
COMP0037 ASSIGNMENT 3

Group: Group L

Members: Yun Fang, Yusi Zhou

April 28, 2020

1 Decision Re-Plan Policy

Let T_W be the time the robot has to wait after the obstacle has been discovered.

Note that:

T is the number of timesteps the obstacle remains in front of the robot.

L_W is the cost of waiting a timestep.

1.1

Let B_1 be the cell where the robot first detects the obstacle. If the robot decides it will wait, then on average the path cost of waiting at aisle B is less than or equal to planning a new path down aisle C. So:

$$\begin{aligned} \mathbb{E}(L_{B_1BG} + \text{waiting cost}) &\leq \mathbb{E}(L_{B_1CG}) \\ L_{B_1BG} + \mathbb{E}(T_W) * L_W &\leq L_{B_1CG} && \Leftarrow \text{the path length and } L_W \text{ are constants} \\ \mathbb{E}(T_W) &\leq \frac{L_{B_1CG} - L_{B_1BG}}{L_W} \\ \frac{1}{\lambda_B} &\leq \frac{L_{B_1CG} - L_{B_1BG}}{L_W} && \Leftarrow \mathbb{E}(T_W) = \mathbb{E}(T) = \frac{1}{\lambda_B} \\ \lambda_B &\geq \frac{L_W}{L_{B_1CG} - L_{B_1BG}} \end{aligned}$$

Therefore, the smallest value of λ_B on average that waiting is the better strategy is $\frac{L_W}{L_{B_1CG} - L_{B_1BG}}$.

1.2

If starting at I and the robot decides to drive down aisle C directly, then on average the path cost via B is greater than that via C, so:

$$\begin{aligned}
 \mathbb{E}(L_{IBG} + \text{waiting cost}) &> \mathbb{E}(L_{ICG}) \\
 L_{IBG} + \mathbb{E}(T_W) * L_W &> L_{ICG} && \Leftarrow \text{the path length and } L_W \text{ are constants} \\
 \mathbb{E}(T_W) &> \frac{L_{ICG} - L_{IBG}}{L_W} \\
 \frac{1}{\lambda_B} &> \frac{L_{ICG} - L_{IBG}}{L_W} && \Leftarrow \mathbb{E}(T_W) = \mathbb{E}(T) = \frac{1}{\lambda_B} \\
 \lambda_B &< \frac{L_W}{L_{ICG} - L_{IBG}}
 \end{aligned}$$

Therefore, the maximum value of λ_B at which the robot will decide to drive directly down C is $\frac{L_W}{L_{ICG} - L_{IBG}}$.

1.3

The event that the robot has to wait after the obstacle has been discovered is now separated into two case: one is the obstacle presents, which has the probability p_B , and the robot remains alive for T time steps; the other one is the obstacle does not presents, which has the probability $(1 - p_B)$. And in the latter case the time that the robot remains alive is 0. Due to that the event the obstacle presents or not and that the time the obstacle remains active are independent, $T_W = p_B * T + (1 - p_B) * 0 = p_B * T$.

Then $\mathbb{E}(T_W) = \mathbb{E}(p_B * T) = p_B * \mathbb{E}(T) = p_B / \lambda_B$.

If the the robot attempt to drive aisle B first, then it means:

$$\mathbb{E}(L_{IBG} + \text{waiting cost}) < \mathbb{E}(L_{ICG})$$

Similarly, we can obtain that:

$$\begin{aligned}
 \frac{p_B}{\lambda_B} &< \frac{L_{ICG} - L_{IBG}}{L_W} \\
 p_B &< \frac{\lambda_B \cdot (L_{ICG} - L_{IBG})}{L_W}
 \end{aligned}$$

Therefore for a fixed value of λ_B if the value of p_B is below $\frac{\lambda_B \cdot (L_{ICG} - L_{IBG})}{L_W}$ the robot will attempt to drive aisle B first.

