Decentralised Submodular Multi-Robot Task Allocation

Murtaza Rangwala

1 Motivation

The task allocation problem in Multi Robot Systems is NP-Hard in most of the cases. Centralised approaches to solving the task allocation problem, relies on each robot to communicate all the data to a central entity. With a centralised solution, we introduce a single point of failure. In order to circumvent this issue, decentralised approaches to solving the task allocation problem have been gaining popularity. In this paper, the problem of task allocation problem is addressed using a decentralised algorithm where each robot relies only on its own utility function, to improve the resilience of its network.

2 Problem Definition

In this paper, the algorithm proposed by the authors solves an adaptation of the classical Task Allocation Problem. For a set of tasks $\mathcal{T}$, and a set of agents $\mathcal{A}$, and for each agent $a \in \mathcal{A}$ a submodular utility (reward) function $f_a : 2^{\mathcal{T}} \rightarrow \mathcal{R}^+$. The aim is to find an allocation, $S^* \in \mathcal{A}$ for the agents such that no two agents share the same tasks, and which maximises the global utility function, $\mathcal{J} : A^{\mathcal{T}} \rightarrow \mathcal{R}^+$, defined as $\mathcal{J}(S) = \sum_{a \in \mathcal{A}} f_a(S_a)$.

3 Technical Approach

The algorithm tackles both the monotone submodular and non-monotonic submodular utility $f_a : 2^{\mathcal{T}} \rightarrow \mathcal{R}^+$. functions. A matrix x is defined such that it contains a fractional value between $[0,1]$, with each row corresponding to an agent and the columns corresponding to each task. This allocation is then constrained, such that each task is not allocated to more than one agent, $\mathcal{K} = \left\{x \in [0,1]^{|\mathcal{T}||\mathcal{A}|} \,\middle|\, \sum_{a \in \mathcal{A}} x_a(t) \leq 1 \; \forall \, t \in \mathcal{T}\right\}$.

A global relaxed allocation is calculated, based on the multilinear extension, given by $F(x) = \sum_{a \in \mathcal{A}} F_a(x_a)$, where $F(x) = \sum_{a \in \mathcal{A}} f(G) \prod_{G \in \mathcal{T}} x(t) \prod_{G \notin \mathcal{T}} (1 - x(t))$, where G(x(t)) is a random set containing each task $t \in \mathcal{T}$, with probability x(t). An additional step is then made to round the values of the fractional solution, which is guaranteed, to maintain the approximation of the fractional solution.

Finally, these allocations are exchanged between other agents to find the best allocation between them.

4 Related Work

For problems, that have a non-linear utility function, the problem becomes NP-hard. Due to this, most of the algorithms are heuristic in nature. Choi et al, in their paper *Consensus based decentralized Auctions for Robust Task Allocation* was the first paper, that had approximation guarantee $\frac{1}{2}$ for montone, positive value utility function, for the classical Task Allocation Problem, using the Sequential Greedy Algorithm.

5 Results

The authors prove that the solution is guaranteed to be within $\frac{1}{e}$ for a positive non-monotonic submodular function, and within $1-\frac{1}{e}$ for positive, monotone and submodular utility function. The experiments that were conducted were also conducted, where it was observed that the theoretical performance guarantees were obtained.

However, the experiments were not conducted on larger number of tasks due to the limitation of the MIP solver, *Gurobi Optimizer*, that was used.