

ADAPTIVE TASK ALLOCATION TO HETEROGENEOUS AGENTS

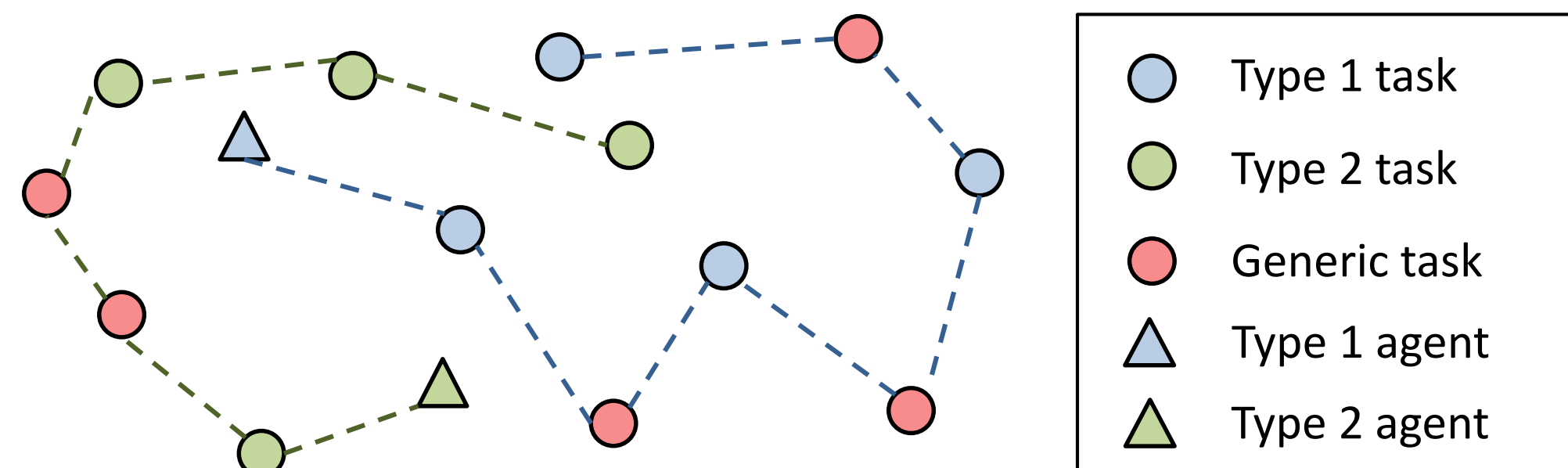
Amritha Prasad*, Han-Lim Choi†, Shreyas Sundaram*

*School of Electrical and Computer Engineering, Purdue University, †Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology (KAIST)
E-mail: prasad20@purdue.edu, hanlimc@kaist.ac.kr, sundara2@purdue.edu

General Framework

Given:

- A set of k heterogeneous agents $\mathcal{A} = \{A_1, A_2, \dots, A_k\}$, where each agent belongs to one of m types.
- Set $D = \{v_1, v_2, \dots, v_k\}$, where v_i denotes the start location of the i^{th} agent.
- A set of tasks T comprising of a set of generic tasks T_0 and sets of type-specific tasks T_i of each type i , $1 \leq i \leq m$.
- A complete graph G on the set of nodes $T \cup D$.



Objective:

- Find an allocation of tasks to agents such that
- all tasks are covered by at least one agent.
 - task-agent compatibility constraints are met.
 - maximum cost for any agent to complete its allocated tasks is minimized.