1.4

In this situation if the robot computes that the path via aisle D is the shortest, it means that:

$$\mathbb{E}(L_{IDG}) < \mathbb{E}(L_{IBG} + T_{WB} * L_W) \quad (1)$$

$$\mathbb{E}(L_{IDG}) < \mathbb{E}(L_{ICG} + T_{WC} * L_W) \quad (2)$$

Where

$$\mathbb{E}(T_{WB}) = p_B / \lambda_B$$

$$\mathbb{E}(T_{WC}) = p_C / \lambda_C$$

By summing (1) and (2) together:

$$\begin{aligned}2 \cdot \mathbb{E}(L_{IDG}) &< \mathbb{E}(L_{IBG} + T_{WB} * L_W) + \mathbb{E}(L_{ICG} + T_{WC} * L_W) \\2 \cdot L_{IDG} &< L_{IBG} + \mathbb{E}(T_{WB}) * L_W + L_{ICG} + \mathbb{E}(T_{WC}) * L_W \\2 \cdot L_{IDG} &< L_{IBG} + L_{ICG} + L_W \cdot (\mathbb{E}(T_{WB}) + \mathbb{E}(T_{WC})) \\2 \cdot L_{IDG} &< L_{IBG} + L_{ICG} + L_W \cdot (p_B/\lambda_B + p_C/\lambda_C) \\L_{IDG} &< 0.5L_{IBG} + 0.5L_{ICG} + 0.5L_W \cdot (p_B/\lambda_B + p_C/\lambda_C)\end{aligned}$$

Therefore, if the path via aisle D is the shortest path then the upper bound on the value of the path length is $0.5L_{IBG} + 0.5L_{ICG} + 0.5L_W \cdot (p_B/\lambda_B + p_C/\lambda_C)$.

2 Implement System in ROS

2.1

We first add a function in the class which returns a cell coordinate as the intermediate destination, according to the aisle passed in. Then in the *planPathToGoalViaAisle()*, we call this function to get the intermediate cell coordinate and search a path from the given start cell to the intermediate cell. If there was a path then we extract and store this path and then search the second path which is from the intermediate cell to the given goal cell. If the second path existed, we extract it and link it to the first one using the provided function *addToEnd()*. At the end of this function we call the *searchGridDraw* to show the first path so that the two paths can be seen simultaneously.

The result is shown in Fig. 1. The start point is marked purple, the intermediate cell is marked green, and the goal is marked blue.

