Distributed Fleet Management in Noisy Environments via Model-Predictive Control

Simon Bøgh, Peter Gjøl Jensen, Martin Kristjansen, Kim Guldstrand Larsen, Ulrik Nyman

AALBORG UNIVERSITY DENMARK

AALBORG UNIVERSITY DENMARK

Motivating Case

Current situation:

- Custom orders consisting of many small components
- Currently picked by humans

To be replaced by:

- A factory floor with Autonomous Mobile Robots (AMRs)
- Multi-agent multi-point pick-up and delivery
- Humans and stochastic travel times

Contributions

- New formalism: Stochastic Work Graphs (SWG)
- Encoding as Euclidean Markov Decision Process (EMDP)
  - Can be solved by UPPAAL STRATEGO
- Benchmark platform
  - Open source (MIT license)
  - Can you do better than our approach for this problem description?

Distributed Planning Algorithm

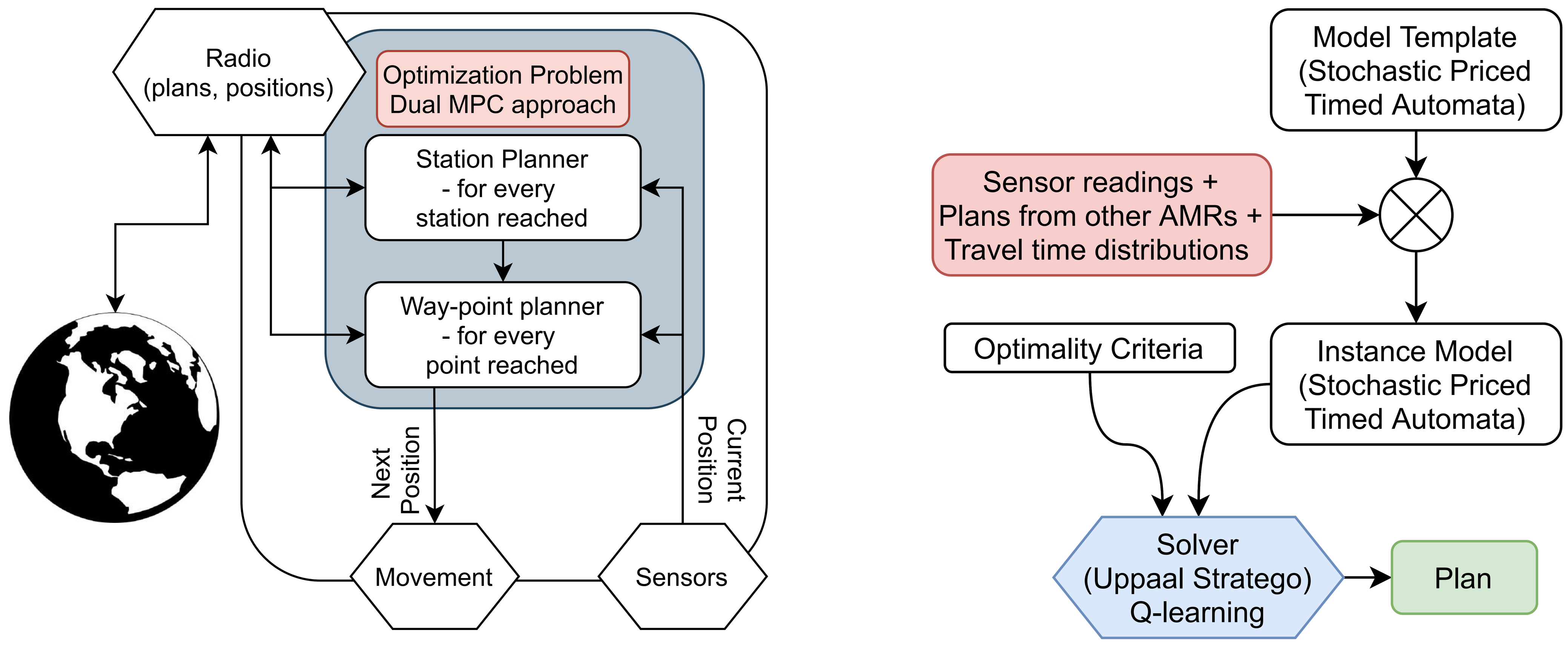
- Continuous distributed re-planning by each AMR.