Problem Formulation

Heterogeneous Agent Path Problem (HAPP):

$$\min_{S_1, S_2, \dots, S_k} \max_{1 \leq j \leq k} P_j^*(S_j)$$

subject to $\bigcup_{j=1}^k S_j = T, S_i \cap S_j = \emptyset, \forall j \neq i,$
 $S_j = V_j \cup R_j, \forall j \in \{1, 2, \dots, k\},$
 $V_j \subseteq T_{f(j)}, R_j \subseteq T_0.$

$P_j^*(\cdot)$: Cost of optimal path (starting from node v_j) on a given set of tasks

$f(j)$: Returns type of agent A_j

S_j : Tasks allocated to agent A_j

V_j : Type-specific tasks allocated to agent A_j

R_j : Generic tasks allocated to agent A_j

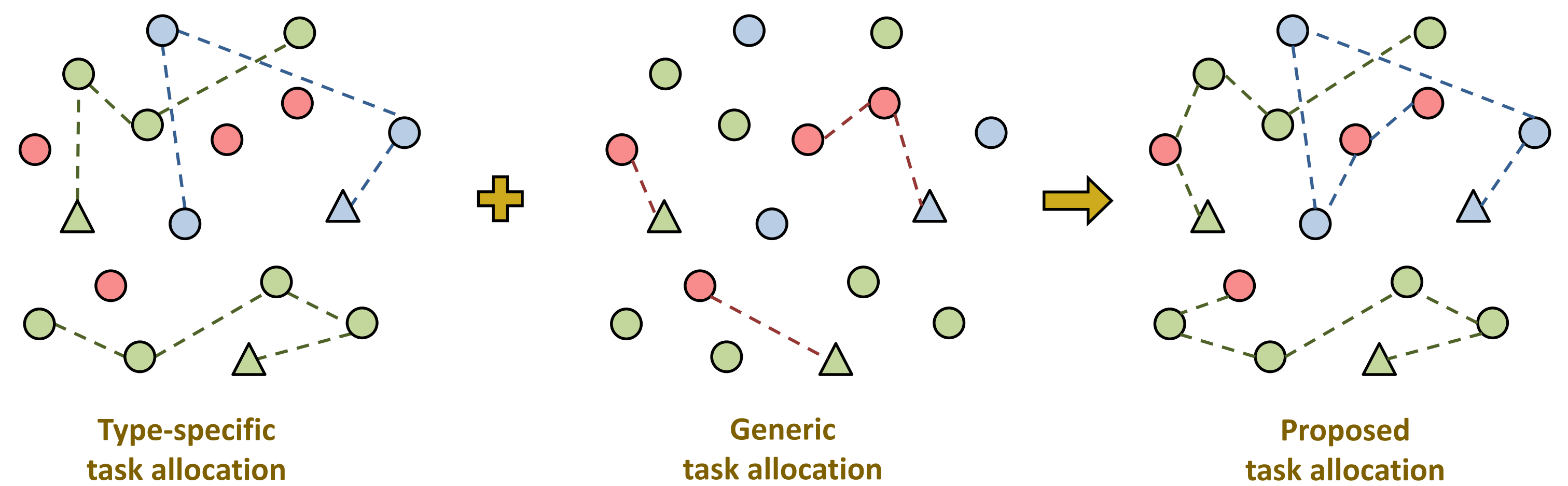
Salient Features

- Robust adaptive allocation of tasks to agents that **can handle failure of agents and cater to new tasks**.
- Most existing work on heterogeneous agents focuses on minimizing the sum of costs incurred by all agents. We minimize the maximum cost - **important in time critical applications** where maximum latency needs to be low.
- We consider **functional heterogeneity**. Most works encapsulate heterogeneity in terms of vehicle dynamics.
- Several works on paths for homogeneous agents to minimize min-max cost. We extend these works to task allocation to **heterogeneous** agents while **minimizing the maximum cost** of the path.

Key Idea:

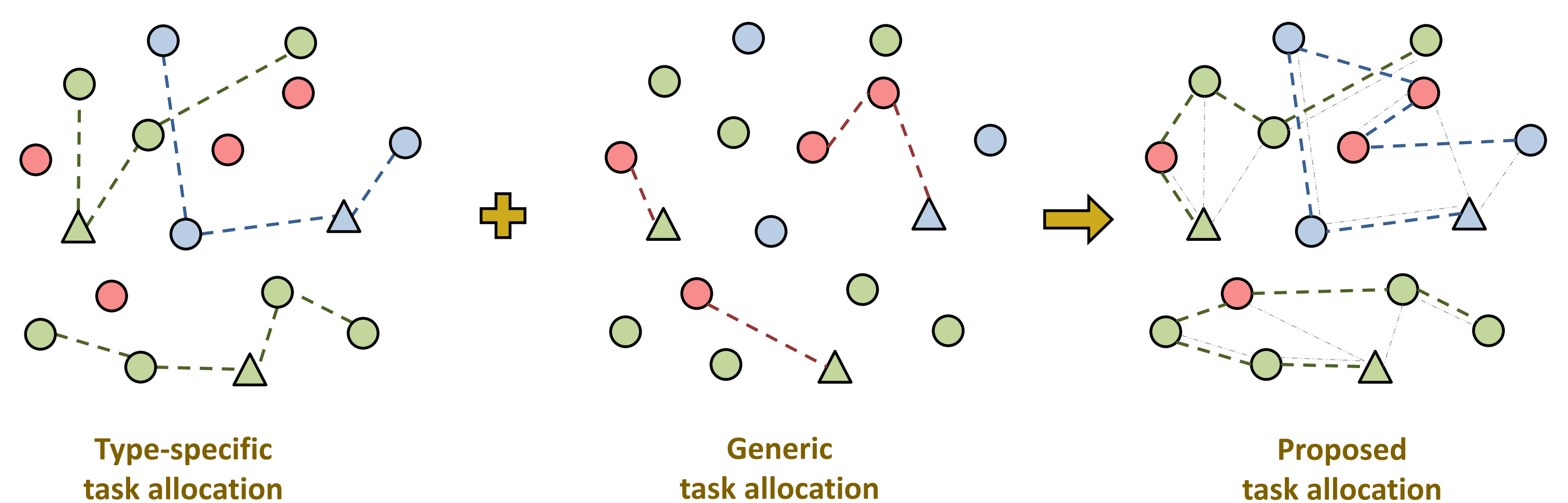
- Consider type-specific tasks and generic tasks separately. We can now use algorithms for homogeneous agent task.
- Two approaches: Path Split and Tree Split
- Existing algorithms for homogenous agent tasks allocation.
 - Min-max Path Cover Problem (Yu et. al., 2017).
 - Min-max Tree Cover Problem (Even et. al., 2004).

Approach 1: HeteroMinMaxPathSplit



- Solve homogeneous agent min-max path cover problem for each set of type-specific tasks T_i (for m_i agents, where m_i is the number of agents of type i).
- Solve the homogeneous agent min-max path cover problem on the set of generic tasks (for k agents).
- Combine the above two solutions to obtain a solution for HAPP.
- If an α -approximation algorithm is used to solve the homogeneous agent problem, we show that our algorithm to solve HAPP has an approximation factor of 3α .
 - Yu et. al. provide a 7-approximation solution to the homogeneous agent min-max path cover problem, which can be used to obtain a 21-approximation algorithm for HAPP.

Approach 2: HeteroMinMaxTreeToPath



- Solve homogeneous agent min-max tree cover problem for each set of type-specific tasks T_i (for m_i agents, where m_i is the number of agents of type i).
- Solve the homogeneous agent min-max tree cover problem on the set of generic tasks (for k agents).
- Combine the above two solutions to obtain a solution for HAPP.
- If a β -approximation algorithm is used to solve the homogeneous agent tree cover problem, we show that our algorithm to solve HAPP has an approximation factor of 4β .
 - A factor of two comes from combining the type-specific subtree and generic task subtree for each agent. An additional factor of two is a result of converting trees to paths.
 - Even et. al. provide a 4-approximation solution to the homogeneous agent min-max tree cover problem, which can be used to obtain a 16-approximation algorithm for HAPP.

Conclusion and References

- Robust adaptive allocation of tasks to agents that is resilient to agent failures.
- Polynomial time approximation algorithms with bounds on worst case performance that is irrespective of number of agents or tasks.
- Future work: Distributed algorithm to solve HAPP.
- Several works on paths for homogeneous agents to minimize min-max cost. We extend these works to task allocation to **heterogeneous** agents while **minimizing the maximum cost** of the path.