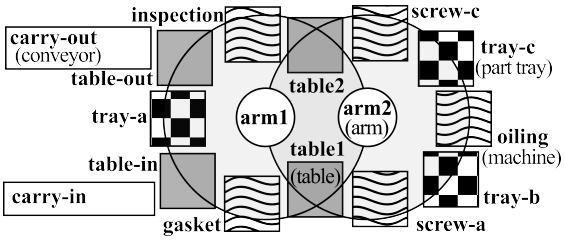
Applying Problem Decomposition to Extremely Large Planning Domains

M. Asai, A. Fukunaga
Univ. Tokyo

Mass-Manufacturing Domain and Loop Unrolling Strategy



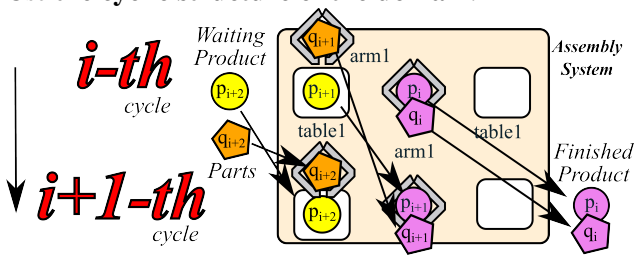
CELL-ASSEMBLY ≈ Gripper + Woodworking + Logistics

- Assemble and paint products while moving them with arms between tables or special purpose machines
- also a *temporal* problem (actions run in parallel)

Large-scale problems in factory assembly require the manufacturing of **100's or 1000's of identical products.** This is clearly beyond the reach of current planners.

$\Longleftrightarrow$ Solved for **homogenous repetitive problems** by ACP [Asai2014].

Use **the cyclic structure of the domain**.



Everythings but indices of the products are the same after a cycle. Each Loop path is identified by Steady State (SS) ≡ a state in the beginning of one cycle, indexed with i.

ex. S_i : (at p_{i+1} table1), (at q_{i+1} arm1), (at p_i arm1), (attached p_i q_i), (painted p_i q_i).

Build a large plan by unrolling the loop.

Need to overcome two obstacles for solving Heterogeneous repetitive problems

Objects are not always completely equivalent : Problem 1

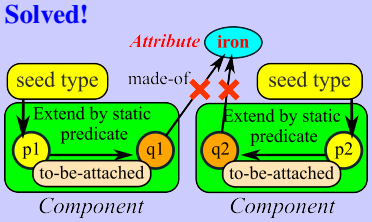
- p_1 : Init (not-painted p_1) → Goal (painted p_1)
- p_2 : Init (painted p_2) → Goal (not-painted p_2)

Sometimes, objects form a STRUCTURE : Problem 2

If (to-be-attatched p_i q_i), we cannot treat them independently
e.g. **fix** $i = 1$ on p_i (p_1) but **iterate** j on q_j (q_1, q_2, q_3 . . .)
→ invalid plan ∵ p_1 is not to-be-attatched with $q_{j\neq 1}$

Classifying Objects w/ Components and Plan-Compatibility Analysis

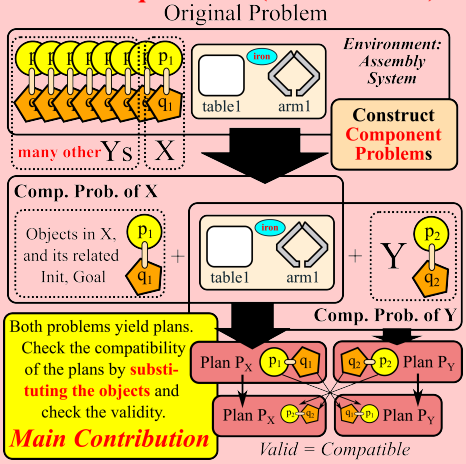
1. Component Abstraction [Botea et.al. 2004]. Prob.2 Solved!



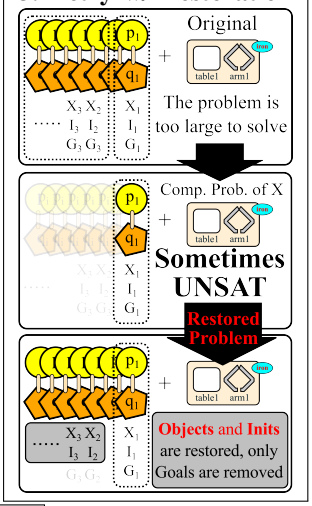
1. Static Graph (Nodes : objects, Edges: $f \in$ Init s.t. never added/deleted)
2. Pick a *seed type* (a PDDL type)
3. Extend until they share any node

New: Attributes prevent extention

2. Component Plan Compatibility: to ensure their equivalence (Prob.1 solved!)



3. Retry w/ Restoration



4. Future work: completing a framework **integrated with ACP**.

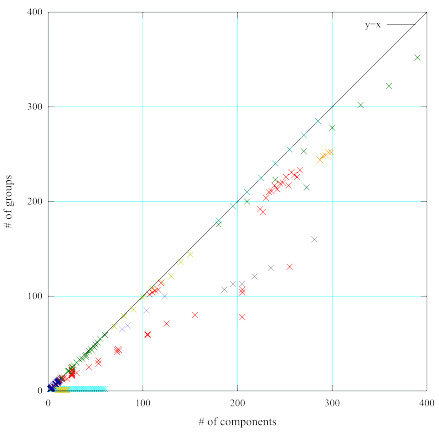
Improving restoration method (even without goals, the restored problem may require excessive memory during preprocessing)

More systematic component detection focused on **larger components**

Interleaving multiple **homogeneous** loops (constructed with classification technique + ACP)

Experiments

- CELL-ASSEMBLY-eachparts:
  - variant of CELL-ASSEMBLY with explicit part-product assignments
  - **All equivalence groups correctly detected**
- **Very large instances of IPC problems**
  - **Large problems** that FD/LAMA failed @ 15GB, 6hrs, Xeon E5410 2.3GHz
  - typed satellite (310 directions), woodworking (p86, 227 parts), openstacks (70 orders / products), elevators (270 passengers), barman-sat11 (93 shots), rover(p40)
  - Counted **how much they are categorized into equivalence group**



Conclusions : Proposed a decomposition-based approach for **heterogeneous** repetitive problems. **Preliminary Results:**

(1) On very large problem instances, our method is able to decompose the problems into compatible components.

(2) **Structural compatibilities were found in large instances of IPC domains.**