Algorithm 1: Distributed Planning Algorithm

Data: The id of the AMR $r \in R$ and the SPSWG $S = (R, I, \emptyset, G)$ with $G = (V, E, \Delta)$

$P_r = \{\rho^{assign}, \dots\}$ is the initial strategies, excluding one for r;
while /*loop forever*/ **do**
  Receive a task τ from coordinator (according to ρ^{assign});
  while $\tau \neq \emptyset$ **do**
    Receive updates to P from other AMRs;
    Make an observation Λ projected into Λ_ϕ;
    Assume w.log. that $\rho'((\mathcal{T}, \Lambda_\phi))(\delta) = 1$ for all $\rho' \in P$;
    Compute σ_P^ϕ for $\phi(S)$ starting in state Λ_ϕ with $\Lambda(r)$ updated with task t;
    Let v be the vertex picked by σ_P^ϕ in Λ_ϕ;
    Send $\sigma_P^\phi \setminus P$ to other AMRs;
    while $\Lambda(r) \neq (v, b, c, \tau)$ *for any values of* b, c, τ **do**
      Receive updates to P from other AMRs;
      Make an observation Λ projected onto Λ_ω;
      Assume w.log. that $\rho'((\mathcal{T}, \Lambda_\phi))(\delta) = 1$ for all $\rho' \in P$;
      Compute σ_P^ω for $\omega(S, v)$ starting in state Λ_ω;
      Send $\sigma_P^\omega \setminus P$ to other AMRs;
      Let w be the vertex picked by σ_P^ω in Λ_ω;
      Send coordinates of w to low-level navigation controls;
    end
    Execute work at v;
    $\tau \leftarrow \tau \setminus \{v\}$;
  end
end

Overall Approach



Results

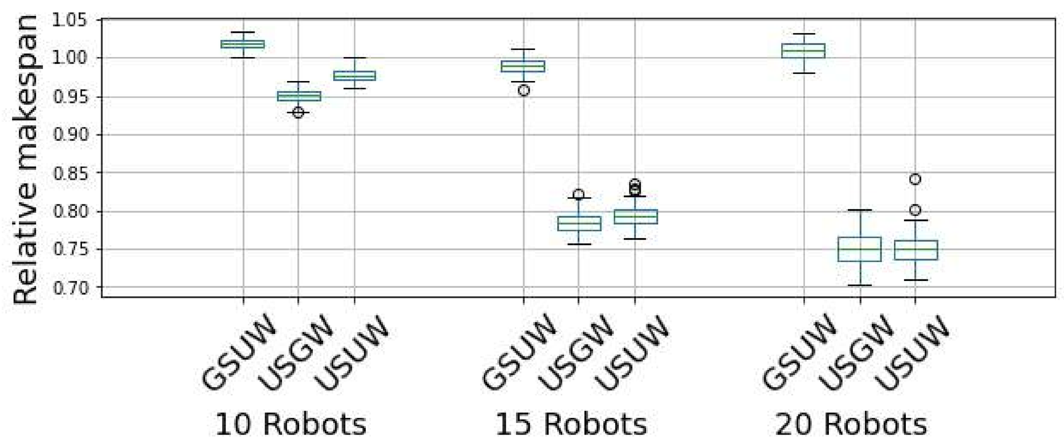


Fig. 1: Uniform, 1000 episode budget.

