achieves all intentions

$$\sigma[1] \quad \sigma[2] \quad \cdots \quad \sigma[j-1] \quad \sigma[j] \quad \sigma[j+1] \quad \cdots \quad \sigma[n]$$

$$\mathcal{B}_1 \quad \mathcal{B}_2 \quad \mathcal{B}_{j-1} \quad \mathcal{B}_j \quad \mathcal{B}_{j+1} \quad \mathcal{B}_n$$

where $\mathcal{B}_j$ is the belief base before the execution of the j^{th} element of an execution trace (i.e. $\sigma[j]$)

An execution trace σ is conflict-free if and only if the following hold:

1. if $\sigma[j] = P \in \Pi$, then $\mathcal{B}_j \vDash context(P)$, i.e. the context of plan P must be met before selection
2. if $\sigma[j] = a \in \Lambda$, then $\mathcal{B}_j \vDash \psi(a)$, i.e. the pre-condition of action \`a' must be met before selection

Execution Trace for Intentions (Cont.)

Mergeable Execution Trace of $\{T_1, \cdots, T_m\}$

To capture the overlapping programs of different intentions

$$\boldsymbol{\sigma}: \quad \sigma[1] \quad \sigma[2] \quad \cdots \quad \underbrace{\sigma[j] \quad \sigma[j+1] \quad \cdots \quad \sigma[j+k-1] \quad \sigma[j+k]}_{} \quad \cdots \quad \sigma[n]$$

k consecutive same element from all difference intentions in σ

$$\boldsymbol{\sigma^m}: \quad \sigma[1] \quad \sigma[2] \quad \cdots \quad \sigma[j] \quad \sigma[j+k+1] \cdots \quad \sigma[n]$$

An execution trace σ is a mergeable execution trace if and only if the following hold:

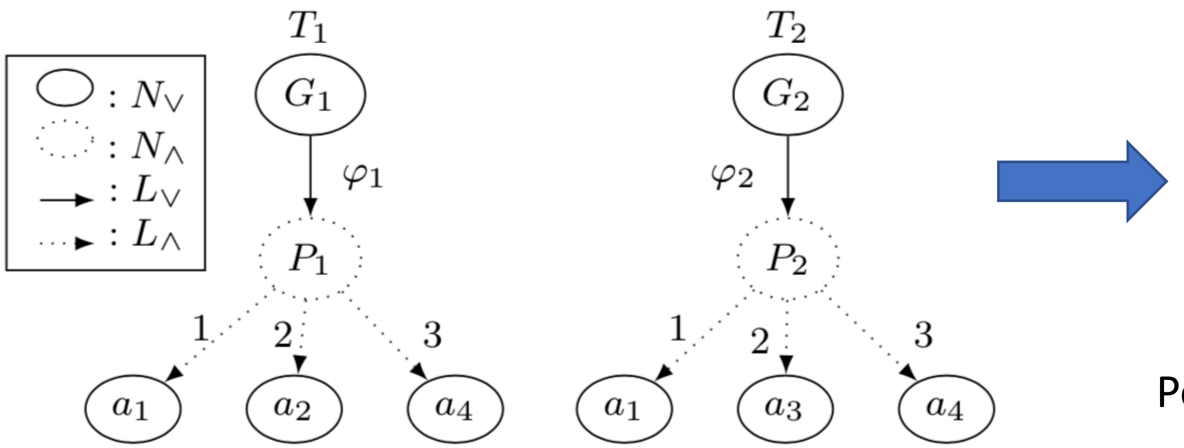
1. $\exists j \in \{1, \cdots, n\}$ such that $\sigma[j] = \sigma[j+1] = \cdots \sigma[j+k]$ where $2 \leq k \leq n - j$;
2. $\forall l \in \{1, \cdots, m\}$, $\nexists s, t \in \{j, \cdots, j+k\}$ where $s \neq t$ such that $\sigma[s] \subseteq \tau(T_l) \subseteq \sigma$ and $\sigma[t] \subseteq \tau(T_l) \subseteq \sigma$;
3. σ^m is a conflict-free execution trace where σ^m is the merged execution trace of σ by reducing each subsequence consisting of consecutive identical elements characterized by 1 and 2 in σ to only one element left.

