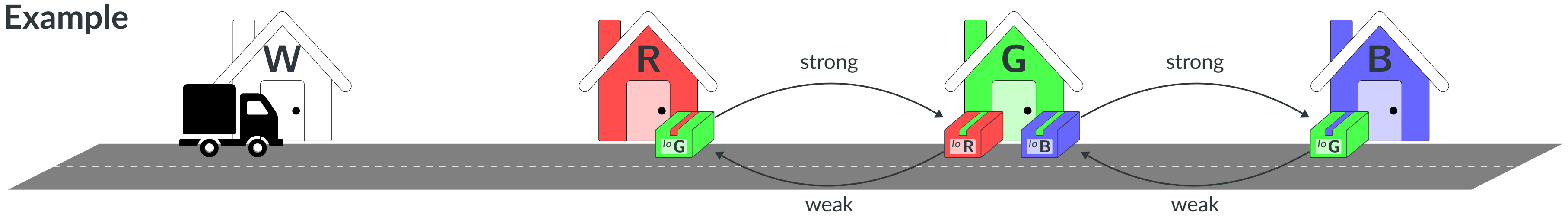
Considering *cyclic dependencies* between landmarks *improves heuristics*.

Example



Landmark: "Visit G to pick up B."
Strong landmark ordering: "Visit R before G because there is no path around it."
Weak landmark ordering: "Visit R after G to deliver R."

Landmark Heuristic

- Every plan must satisfy all landmarks at least once.
- Use ***operator-counting framework*** to estimate cost:

$$\min \sum_{a \in \mathcal{A}} \mathrm{Y}_a \cdot cost(a) \quad \text{s.t.} \qquad (1)$$

$$\mathrm{Y}_a \geq 0 \quad \text{for all actions } a \in \mathcal{A} \qquad (2)$$

$$\mathrm{Y}_L := \sum_{a \in L} \mathrm{Y}_a \geq 1 \quad \text{for all landmarks } L \in \mathcal{L} \qquad (3)$$

- Example: $h^{\mathrm{LM}} = 3$ because R, G, and B must all be visited at least once.

Cyclic Landmark Heuristic

- R must be visited both before and after G.
- ***Cyclic dependency***: one landmark per cycle required twice:

$$\sum_{L \in \mathcal{L}(c)} \mathrm{Y}_L \geq |\mathcal{L}(c)| + 1 \quad \text{for all cycles } c \in \mathcal{C} \qquad (4a)$$

- Example: $h^{\mathrm{cycle}} = 4$ because visiting G twice resolves both cycle constraints.

Strong Cyclic Landmark Heuristics

- R cannot be delivered when first visiting R.
- Only landmarks with ***incoming weak ordering*** can resolve cycles:

$$\sum_{L \in \mathcal{L}^{\mathrm{w}}(c)} \mathrm{Y}_L \geq |\mathcal{L}^{\mathrm{w}}(c)| + 1 \quad \text{for all cycles } c \in \mathcal{C} \qquad (4b)$$

- Example: $h^{\mathrm{strong}} = 5$ because R and G must be visited twice to resolve both cycles.

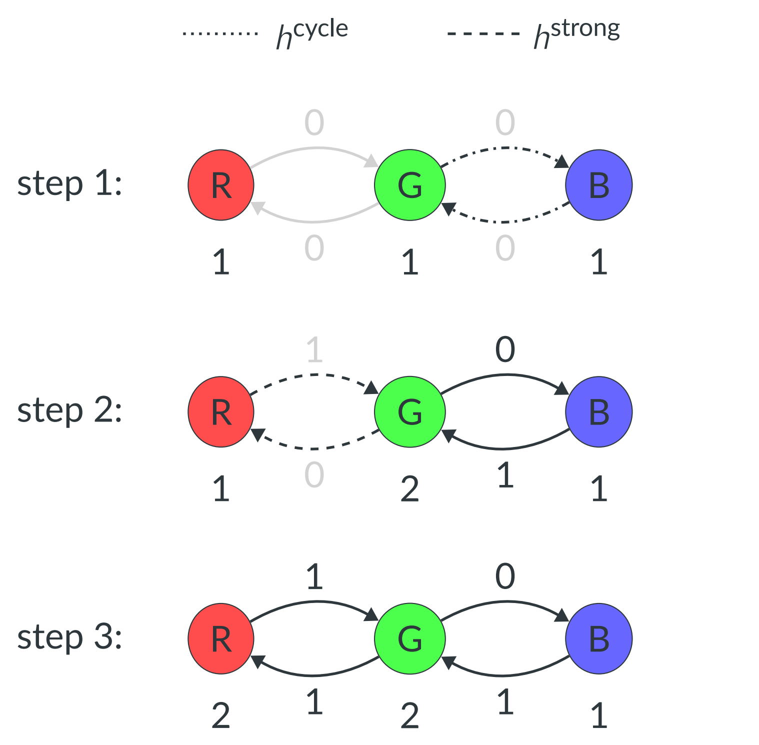
Finding Cycles in LM Graphs

Johnson's Algorithm

- Finds ***all*** elementary cycles.
- Infeasible in graphs with many cycles.

Oracle Approach

- Few cycles are often sufficient to cover all cycles.
- Use ***implicit hitting set*** algorithm to find a sufficient subset of all cycles iteratively:
  1. Solve LP (initialized using Eq. (1–3)).
  2. Construct weighted graph with $w_{L \to L'} = \mathrm{Y}_{L'} - 1$.
  3. Compute shortest cycles using Floyd-Warshall.
  4. Add constraint (4) of ***most uncovered*** cycle c with minimal $\sum_{L \to L' \in c} w_{L \to L'} < 1$.
  5. Repeat until all cycles are covered.
- Disadvantage: needs multiple LP runs.



Exploiting Cyclic Dependencies in Landmark Heuristics

Clemens Büchner, Thomas Keller, and Malte Helmert

full paper

University of Basel