Handling Planning Failures with Virtual Actions

Jicheng Fu[1], Sijie Tian[1], Vincent Ng[2], Farokh Bastani[2], and I-Ling Yen[2]

[1]The University of Central Oklahoma
[2]The University of Texas at Dallas

Introduction

- Artificial intelligence (AI) planning
    - Seeks to generate a plan of actions that leads the system from the initial state to the goal
    - Declarative and goal-oriented
        - Enables users to focus on “what to do”
        - AI planner automatically determines “how to get it done”

- Hence, AI planning has been widely used in many fields
    - e.g., intelligent agents, autonomous robots, web service compositions, etc.

Issues

- When given a problem to solve,
  - The AI planner either returns a solution if one exists
  - Or reports that no solution is found
    - Leaves no clues for people to trace the causes of the planning failure
- The practicality of AI planning
  - Depends heavily on the completeness of the planning domains
    - In reality, planning domains are not always complete
    - E.g., AI planning is widely used in automated web service composition
      - It is unrealistic to assume that all necessary services are available in the Internet
    - Incomplete domains constantly result in planning failures

Goal of the Study

- Propose virtual actions in the event of planning failure
- Virtual actions enable traditional planners to succeed
  - Hence, return an incomplete plan instead of merely an error message
- The specifications of the virtual actions suggest what the missing parts may contain
  - Providing important clues to users as to the nature of the failure

General Algorithm

- In the event of planning failure
  - Step1: Forward planning
    - Start from the initial state and proceed as far as possible towards the goal until it reaches the farthest place p_f
  - Step 2: Backward planning
    - Start from the goal and proceed as far as possible towards the initial state until it reaches the farthest place p_b
  - Step 3:
    - Propose a virtual action to enable both directions to succeed
- Two questions naturally arise
  - How to determine the farthest place p_f from the initial state or the farthest place p_b from the goal?
  - How to create a virtual action to enable both planning directions to succeed?

Question 1

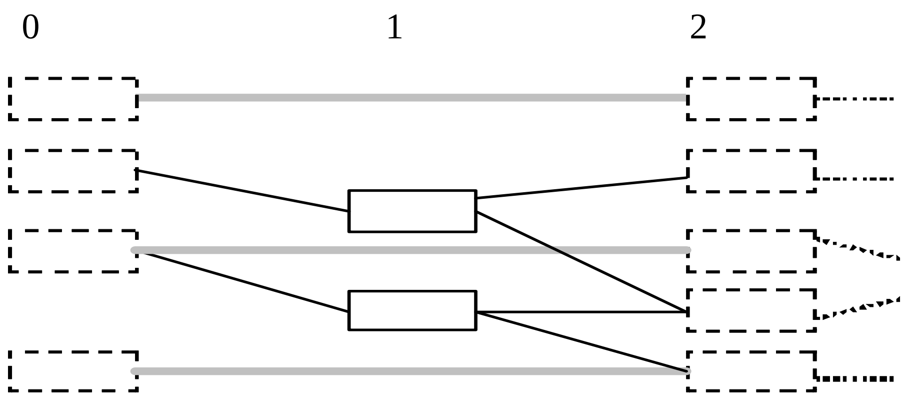
- **How to determine the farthest place from the initial state or the farthest place from the goal?**

  Use the planning graph's intrinsic feature --- level-off

- Planning graph

  Directed, leveled graph consisting of proposition and action nodes arranged in levels