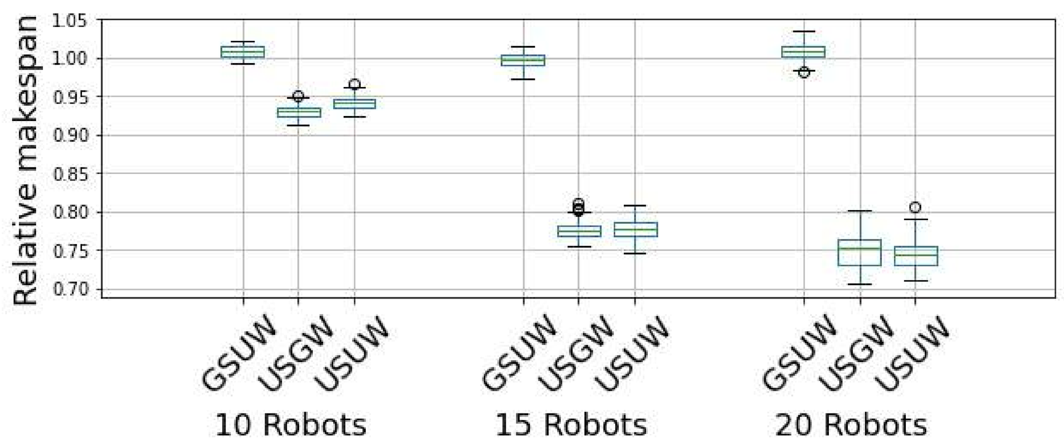


Fig. 2: Uniform, 5000 episode budget.

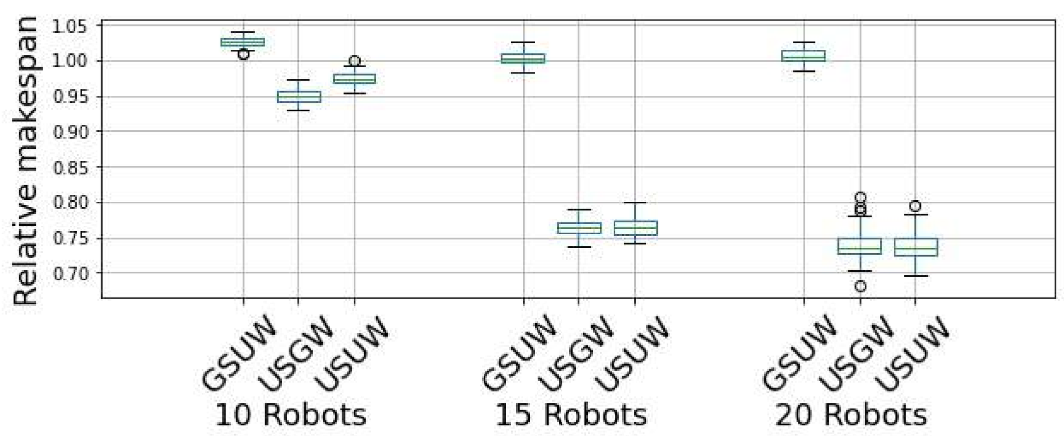


Fig. 3: Triangular, 1000 episode budget.

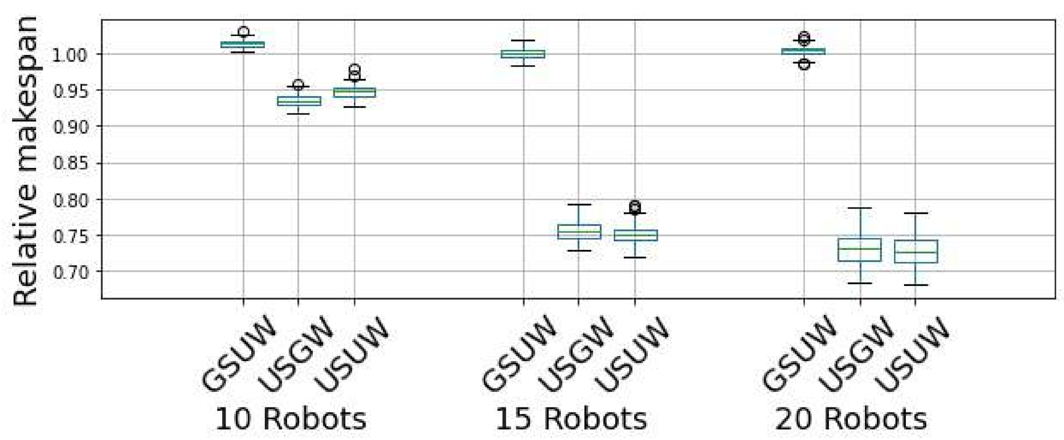


Fig. 4: Triangular, 5000 episode budget.