Execution Trace for Intentions (Cont.)

Maximal-merged Trace of $\{T_1, \cdots, T_m\}$

To capture the most merged execution trace of multiple intentions

The merged execution trace σ^m of a mergeable execution trace σ of $\{T_1, \cdots, T_m\}$ is maximal-merged if there is no another mergeable execution trace σ' of $\{T_1, \cdots, T_m\}$ such that $|\sigma'^m| < |\sigma^m|$ where $|\sigma|$ stands for the length of σ.



the potential maximal-merged trace of $\{T_1, T_2\}$

$$\sigma^m = G_1; P_1; G_2; P_2; a_1 \,; a_2 \,; a_3 \,; a_4$$

Perform action a_1 and a_4 once for both two goals T_1 and T_2

Indexing Nodes

To ensure that e.g. the same actions in distinct plans is seen as different

A node n is a top-level goal of intention T: $T(\bar{n})$

The nodes of actions and subgoals of intention T: $n^{P,j,T}$ to denote the j^{th} member of $body(P)$ in T

A plan node in intention T: n^{T}

Initial node set for intentions $\{T_1, \cdots, T_m\}$: $z_0 = \{T_1(\bar{n}), \cdots, T_m(\bar{n})\}$

Terminal node set for a goal node: a collection of the last element of each execution trace of a goal

Terminal node set for intentions $\mathrm{I} = \{T_1, \cdots, T_m\}$

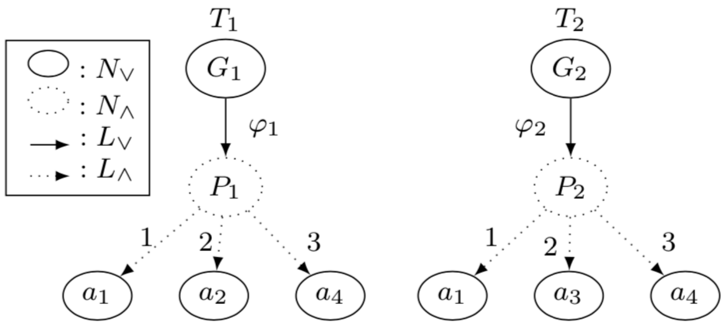
$z_g = \{tn_1, \cdots, tn_m\}$ where tn_i is a terminal node of $T_i(\bar{n})$

$z_g \rhd_{tn} I$ if z_g is a terminal node set of I

$$\tau(T_1) : \begin{pmatrix} node & G_1 & P_1 & a_1 & a_2 & a_4 \\ index & T_1(\bar{n}) & P_1^{T_1} & a_1^{P_1,1,T_1} & a_2^{P_1,2,T_1} & a_4^{P_1,3,T_1} \end{pmatrix}$$

initial node *terminal node*

$$\tau(T_2) : \begin{pmatrix} node & G_2 & P_2 & a_1 & a_3 & a_4 \\ index & T_2(\bar{n}) & P_2^{T_2} & a_1^{P_2,1,T_2} & a_3^{P_2,2,T_2} & a_4^{P_2,3,T_2} \end{pmatrix}$$



