Reactive Architecture Memo

Scott Benson and Nils J. Nilsson

February, 1994

1 Action Selection and Cable Routing

We consider the problem of how a robot should choose actions when it has a number of different goals of different urgencies. Suppose the goals are $G_1, \ldots, G_i, \ldots G_N$, where each G_i is a condition to be made true by the robot. (It is not necessary that all the G_i be true simultaneously.) When goal G_i is achieved, the robot is rewarded by an amount R_i. The robot's strategy is to choose actions to maximize total discounted reward. Thus, at time $t = 0$, if the robot chooses to achieve goal G_i at time t_i, the present value of its reward for achieving that goal is $\gamma^{t_i} R_i$, where $\gamma < 1$ is the discount factor. The total discounted reward for achieving all of the goals is:

$$R_{total} = \sum_{i=1}^{N} \gamma^{t_i} R_i$$

We assume the robot knows, for any order of achieving goals, the times t_i at which it would achieve each goal.

For γ close to 1, we can approximate R_{total} by expanding $\gamma^{t_i} = 1 + t_i \log \gamma + (t_i \log \gamma)^2/2! + \ldots$ and dropping the high-order terms:

$$\hat{R}_{total} = \sum_{i=1}^{N} (1 + t_i \log \gamma) R_i$$

We define $\beta = \log \gamma$ and since $\gamma < 1$, $\beta < 0$. Therefore, to maximize $\hat{R}_{total}$, we should minimize $\sum_{i=1}^{N} t_i R_i$ by appropriate choice of the order of goal achievement.

Minimizing $\sum_{i=1}^{N} t_i R_i$ is also known as the "cable-routing problem." (See reference.) We have a cable bundle that begins at a position P_0 and must visit N other positions $(P_1, \ldots, P_i \ldots, P_N)$ in some sequence, dropping off R_i cables at place P_i. The total distance traveled by the bundle to place P_i is t_i. In cable routing, we want to minimize the total amount of cable.

2 A Greedy Algorithm

We propose the following greedy algorithm for minimizing $\sum_{i=1}^{N} t_i R_i$:

After visiting (achieving) goal G_i (and getting reward R_i), we next work on that goal G_j for which R_j/t_{ij} is largest (over all j), where t_{ij} is the estimated time it takes to go from where we are now (having just achieved G_i) to goal G_j. "Working on" goal G_j means that we consider the T-R program for goal G_j and perform the action, whatever it is, that is specified by that program. That is, we always work on next whatever is estimated to give us the greatest benefit/cost ratio.

3 Reference

Blum, A., *et al.*, "The Minimum Latency Problem," *Proc. of STOC-94*, pp. 163-171, Association for Computing Machinery, 1994.