Error-Bounded Approximation of Pareto Fronts in Robot Planning Problems – Supplementary Material –

Alexander Botros*, Nils Wilde**, Armin Sadeghi*, Javier Alonso-Mora**, and Stephen L. Smith*

* Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, ON, Canada
{alexander.botros, a6sadegh, stephen.smith}@uwaterloo.ca,
** Department for Cognitive Robotics, 3ME
Delft University of Technology, Delft, Netherlands
{n.wilde, j.alonsomora}@tudelft.nl,

A Implementation of Max-Min Neighbourhood Regret

Below we detail the implementation of the linear program in Equation (11). We formulate the convex hull constraint $\boldsymbol{w} \in \mathcal{C}(N)$ using a scalars $\lambda^1, \dots, \lambda^n \in [0,1]$ to write $\boldsymbol{w}$ as a convex combination of the neighboughood weights $\boldsymbol{w} = \lambda^1 \boldsymbol{w}^1 + \dots + \lambda^n \boldsymbol{w}^n$. Thus, the equality constraints are given by

$$\begin{bmatrix} 1 & & \dots & 1 & 0 & \dots & 0 \\ 0 & & \dots & 0 & 1 & \dots & 1 \\ -1 & 0 & \dots & 0 & w_1^1 & \dots & w_1^n \\ 0 & -1 & \dots & 0 & w_2^1 & \dots & w_2^n \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & & \dots & -1 & w_n^1 & \dots & w_n^n \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ \lambda^1 \\ \vdots \\ \lambda^n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{1}$$

The first row ensures that $\boldsymbol{w}$ is normalized, i.e., lies in $\mathcal{W}$ where all its components sum to 1. The second ensures the same for $\lambda^1, \dots, \lambda^n$. The other rows ensure that the i-th element of the vecotr $\boldsymbol{w}$ is a convex combination of the i-th component of all neighbourhood vectors $\boldsymbol{w}^1, \dots, \boldsymbol{w}^n$. In the objective function, we write $P(\boldsymbol{w})$ using the same convex combination as

$$x - \sum_{i=1}^{n} \lambda^i u(\boldsymbol{w}^i). \tag{2}$$

Finally, we require that $w_i \geq 0$ and $\lambda^i \geq 0$ for all $i = 1, \dots, n$.

B Simulation Results

In this section, we evaluate the performance of the proposed algorithm on a NP-hard optimization problem, namely traveling salesman problem with multiple agents.

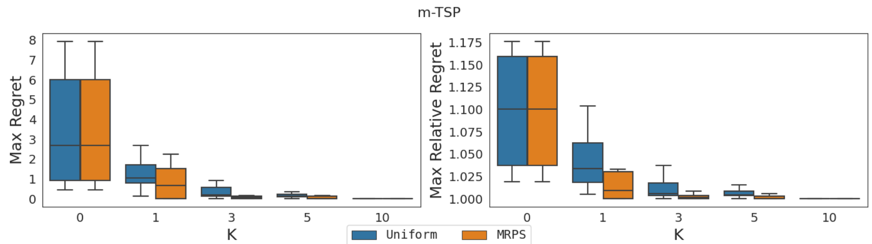

Fig. 1: Sampling trade-offs between min-sum and min-max tour lengths for mTSP.

Given a tour T in $G = (V, E, c)$, we denote the cost of the tour with $\text{cost}(T)$. Then the variation of the multi-traveling salesman problem is defined as follows:

Problem 1 (Multi Traveling Salesman Problem (mTSP)). Given a graph $G = (V, E, c)$, a set of m robots, the weights w_1, w_2, and a depot $d \in V$, the objective is to find a set of m tours, $T_1, \dots, T_m$, starting at d such that the linear combination of the total tour length and the maximum tour length is minimized, i.e.,

$$\min_{T_1,\dots,T_m} w_1 \sum_{i=1}^{m} \text{cost}(T_i) + w_2 \max_{i \in \{1,\dots,m\}} \text{cost}(T_i).$$

The two features, total tour length and maximum tour length, represent the total energy consumption of a fleet of robots to collectively service the tasks and the maximum time to service the tasks, respectively. Note that depending on the user-preferences or different scenarios, one of the features become more important. For instance, in a monitoring scenario with a set of robots, depending on the time and the state of the observed environment, the service time is more important than the total energy consumption of the fleet. Observe that finding the optimal solution of the mTSP problem for given w_1, w_2 is computationally expensive, therefore, changing the strategy based on the state of the environment in an online fashion is not feasible. Using the proposed approach, we generate set of candidate sample weights that minimize the maximum regret for any possible scenario or user preference. Figure 1 shows the results of experiment with 15 vertices to observe and 10 robots. The left figure illustrates the maximum regret using the samples with the proposed method and the uniform sampling on the weight set. Note that the maximum regret of the proposed method almost converges to 0 with 3 samples, while uniform sampling needs 10 samples to achieve the same. Overall the difference between the methods is larger on the maximum relative regret measure.