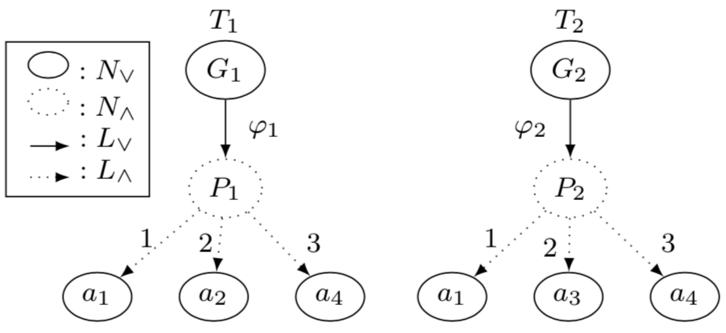
Progression Links

To visualise the progression order of execution elements in the context of indexes

$$\tau(T_1): \begin{pmatrix} node & G_1 & P_1 & a_1 & a_2 & a_4 \\ index & T_1(\bar{n}) & P_1^{T_1} & a_1^{P_1,1,T_1} & a_2^{P_1,2,T_1} & a_4^{P_1,3,T_1} \end{pmatrix}$$

initial node   *terminal node*

$$\tau(T_2): \begin{pmatrix} node & G_2 & P_2 & a_1 & a_3 & a_4 \\ index & T_2(\bar{n}) & P_2^{T_2} & a_1^{P_2,1,T_2} & a_3^{P_2,2,T_2} & a_4^{P_2,3,T_2} \end{pmatrix}$$



The progression links of execution trace $\tau(T_1)$

$(T_1(\bar{n}) \rightarrow P_1{}^{T_1})$

$(P_1{}^{T_1} \rightarrow a_1{}^{P_1,1,T_1})$

$(a_1{}^{P_1,1,T_1} \rightarrow a_2{}^{P_1,2,T_1})$

$(a_2{}^{P_1,2,T_1} \rightarrow a_4{}^{P_1,3,T_1})$

They are also called primitive progression links

The progression links of execution trace $\tau(T_2)$

$(T_2(\bar{n}) \rightarrow P_2{}^{T_2})$

$(P_2{}^{T_2} \rightarrow a_1{}^{P_2,1,T_2})$

$(a_1{}^{P_2,1,T_2} \rightarrow a_3{}^{P_2,2,T_2})$

$(a_3{}^{P_2,2,T_2} \rightarrow a_4{}^{P_2,3,T_2})$

Overlap Set of Multiple Intentions

To compute all potential overlapping programs among a set of intentions

The overlap set of $\{T_1, \cdots, T_m\}$ is a set of tuples of the form $\langle (idx_b^1 \rightarrow idx_e^1), \cdots, (idx_b^k \rightarrow idx_e^k) \rangle$ if:

1. $\mathrm{J}(idx_e^1) = \cdots = \mathrm{J}(idx_e^k)$ where $\mathrm{J}(idx_e^i)$ represents the actual node of the ending index idx_e^i;
2. $\forall l \in \{1, \cdots, m\}, \nexists s, t \in \{j, \cdots, j+k\}$ where $s \neq t$ s.t. $(idx_b^s \rightarrow idx_e^s) \in \tau(T_l)$ and $(idx_b^t \rightarrow idx_e^t) \in \tau(T_l)$;

The progression links of execution trace $\tau(T_1)$

$(T_1(\bar{n}) \rightarrow P_1{}^{T_1})$ $\quad$ $(P_1{}^{T_1} \rightarrow \boldsymbol{a_1}{}^{\boldsymbol{P_1,1,T_1}})$

$(a_1{}^{P_1,1,T_1} \rightarrow a_2{}^{P_1,2,T_1})$ $\quad$ $(a_2{}^{P_1,2,T_1} \rightarrow a_4{}^{P_1,3,T_1})$

The progression links of execution trace $\tau(T_2)$

$(T_2(\bar{n}) \rightarrow P_2{}^{T_2})$ $\quad$ $(P_2{}^{T_2} \rightarrow \boldsymbol{a_1}{}^{\boldsymbol{P_2,1,T_2}})$

$(a_1{}^{P_2,1,T_2} \rightarrow a_3{}^{P_2,2,T_2})$ $\quad$ $(a_3{}^{P_2,2,T_2} \rightarrow a_4{}^{P_2,3,T_2})$

The overlap set of intention $\{T_1, T_2\}$ has two elements as follows:

1. $\langle (P_1{}^{T_1} \rightarrow \boldsymbol{a_1}{}^{\boldsymbol{P_1,1,T_1}}), (P_2{}^{T_2} \rightarrow \boldsymbol{a_1}{}^{\boldsymbol{P_2,1,T_2}}) \rangle$ where $\mathrm{J}(a_1{}^{P_1,1,T_1}) = \mathrm{J}(a_1{}^{P_2,1,T_2}) = a_1$;
2. $\langle (a_2{}^{P_1,2,T_1} \rightarrow \boldsymbol{a_4}{}^{\boldsymbol{P_1,3,T_1}}), (a_3{}^{P_2,2,T_2} \rightarrow \boldsymbol{a_4}{}^{\boldsymbol{P_2,3,T_2}}) \rangle$ where $\mathrm{J}(a_4{}^{P_1,3,T_1}) = \mathrm{J}(a_4{}^{P_2,3,T_2}) = a_4$

Overlap Progression Links

Let an element of overlap set of $\{T_1, \cdots, T_m\}$ be $\langle (idx_b^1 \to idx_e^1), \cdots, (idx_b^k \to idx_e^k) \rangle$.

Then we have a corresponding overlap progression link $\alpha^o = (\{idx_b^1, \cdots, idx_b^k\} \to \{idx_e^1, \cdots, idx_e^k\})$

such that the side of α^o is $size(\alpha^o) = \mathrm{k} - 1$, i.e. merging $k - 1$ extra primitive progression links.
by fault, the size of a primitive progression link α^p is $size(\alpha^o) = 0$, i.e. no merging at all.

The overlap set of intention $\{T_1, T_2\}$ has two elements as follows:

1. $\langle (P_1{}^{T_1} \to a_1{}^{P_1,1,T_1}), (P_2{}^{T_2} \to a_1{}^{P_2,1,T_2}) \rangle$ $\Rightarrow$ $(\{P_1{}^{T_1}, P_2{}^{T_2}\} \to \{a_1{}^{P_1,1,T_1}, a_1{}^{P_2,1,T_2}\})$
2. $\langle (a_2{}^{P_1,2,T_1} \to a_4{}^{P_1,3,T_1}), (a_3{}^{P_2,2,T_2} \to a_4{}^{P_2,3,T_2}) \rangle$ $\Rightarrow$ $(\{a_2{}^{P_1,2,T_1}, a_3{}^{P_2,2,T_2}\} \to \{a_4{}^{P_1,3,T_1}, a_4{}^{P_2,3,T_2}\})$

Intention Interleaving Planning Formalism

A First-principles Planning (FPP) problem of interleaving intentions $I = \{T_1, \cdots, T_m\}$ is a tuple

$$\Omega = \langle \Sigma, X, O, s_0, S_G \rangle$$

a finite set of (propositional) atoms

$X = \bigcup_{j=1}^{m} T_j(N_\vee \cup N_\wedge)$ is the set of node indexes of I

$O = O^p \cup O^o$ is a set of progression links

where O^p (resp. O^o) is the collection of primitive (resp. overlap) progression links

$s_0 = \mathcal{B}_0 \cup z_0$ is the initial state

where $\mathcal{B}_0$ is the initial belief base and z_0 is the initial node set of I

$S_G = \{z_g | z_g \rhd_{tn} I\}$ is the goal state

where z_g is the terminal node set of I

Intention Interleaving Planning Formalism (Cont.)