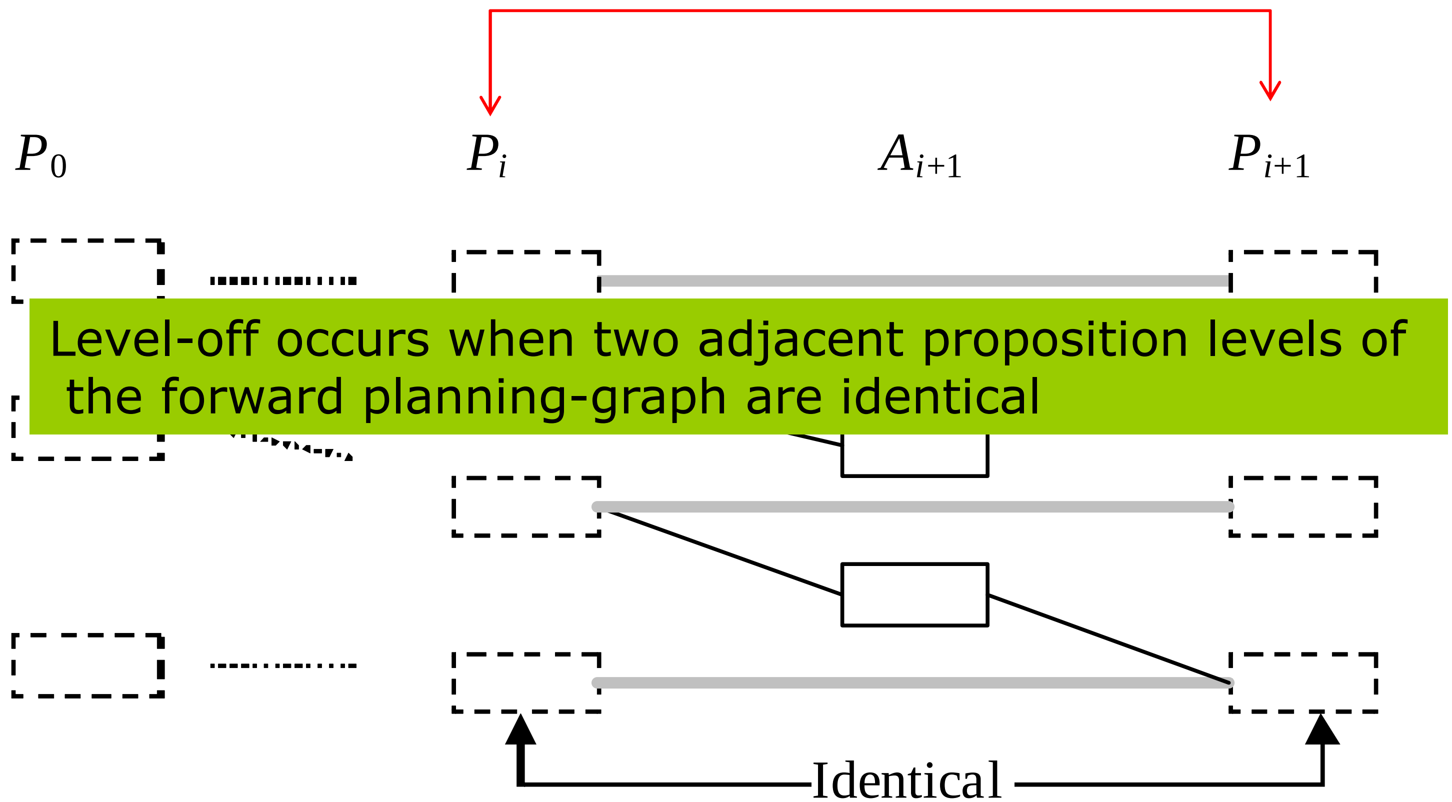
  - Even-numbered levels contain proposition nodes
  - Odd-numbered levels contain action nodes



Level-off

- **How can we determine the farthest place from the initial state or the farthest place from the goal?**

  Use the planning graph's intrinsic feature --- level-off



Question 1 (cont.)

- **How can we determine the farthest place from the initial state or the farthest place from the goal?**

  Use the planning graph's intrinsic feature --- level-off

  Level-off occurs when all the possible actions have been applied to the planning graph but the goal condition still cannot be reached

  - **The proposition level at which level-off occurs represents the farthest level from the initial state**

- How about the backward planning?

Deterministic Planning: the Focus of This Study

Definition 1. A **deterministic** planning domain is a 4-tuple $\Sigma = \langle P, S, A, \gamma \rangle$, where:

- P is a finite set of propositions;
- $S \subseteq 2^P$ is a finite set of states in the system;
- A is a finite set of actions; and
- $\gamma: S \times A \rightarrow S$ is the state-transition function.

- An action a in Σ consists of a precondition, $pre(a)$, and an effect, $eff(a)$.
  - $eff(a)$ is composed of two parts: the add effect and the delete effect
  - For example, the action "move(A, B)" will generate
    - the add effect of the robot being at B and
    - the delete effect is the robot being at A.

How to Deal with Backward Planning?

