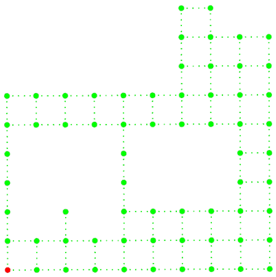
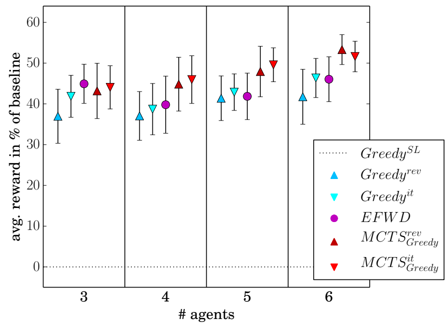
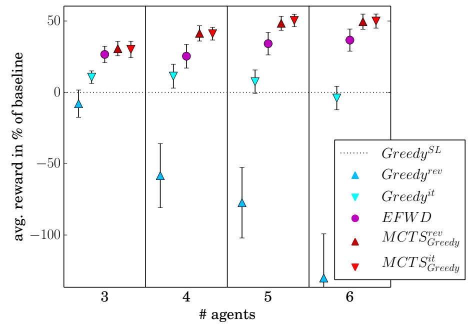


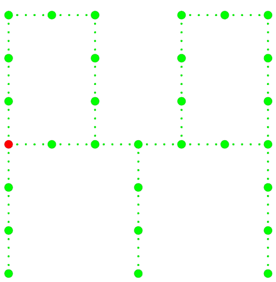
(a) *office-world*, #nodes=66



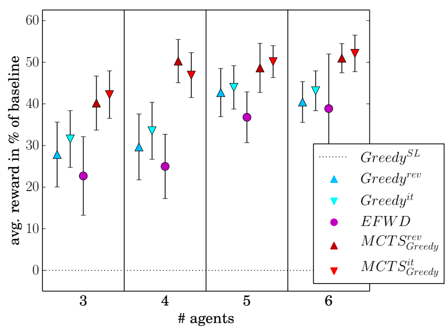
(b) *office-world*, unlimited capacity



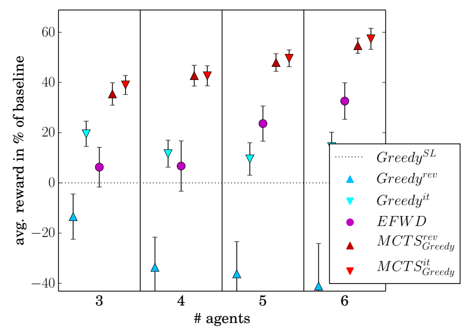
(c) *office-world*, limited capacity



(d) *warehouse-small*, #nodes=30



(e) *warehouse-small*, unlimited capacity



(f) *warehouse-small*, limited capacity

Figure 3: Comparing the different approaches in the worlds *office* (a-c) and *warehouse-small* (d-f) with infinite capacities and uniformly distributed task appearances and with limited capacities and distributed task appearances. The whiskers show the 95% confidence intervals.

take care of some task allocation, which helps to improve the action selection. Therefore, we focus on these two approaches for further evaluation.

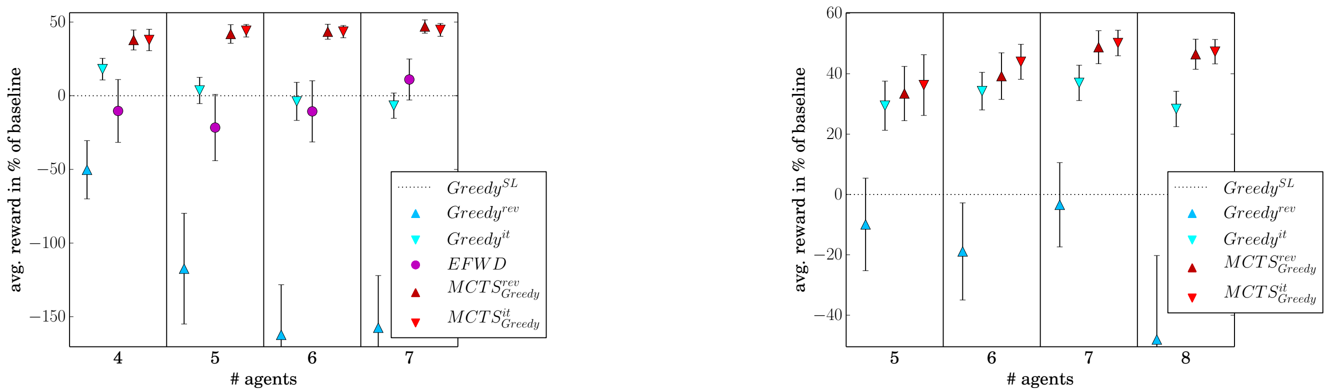
Planning times.