A FPP problem of interleaving intentions $I = \{T_1, \cdots, T_m\}$ is a tuple

$$\Omega = \langle \Sigma, X, O, s_0, S_G \rangle$$

$$O = O^p \cup O^o$$

STRIPS PROGRESSION LINKS

link α^p	$pre(\alpha^p)$	$del(\alpha^p)$	$add(\alpha^p)$
$(idx_b \rightarrow P^T)$	$idx_b \cup \varphi$	$\{idx_b\}$	$\{P^T\}$
$(idx_b \rightarrow a^{P,j,T})$	$idx_b \cup \psi(a^{P,j,T})$	$\phi^- \cup \{idx\}$	$\phi^+ \cup \{a^{P,j,T}\}$
$(idx_b \rightarrow G^{P,j,T})$	idx_b	$\{idx\}$	$\{G^{P,j,T}\}$

$$\alpha^o = \left(\{idx_b^1, \cdots, idx_b^k\} \rightarrow \{idx_e^1, \cdots, idx_e^k\}\right) \in O^o$$

in which $\alpha_i^p = \left(idx_b^i \rightarrow idx_e^i\right) \in O^p$

- $pre(\alpha^o) = pre(\alpha_1^p) \cup \cdots \cup pre(\alpha_k^p)$
- $del(\alpha^o) = del(\alpha_1^p) \cup \cdots \cup del(\alpha_k^p)$
- $add(\alpha^o) = add(\alpha_1^p) \cup \cdots \cup add(\alpha_k^p)$

Intention Interleaving Planning Formalism (Cont.)

A FPP problem of interleaving intentions $I = \{T_1, \cdots, T_m\}$ is a tuple $\Omega = \langle \Sigma, X, O, s_0, S_G \rangle$

Definition 1: The result of applying a progression link $\alpha \in O$ to a state $s = \mathcal{B} \cup z$ is described by the transition function $f: 2^{\Sigma} \cup 2^{X} \times O \rightarrow 2^{\Sigma} \cup 2^{X}$ defined as follows:

$$f(s, \alpha) = \begin{cases} (s \backslash \mathrm{del}(\alpha)) \cup add(\alpha) & if\ s \vDash pre(\alpha) \\ undefined & otherwise \end{cases}$$

Definition 2: The result of applying a sequence of progression links to a state specification s is defined inductively:

$$\mathrm{Res}(\mathrm{s}, \langle \quad \rangle) = s$$

$$\mathrm{Res}(\mathrm{s}, \langle \alpha_0; \cdots; \alpha_n \rangle) = Res(f(s, \alpha_0), \langle \alpha_1; \cdots; \alpha_n \rangle)$$

Definition 3: A sequence of progression links $\Delta = \langle \alpha_0; \alpha_1; \cdots; \alpha_n \rangle$ is a solution to a FPP problem $\Omega = \langle \Sigma, X, O, s_0, S_G \rangle$, denoted as $\Delta = sol(\Omega)$, iff $\mathrm{Res}(\mathrm{s}, \Delta) \vDash S_G$. We also say that Δ is optimal if the sum of the size of the progression link $size(\alpha_i)$ is maximum where $i = 0, \cdots, n$.

Theorem: we have a maximal-merged trace σ^m of intention $I = \{T_1, \cdots, T_m\}$ if and only if there exists an optimal solution Δ to Ω.

Intention Interleaving Planning Formalism (Cont.)

A FPP problem of interleaving intentions $I = \{T_1, \cdots, T_m\}$ is a tuple $\Omega = \langle \Sigma, X, O, s_0, S_G \rangle$

Algorithm 1: Intention Interleaving Replanning

To adapt to the dynamic environment

Input: Planning problem $\Omega = \langle \Sigma, X, O, s_0, S_G \rangle$

1 $\alpha_0; \ldots; \alpha_n \leftarrow sol(\Omega)$ `/* FPP solution */`

2 $i \leftarrow 0, \alpha \leftarrow \alpha_0, s \leftarrow s_0$ `/* initialisation */`

3 **while** $s \notin \Upsilon$ **do**

4   **if** $f(s, \alpha) = undefined$ **then**

5         $idx_b \leftarrow \text{BEGINNING-INDEX}(\alpha)$

6         $G \leftarrow \text{BACKTRACK}(idx_b)$ `/* backtrack */`

7         $s_0 \leftarrow \mathcal{B} \cup z \setminus \{idx_b\} \cup \{G\}$ `/* modify state */`