- **Definition 2.** A **planning problem** is a triple $\langle s_0, g, \Sigma\rangle$, where s_0 is the initial state, g is the goal condition, and Σ is the planning domain
- **Backward planning:** we construct the planning graph based on the reversed planning problem $\langle g, s_0, \sum^{-1}\rangle$, where

  g serves as the initial state;

  s_0 serves as the goal; and

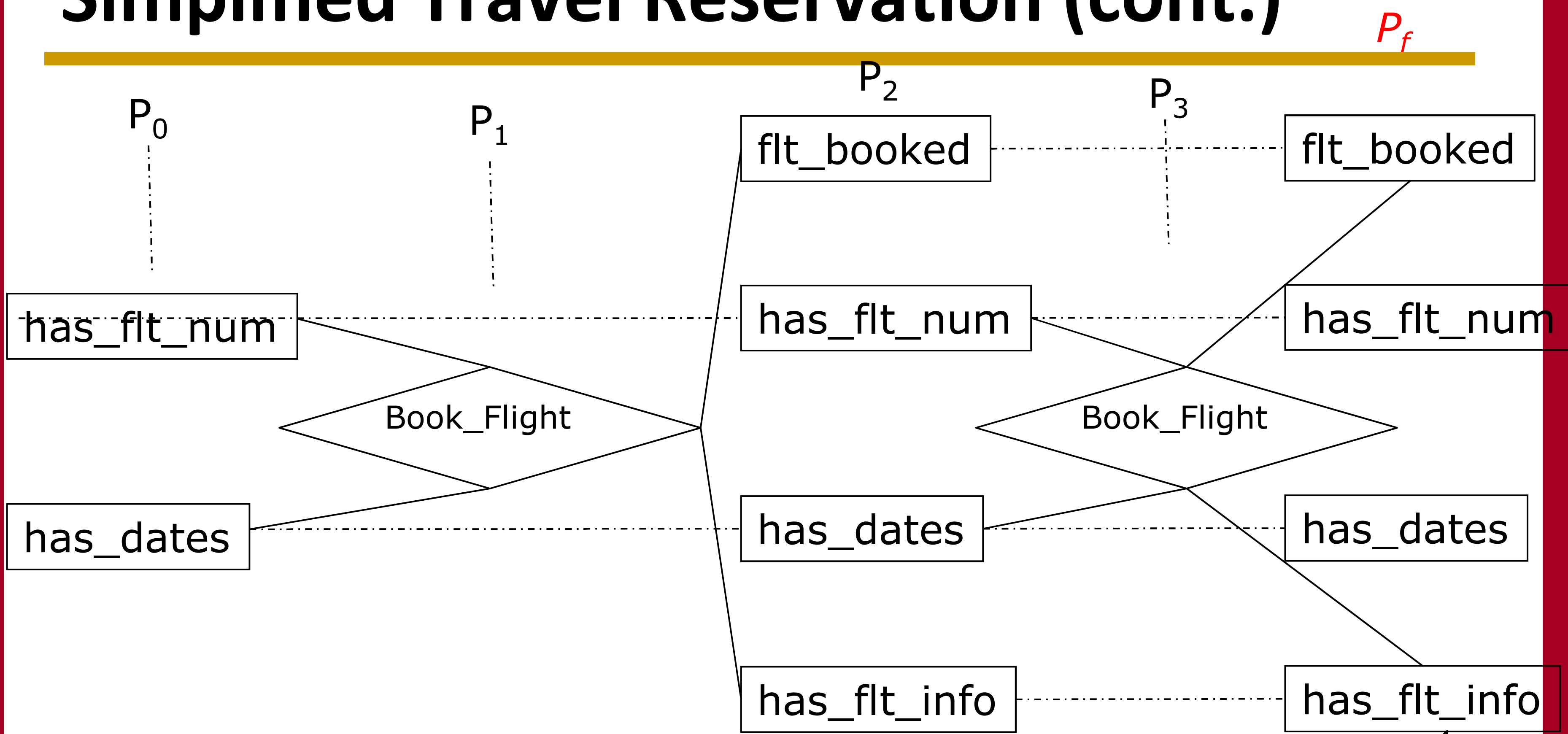
  the preconditions and effects of actions in $\sum^{-1}$ are the effects and preconditions of the corresponding actions in $\sum$.

Example: Simplified Travel Reservation

Action	Precondition	Effect
Book_Flight	has_flt_num, has_dates	flt_booked, has_flt_info
Book_Hotel	has_flt_info, has_dates	ht_booked, has_ht_info
Book_Shuttle	has_flt_info, has_ht_info, has_dates	st_booked

If the domain is incomplete, e.g.,
the action “Book_Hotel” is missing … …

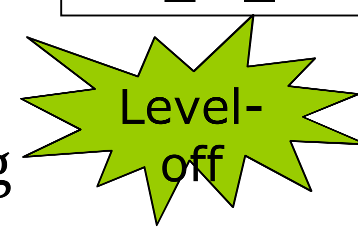
Simplified Travel Reservation (cont.)