To investigate the planning times, we randomly sampled 200 different states and averaged the time it takes to plan in these states. We also included the state-of-the-art *EFWD* approach. The results are summarised in Figure 2d. We can see that the search times of all approaches increase roughly linearly with the number of agents. However, while the MCTS-based approaches have a very consistent planning time, the *EFWD* approach varies greatly. This is due to the k-nearest task approximation. When there is no task active, the planning time is nearly 0. However, as soon as there are one or more tasks active, the planning time increases greatly. We can also see that $MCTS^{rev}_{Greedy}$ is significantly faster than $MCTS^{it}_{Greedy}$, since it does not need to iteratively assign the tasks after the evaluation.

Limited vs unlimited capacities.

We have evaluated the approaches in settings where the agents have unlimited capacity and the tasks appear uniformly distributed over the warehouse. Essentially, when the agents have unlimited capacities, the depot node is obsolete, since the agents will never return to unload. Figure 3 shows the result for the *warehouse-small* environment and the *office* environment (the *office* is the same as in [9]). In comparison, with unlimited capacities (cf. Figure 3e and 3b), the resulting performances are closer to one another than with limited capacities (cf. Figure 3f and 3c). The MCTS based approaches still outperform all other approaches. The proposed heuristics without MCTS ($Greedy^{rev}$ and $Greedy^{it}$) perform as good as the *EFWD* approach. Most significantly, the $Greedy^{rev}$ policy performs almost as good as the $Greedy^{it}$ policy, which is in great contrast to the limited setting, where it yields even less reward than the $Greedy^{SL}$ baseline. This can be explained by the structure of the unlimited setting. The adaptive partitioning of $Greedy^{rev}$ works a lot better, since the agents spread due to the appearing tasks, and do not have to come back to the depot again. Thus assigning the tasks based on their locations works well. *EFWD* performs much better in the *office-world* in comparison to the *warehouse-small*. This is most likely due to the structure of the worlds. The *office-world* is much more interconnected, with almost no dead ends. Since *EFWD* does no positioning when there are no tasks present, it helps that the average shortest path length between nodes is a lot shorter in *office-world*.

To conclude, while the presented heuristics work really well in simple environments already by themselves, adding MCTS search still improves their performance. *EFWD* yields good performance in highly connected worlds.



(a) *warehouse-medium*, #nodes=66

(b) *warehouse-large*, #nodes=214

Figure 4: Comparing the different approaches in larger warehouse sizes and with different numbers of agents. The whiskers show the 95% confidence intervals.

Larger warehouses.

Additionally, we compared the different approaches in two larger sized warehouse models, *warehouse-medium* with $n = 66$ (cf. Figure 5) and *warehouse-large* with $n = 214$ (cf. Figure 1). For the large warehouse, we increased the number of simulated steps to 250 and the number of repetitions was decreased to 15.

EFWD was not able to complete any run in *warehouse-large* due excessive planning times, i.e. more than 1000 seconds for one step. The results are shown in Figure 4. *EFWD* shows a generally increasing performance for more agents, but the relative rewards against the baseline are varying. Only for larger numbers of agents it can outperform the baseline and standalone rollout strategies. We can see that the two MCTS approaches perform nearly identically for *warehouse-small* and *warehouse-medium*. The good performance of $Greedy^{it}$ in *warehouse-large* is due to the high chance that there are many tasks active in a larger world. Therefore, iteratively assigning the best tasks yields a very good result. As soon as the number of agents increases however, MCTS search improves the result again, since there are relatively fewer tasks to distribute, and positioning becomes more important again.

Positioning.

To show the effect of positioning, we let $MCTS^{rev}_{Greedy}$ run for 50 steps, while we disabled the appearance of new tasks assumed by the agents' world model. Figure 5 shows that the robots are nicely spread out. This is in stark contrast when using the heuristics without any MCTS search and also the state-of-the-art *EFWD* approach. These remain in the same position if no tasks is present.

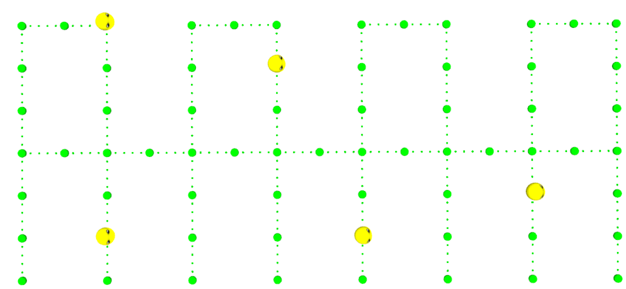


Figure 5: Positioning of $MCTS^{rev}_{Greedy}$ in the world *warehouse-medium* after 50 steps. All agents started in the top left corner.