8         $sol'(\Omega) \leftarrow \text{FPP}(\langle \Sigma, X, O, s_0, S_G \rangle)$ `/* replan */`

9         $\alpha_0; \ldots; \alpha_n \leftarrow sol'(\Omega)$

10         $\alpha \leftarrow \alpha_0, i \leftarrow 0$ `/* re-initialisation */`

11   EXECUTE α

12     $s \leftarrow f(s, \alpha)$

13     $i \leftarrow i + 1$

14     $\alpha \leftarrow \alpha_{i+1}$

line 5-7 instruct the procedures for failure backtracking and initial node state modification

Implementation

Operator Files
(containing progression links)

Planning Domain Definition Language (PDDL)

Fact Files
(containing initial/goal state description)

primitive progression links:

(:action $(idx_b \rightarrow P^T)$
:precondition (and idx_b $context(P)$)
:effect (and (not idx_b) P^T))
(:action $(idx_b \rightarrow a^{P,j,T})$
:precondition (and idx_b $\psi(a^{P,j,T})$)
:effect (and (not ϕ^-) ϕ^+ (not idx_b) $a^{P,j,T}$))
(:action $(idx_b \rightarrow G^{P,j,T})$
:precondition idx_b
:effect (and (not idx_b) $G^{P,j,T}$))

overlap progression links:

(:action $(\{idx_b^1, \ldots, idx_b^k\} \rightarrow \{idx_e^1, \ldots, idx_e^k\})$)
:precondition (and $pre(\alpha_1^p) \ldots, pre(\alpha_k^p)$)
:effect (and $add(\alpha_1^p) \ldots add(\alpha_k^p)$
(not $del(\alpha_1^p)$) ... (not $del(\alpha_k^p)$)
(increase (efficiency-utility) $size(\alpha^o)$)))

(:objects $\forall x \in X, \forall$ BELIEF_ATOMS $\in \Sigma$)

declare all objects in the plan problem instance

(:init $\mathcal{B}_0, \forall T \in I, T(\bar{n})$)

initial belief base and the top-level goals of intentions

(:goal (and (or $tn_1^1 \ldots tn_{k_1}^1$) ... (or $tn_1^m \ldots tn_{k_m}^m$))

reach any terminal node of each intention

Evaluation: A Manufacturing Scenario

		Operation 1	Operation 2	Operation 3
block 1		twisting-drilling 10cm	reaming	boring
block 2		twisting-drilling 15cm	reaming	boring
block 3		twisting-drilling 20cm	reaming	boring
block 4		twisting-drilling 25cm	reaming	boring

EFFECTIVENESS ANALYSIS OF APPROACH

	2.1	2.2	3.1	3.2	3.3	4.1	4.2	4.3	4.4
2	17%	33%	11%	22%	33%	8%	17%	25%	33%
3	22%	44%	15%	30%	44%	11%	22%	33%	44%
4	25%	50%	17%	33%	50%	13%	25%	38%	50%
5	27%	53%	18%	36%	53%	13%	27%	40%	53%
6	28%	56%	19%	37%	56%	14%	28%	42%	56%
7	29%	57%	19%	38%	57%	14%	29%	43%	57%
8	29%	58%	19%	39%	58%	15%	29%	44%	58%

Details can be found in my github

https://github.com/Mengwei-Xu/manufacturing-evaluation

Summary:

1. Formalise an intention as AND/OR graph
2. Formalise the conflict-free execution trace of multiple intentions
3. Formalise the maximal-merged execution trace of multiple intentions
4. Define the concept of overlapping programs between different intentions
5. Both formally and practically compile the intention interleaving problem into a planning problem
6. Provide a preliminary evaluation of a planning-centric intention interleaving problem

Future Work:

1. A complete algorithm of computing overlap set of intentions

2. Further test the costs and benefits of our approach empirically in a wider range of applications

3. Investigate the collaboration between multi-BDI agents, e.g. how to discover and exploit collaboration opportunities