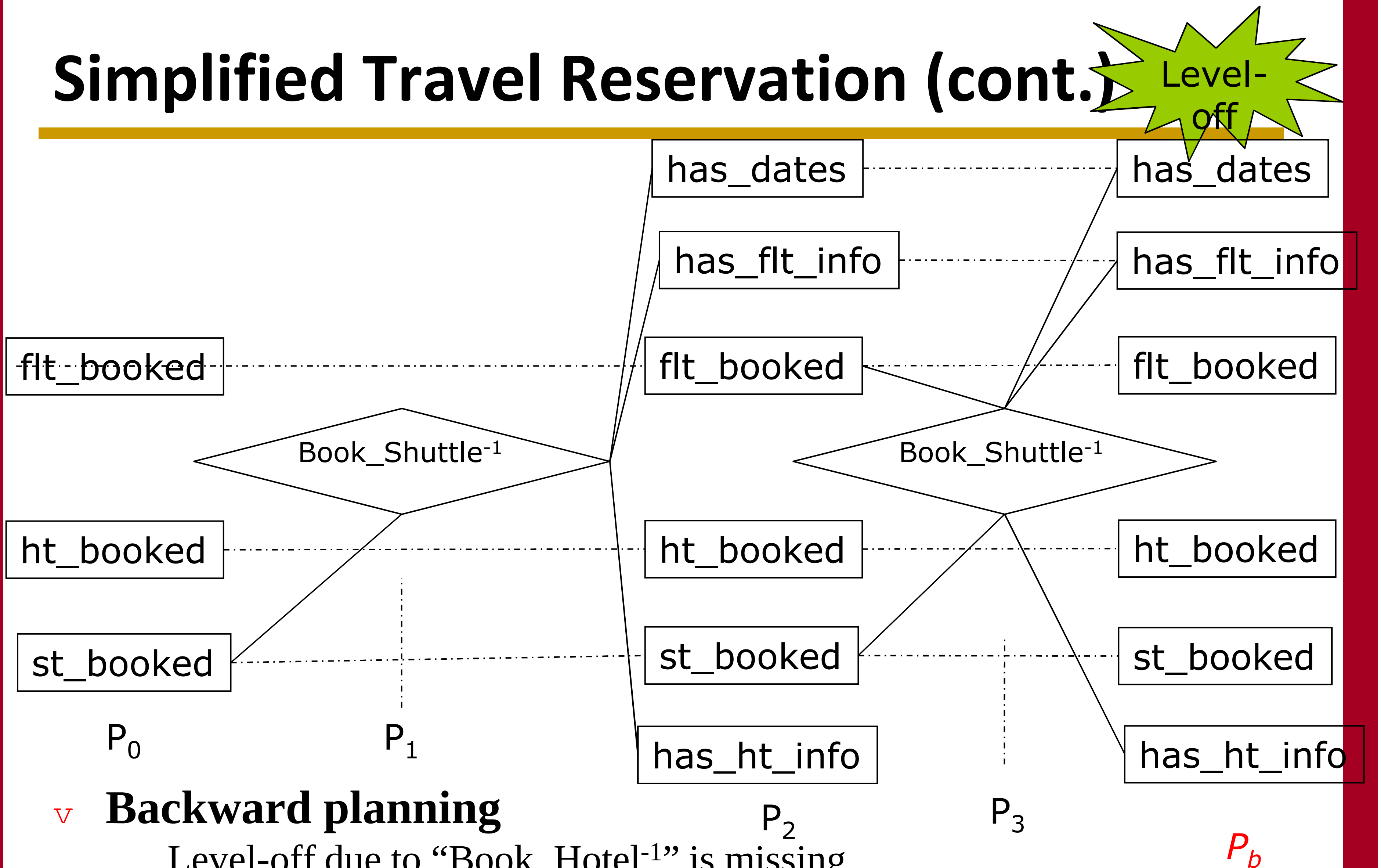
- **Forward planning**

  Level-off due to "Book_Hotel" action is missing

  P_f = {has_flt_num, has_dates, flt_booked, has_flt_info}



Simplified Travel Reservation (cont.)