6. CONCLUSIONS & FUTURE WORK

In this paper, we introduced a Monte Carlo Tree Search approach for the problem of multi-robot warehouse commissioning. The problem is modeled as an extended SPATAP to include the capacity constraints of robots. Our empirical evaluation shows that we can greatly improve the current state-of-the-art, while also being able to solve much larger problems.

Some possible routes for future work include tuning the node evaluation function `NV`. For instance, we can include a weighted connectivity of the nodes. More specifically, we can compute a force-field, based on the task appearance probabilities and the locations of the agents. Other possibilities are to improve the MCTS search, e.g. by introducing node priors as for instance shown in [12].

Currently, we are working on deploying the approach on real robots. We are setting up a multi-robot approach that is inspired by the RoboCup@Work [18] competition. Multiple robots have to pick up items in the environment and bring them to a common depot node. A further idea is to use our work from [8] as a local collision avoidance mechanism that allows the robots to *share* the edges and nodes on a graph. As each node represents various compartments, we intent to rely on such a local low-level conflict resolution if multiple robots have to be at the same node. Additionally, we will investigate to what extent adding large penalties for occupying the same node will mitigate the problem of sharing nodes.

Acknowledgements.

F.O. is supported by NWO Innovational Research Incentives Scheme Veni #639.021.336.

REFERENCES

[1] S. Amador, S. Okamoto, and R. Zivan. Dynamic multi-agent task allocation with spatial and temporal constraints. In *Proc. of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1384–1390, 2014.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, May 2002.

[3] R. Bjarnason, A. Fern, and P. Tadepalli. Lower bounding klondike solitaire with monte-carlo planning. In *Proc. of the International Conference on Automated Planning and Scheduling*, pages 26–33, 2009.

[4] C. Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proc. of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 195–210. Morgan Kaufmann, 1996.

[5] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Trans. on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[6] J. Capitán, M. T. J. Spaan, L. Merino, and A. Ollero. Decentralized multi-robot cooperation with auctioned POMDPs. *International Journal of Robotics Research*, 32(6):650–671, 2013.

[7] H.-L. Choi, L. Brunet, and J. P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. on Robotics*, 25(4):912–926, 2009.

[8] D. Claes, D. Hennes, K. Tuyls, and W. Meeussen. Collision avoidance under bounded localization uncertainty. In *Proc. of the International Conference on Intelligent Robots and Systems*, pages 1192–1198, 2012.

[9] D. Claes, P. Robbel, F. A. Oliehoek, K. Tuyls, D. Hennes, and W. van der Hoek. Effective approximations for multi-robot coordination in spatially distributed tasks. In *Proc. of the International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 881–890. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[10] R. Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *Computers and Games: 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers*, pages 72–83. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[11] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proc. of the IEEE*, 94(7):1257–1270, 2006.

[12] S. Gelly and D. Silver. Combining online and offline knowledge in uct. In *Proc. of the 24th international conference on Machine learning*, pages 273–280. ACM, 2007.

[13] B. P. Gerkey and M. J. Mataric. Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 3862–3868. IEEE, 2003.

[14] S. Henn, S. Koch, and G. Wäscher. Order batching in order picking warehouses: a survey of solution approaches. In *Warehousing in the Global Supply Chain*, pages 105–137. Springer, 2012.

[15] B. Kartal, E. Nunes, J. Godoy, and M. Gini. Monte carlo tree search with branch and bound for multi-robot task allocation. In *The IJCAI-16 Workshop on Autonomous Mobile Service Robots*, 2016.

[16] M. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49(2-3):193–208, 2002.

[17] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer Berlin / Heidelberg, 2006.

[18] G. K. Kraetzschmar, N. Hochgeschwender, W. Nowak, F. Hegger, S. Schneider, R. Dwiputra, J. Berghofer, and R. Bischof. RoboCup@Work: Competing for the Factory of the Future. In *RoboCup Symposium*, 2014.

[19] P. Nijssen and M. H. Winands. Monte carlo tree search for the hide-and-seek game scotland yard. *IEEE Trans. on Computational Intelligence and AI in Games*, 4(4):282–294, 2012.

[20] M. L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[21] M. Quigley et al. ROS: An open-source Robot Operating System. In *Proc. of the Open-Source Software Workshop (ICRA)*, 2009.

[22] E. Schneider, E. I. Sklar, S. Parsons, and A. T. Özgelen. Auction-based task allocation for multi-robot teams in dynamic environments. In *Conference Towards Autonomous Robotic Systems*, pages 246–257. Springer, 2015.

[23] C. Seward. Accelerating warehouse operations with neural networks. `https://tech.zalando.de/blog/accelerating-warehouse-operations-with-neural-networks/`, 2015.

[24] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[25] J. Sleight and E. H. Durfee. A decision-theoretic characterization of organizational influences. In *Proc. of the International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 323–330, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[26] J. Wu and E. H. Durfee. Resource-driven mission-phasing techniques for constrained agents in stochastic environments. *Journal of Artificial Intelligence Research*, 38:415–473, 2010.