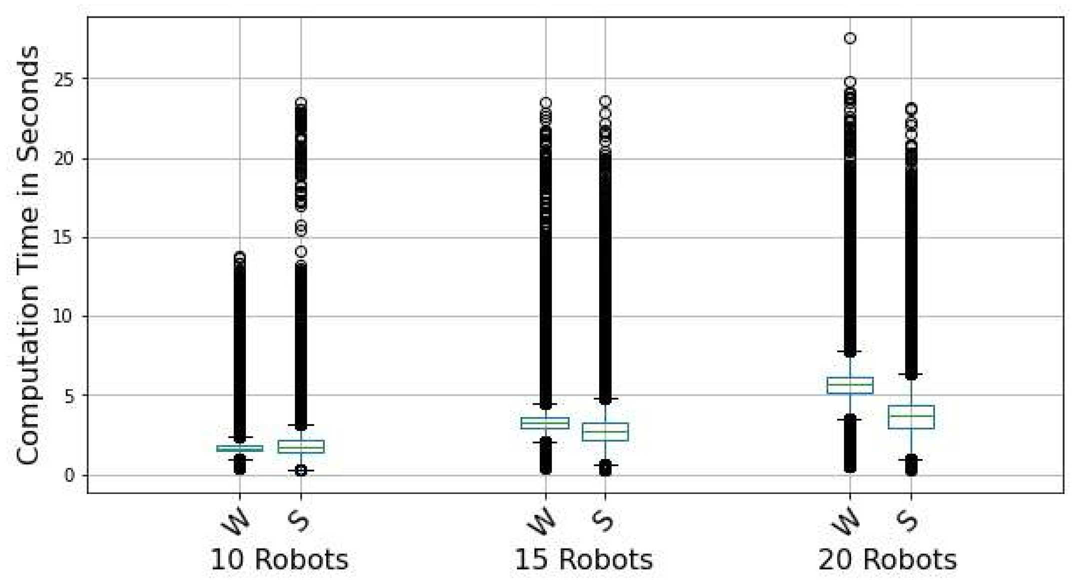
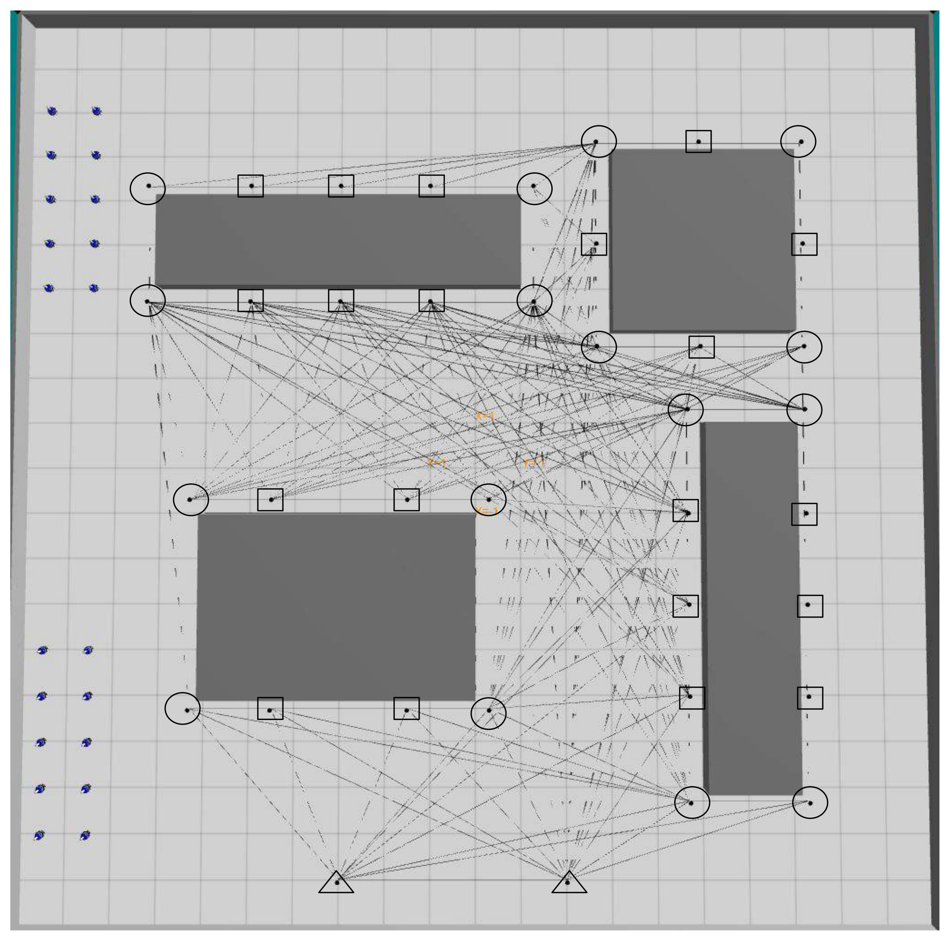
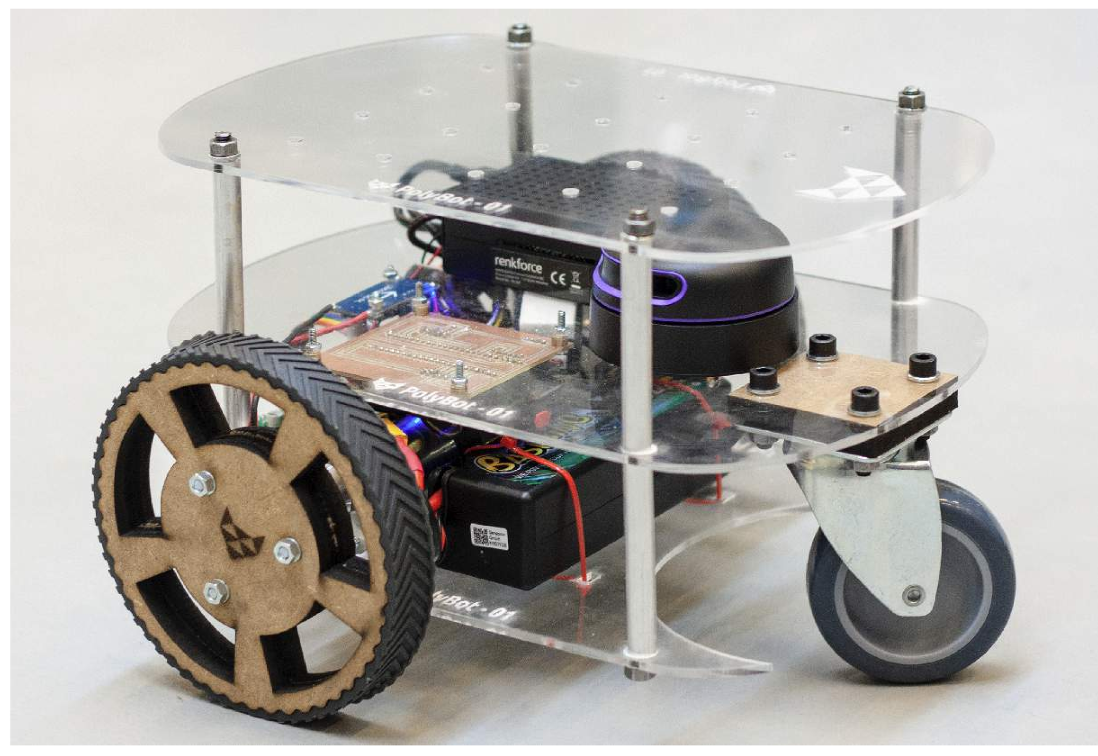


Fig. 5: CPU time spent for synthesizing a single way-point (W) or station (S) plan with a 1000 episode budget.

Experimental Map



Autonomous Mobile Robot



Conclusion

- 25% reduction in makespan compared to baseline
- 27.5% with more congestion
- No improvement for the near-term planning

Future Work

- Experiments with real robots
- Compare against other methods