- **Backward planning**
  - Level-off due to "Book_Hotel^{-1}" is missing
  - p_b = {flt_booked, ht_booked, st_booked, has_ht_info, has_dates, has_flt_info}

Question 2: Propose the Virtual Action

- p_f is the set of propositions in the last proposition level in the ***forward planning graph*** when level-off occurs
  - Contains the precondition of the virtual action
- p_b is the set of propositions in the last proposition level in the ***backward planning graph*** when level-off occurs
  - Contains the effect of the virtual action
- We focus on propositions that are only available in the forward or the backward planning but not both
  - Precondition of the virtual action is $P_{pre} = p_f - p_b$
  - Effect of the virtual action is $P_{eff} = p_b - p_f$

Question 2: Propose the Virtual Action (cont.)

- For the simplified travel reservation example
  - Precondition of the virtual action is $P_{pre} = p_f - p_b$
    - {has_flt_num, has_dates, flt_booked, has_flt_info} – {flt_booked, ht_booked, st_booked , has_ht_info, has_flt_info, has_dates}

      = {has_flt_num}
  - Effect of the virtual action is $P_{eff} = p_b - p_f$
    - {ht_booked, st_booked, has_ht_info}

The virtual action has recovered most of the information of the missing action "Book_Hotel"!!

Question 2: Propose the Virtual Action (cont.)

- However, the example is largely simplified
- The real world problems are much more complex

  We evaluated the above approach with benchmark problems from International Planning Competitions (IPCs)

  - The size of ($P_{re} = p_f - p_b$) is about 10, which is reasonable for humans to comprehend
  - The size of ($P_{eff} = p_b - p_f$) can be large (usually > 70)
    - There are > 2^{70} possible subsets !!!
    - Impractical to exhaustively enumerate all subsets to select the best one as the effect

- **Solution**: Using the genetic algorithm to determine the effect of the virtual action

Outline of Genetic Algorithm

1. Create an initial population of randomly generated chromosomes
2. Perform selection on the population based on the fitness values evaluated by a fitness function
3. Perform crossover and mutation on the selected chromosomes to produce the child population
4. If the max number of generations is exceeded, return the fittest chromosome
5. If any chromosome has a fitness value greater than or equal to the fitness threshold
6.   return the chromosome
7. Otherwise, return to step 2

Using GA to Determine the Effect of the Virtual Action

- Encoding a chromosome
  - Using binary string
  - Each bit in the chromosome corresponds to a proposition in P_{eff}
    - the propositions in P_{eff} are indexed
  - Example
    - If P_{eff} is indexed as {ht_booked, has_ht_info, st_booked}
    - The chromosome "110" denotes the subset {ht_booked, has_ht_info}
      - Because the bit corresponding to "st_booked" is 0 and therefore is excluded from the subset.

Using GA to Determine the Effect of the Virtual Action (cont.)

- Fitness function
  - For each chromosome c, we create a virtual action $a_v(c)$
  - The virtual action $a_v(c)$ may enable the traditional planner to generate a plan
  - If a plan is found
    - The fitness value of c is computed as the size of this plan
  - Otherwise, the fitness is -1
- Why is a chromosome fitter if a longer plan is generated?
  - A longer plan implies that more real actions are used, and that the role played by the virtual action is smaller
  - Although counter examples can be found, i.e., a shorter plan is fitter, this greedy strategy works well in practice

Using GA to Determine the Effect of the Virtual Action (cont.)

- Issue with the fitness function
  - Impractical to run a traditional planner to obtain a plan for each chromosome, especially when the population size or the number of GA iterations is large.
- **Solution:** use relaxed plans to improve efficiency
  - Relaxed actions and relaxed plans are widely used in heuristic search
  - Relaxed actions ignore their delete effects
  - Therefore, no two actions are mutually exclusive with each other
  - As a result, a relaxed plan can be quickly obtained

Evaluation

- All the problem domains are from International Planning Competitions (IPCs)

Domain	Description
Barman [BM]	A robot barman is responsible for manipulating drink dispensers, glasses, and a shaker. The goal is to find a plan of the robot's actions that serves a desired set of drinks.
PSR	Depending on the states of the switches and electricity supply devices, the flow of electricity through the network is given by a transitive closure over the network connections at any point in time
Openstacks [OS]	A manufacturer may have many orders. Each order consists of different products, which can only be made one at a time. The goal is to have all the orders shipped with a minimum number of stacks
ebookstore [EB]	The user provides a book title and author, credit card information and the address, as well as information about the shipping dates and the customs cost for the specific item.

Number of Actions Tested for Each Domain

- One or two action(s) are removed at a time from the benchmark domains
  - The removed actions were involved in the plans to the planning problems.
  - In other words, the removal of these actions will result in planning failures.
  - Removing more than 2 actions can be mimicked by removing 2 actions.

