Coordination-aware task allocation via a lightweight supervisory layer

An event-driven approach to multi-agent Search and Rescue

Author
T.R. Schutgens
Supervisors
A. Jamshidnejad
S. Schoonebeek

TU Delft

1. The problem

Search and Rescue (SaR) can benefit from autonomous agents.

In SaR missions, we need to:
- Find victims quickly
- Explore as much as possible
- Remain responsive to rescue

With multiple agents, we need to avoid conflicts so we need Supervision:

How can a lightweight supervisory framework dynamically allocate sub-tasks within a robotic team to optimize the trade-off between safety constraints and performance objectives under environmental uncertainty?

2. The Testbed

Tiny 2D grid-based testbed
Properties include:
- Agents and victims
- Propagating fire
- Vulnerability and perception

Map generated procedurally:
- Rooms and tunnels with sizes
- Initial fire
- Vulnerability

Example 50x50 map

3. Supervisory Architecture

The supervisory task-allocation layer will
- Sit above the controllers
- Aim to allocate tasks as efficiently as possible.

The supervisor will:
- Let agents act freely
- Intervene when **event** occurs

Event types:
- Spatial conflict: agents come too close
- Impatience: agent stops discovering new territory
- Victim discovered: high-priority rescue trigger

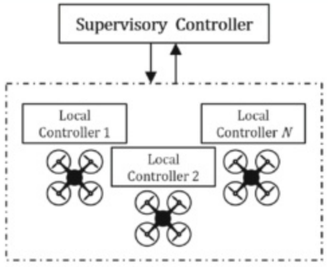


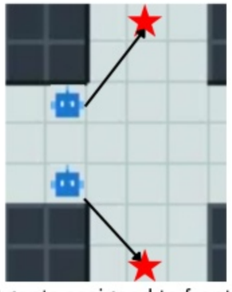
Diagram of Architecture

4. Event Resolution

Task generation

The supervisor will, when event arises:
- Create new task(s)
- Compute bidding costs for agents
- Assign task to lowest cost agent

Frontiers:
- Border between explored and unexplored area



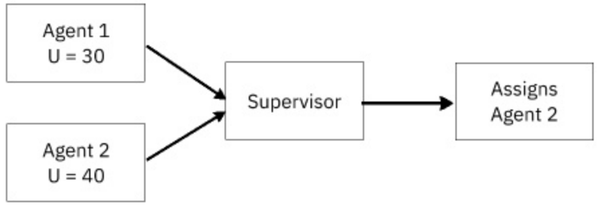
Agents assigned to frontiers

Auction System

- SSI auction, single-task capacity per agent
- Capability multiplier gates rescue-only tasks
- Burgard spatial penalty drives dispersion
- All agents bid when victim discovered

$$U_{i,j} = (B_j \cdot C_{i,j}) - T_{i,j} - P_{i,j}$$

Function used in the auction:
- U = Utility
- B = Base reward (task type)
- C = Capability multiplier
- T = Travel cost
- P = Penalty (Bulgard spatial exclusivity)



5. Results

Supervised compared to 2 baselines across same seeds:
- Unsupervised (fully decentralized)
- Shared (only has shared map)
- Success when all victims are found before timeout

Homogenous fleet:
- Much higher success rate
- Higher exploration velocity
- Less exploration redundancy

Heterogenous fleet:
- Exploration increase persists
- Does not yet translate to more saved victims

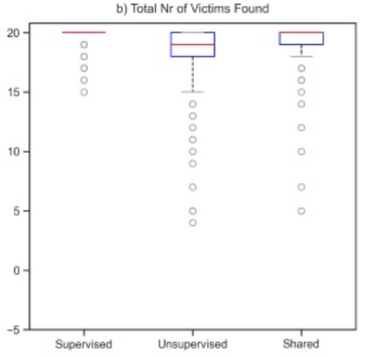
Shared baseline has much higher communication bandwidth
- Intervention duration averages around 11ms
- Supervision reduces inter-agent collisions
- Damage reduced by 25%

Success rates and times across 216 paired simulations:

Architecture	Success Rate	Time to All Found
Supervised	91.7%	177.59 ± 43.93
Unsupervised	50.5%	220.81 ± 53.37
Shared	75.9%	184.02 ± 50.03

Average time steps to reach milestone:

Milestone	50%	70%	80%	85%	90%
Supervised	81	108	126	136	151
Unsupervised	92	137	165	183	201
Shared	80	108	128	139	153



6. Conclusions

What we showed:
- Event-driven supervision outperforms decentralized coordination for exploration
- Active allocation also outperforms shared-map awareness
- Safety does not suffer under these gains
- Exploration and rescuing are not yet balanced properly

What remains open:
- More interesting heterogeneous fleet
- Lossy communication
- Comparison against other supervisory methods