	BM	EB	OS	PSR
1 action	11	6	5	5
2 actions	7	4	3	6

Reco

- Son … more imp
  - C

For 57% of the cases, the virtual actions recovered all of the key propositions (i.e., complete); and for 21% of the actions, the virtual actions recovered some of the key propositions (i.e. partial)

Domain	**Completely**		**Partially**		**Missed**		**Total**	
	1 act	2act	1act	2act	1act	2act	1act	2act
BM	7	2	2	3	2	2	11	7
EB	6	4	0	0	0	0	6	4
OS	3	1	0	2	2	0	5	3
PSR	3	1	0	3	2	2	5	6

Precision and Recall

- Non-key propositions are also important
- For virtual action a_v, we evaluate how well $pre(a_v)$ and $eff(a_v)$ match the precondition and effect of the removed action (or action pair), $pre(a_r)$ and $eff(a_r)$.
- Applied two evaluation metrics, *precision* and *recall*
  - Precision: the percentage of the propositions in $eff(a_v)$ that appears in $eff(a_r)$
  - Recall: the percentage of propositions in $eff(a_r)$ that appears in $eff(a_v)$

Precision and Recall (cont.)

- Example
  - The action of “picking up container” is missing in the BM domain
    - Whose effect includes three propositions, namely, (*holding hand container*), (*not* (*ontable container*)), and (*not* (*handempty hand*))
  - The effect of the virtual action is
    - (*holding hand container*) and (*clean shot*)
      - Recovered the key proposition (*holding hand container*), but missed the other two propositions
      - Has an irrelevant proposition, i.e., (*clean shot*)
  - The precision is 1/2
    - One over two propositions in the effect of virtual action is correct
  - The recall is 1/3
    - There are three propositions in the effect of the removed real action

Preci

On average, the actions recalled ≥ 30% of the preconditions and ≥ 45% of the effects, and have a precision of ≥ 32% (preconditions) and ≥ 27% (effects).

	Precision				Recall			
	Precond		effect		precond		effect	
	1act	2act	1act	2act	1act	2act	1act	2act
BM	0.28	0.50	0.12	0.22	0.14	0.29	0.45	0.35
EB	0.67	0.50	0.58	0.67	0.67	0.67	1.00	1.00
OS	0.31	0.42	0.23	0.15	0.46	0.41	0.45	0.21
PSR	0	0.25	0.22	0.40	0	0.25	0.35	0.31
Avg	0.32	0.41	0.27	0.34	0.30	0.37	0.56	0.45

Human Evaluations

- We educated 17 human participants consisting of 12 undergraduate and 5 graduate students
  - All the participants had few or no knowledge about AI planning prior to enrolling in this study
- For each domain, an instruction document was prepared for them to read, including the domain description, how to read the actions on the domain, and the list of actions defined on the domain
- Handed out 52 virtual actions to different human participants, and for each virtual action a_v, we asked them to identify all the real actions in the domain that they thought similar to a_v.

Human Evaluations (cont.)

- If the evaluation is for one action, the response has a score of 1/*n* with *n* being the rank of the correct action
  - E.g., the participant identified and ranked a_1 and a_2 as relevant actions
  - If a_2 is the real action removed, the score is 1/2 because the rank of a_2 is 2 in the participant’s answer

Human Evaluations (cont.)

- If the evaluation is for two actions,
  - The response has a score of 1 (i.e., completely correct) if the first two actions are the right actions
  - Otherwise, the score is $1/m + 1/n$, where m and n are ranks of the right actions.
  - If the response only consists of 1 right action, the score is $1/2n$, where n is the rank of the right action.

Hu

On average, 73% of the answers included at least one of the right actions.

Domain	One Action		Two Actions	
	# of judgments	avg. score	# of judgments	avg. score
BM	6	0.25	6	0.23
EB	8	1	7	0.51
OS	8	0.73	7	0.43
PSR	6	0.31	4	0.19
Avg	---	0.61	---	0.36

Conclusion

- We proposed to use virtual actions to recover the missing information in the event of planning failure
- We used three different ways to evaluate the proposed approach
  - Key propositions recovery (could be subjective since different people may identify different key propositions)
  - Precision and recall (objective, but less intuitive)
  - Human evaluation (intuitive)
  - The results were consistent
- On average, 73% of the answers from the 17 research participants are partially or completely correct
  - Our proposed approach is promising