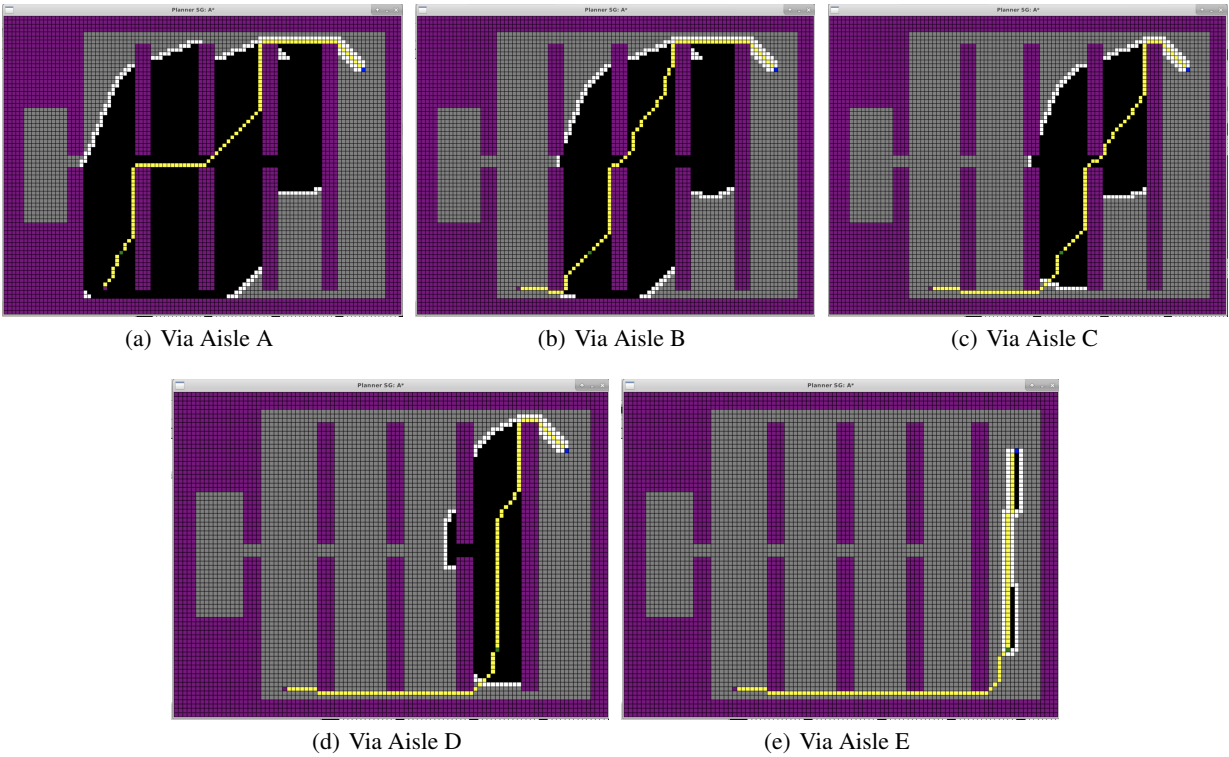


Fig. 1. Planned Routes Down all the Different Initial Aisles

2.2

We first implement the *waitUntilTheObstacleClears()*. To meet the requirements we first find the obstructed cell in the current path waypoints. Then the function keeps checking the cell status with an infinite loop, and returns only if the status has been updated and became empty.

To determine whether the robot should wait or not, we need to compute the travel cost of the remaining original path and the new planned path from the current position. For the latter we can simply use the function *planPathToGoalViaAisle()* to re-plan a new path and then obtain the travel cost from the provided path property *path.travelCost*. Coming to the remaining original path cost, we need to get the travel cost of the path from original start cell to current position first. Then what we want is the travel cost of the current (or original) path minus it.

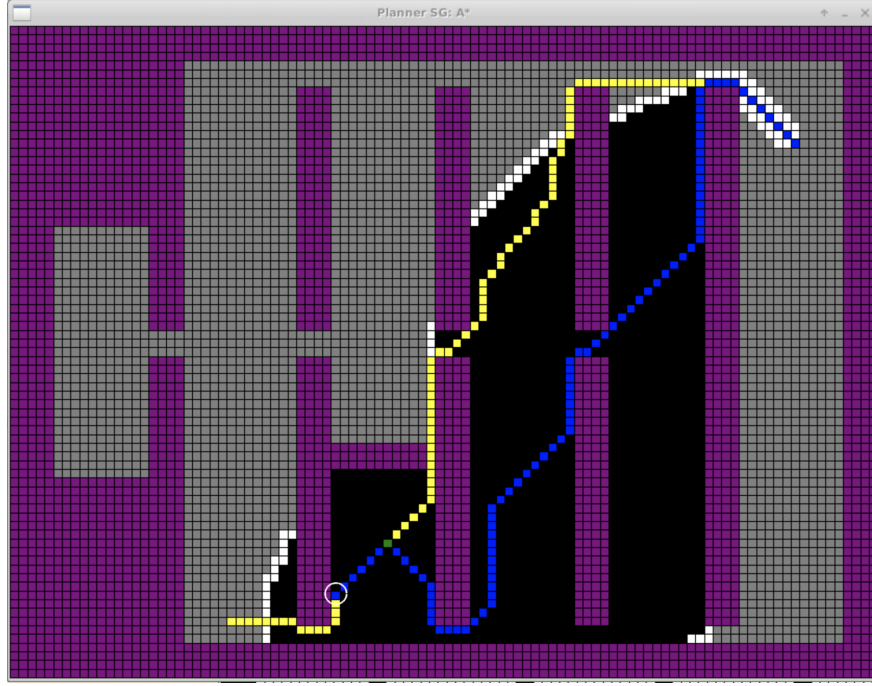
At the end of the function we calculate the threshold value of the expected $T_W (= \frac{\text{remaining original path cost} - \text{new path cost}}{L_W})$ and also the corresponding $\lambda_B (= 1/\mathbb{E}(T_W))$, and output to terminal. If the T_W we feed in is less than or equal to the threshold value, then the function *shouldWaitUntilTheObstacleToClear()* returns true.

```

remaining original Path Cost: 99.0832611207
New Path Cost: 112.053823869
threshold  $T_W$ : 6.48528137424
threshold  $\lambda_B$ : 0.154195314327

```

(a) Data



(b) The search grid. The robot detects the obstacle when it reached the circled cell.

Fig. 2. Result of a single run

Fig.2 shows the result of a single run of our implementation. From the Fig.2(a) we can see that the new path cost is 112.0538 while the remaining original path cost (excluding the waiting time) is 99.0833. If the wait time we feed in is less than or equal to the threshold T_W (6.4853) then the robot will wait until the obstacle clear, else the robot will drive down aisle C. In other words, the minimum value of T_W to make the robot move is 6.4853, and thus the maximum value of λ_B below which the robot will move if it encounters the obstacle is 0.1541

2.3

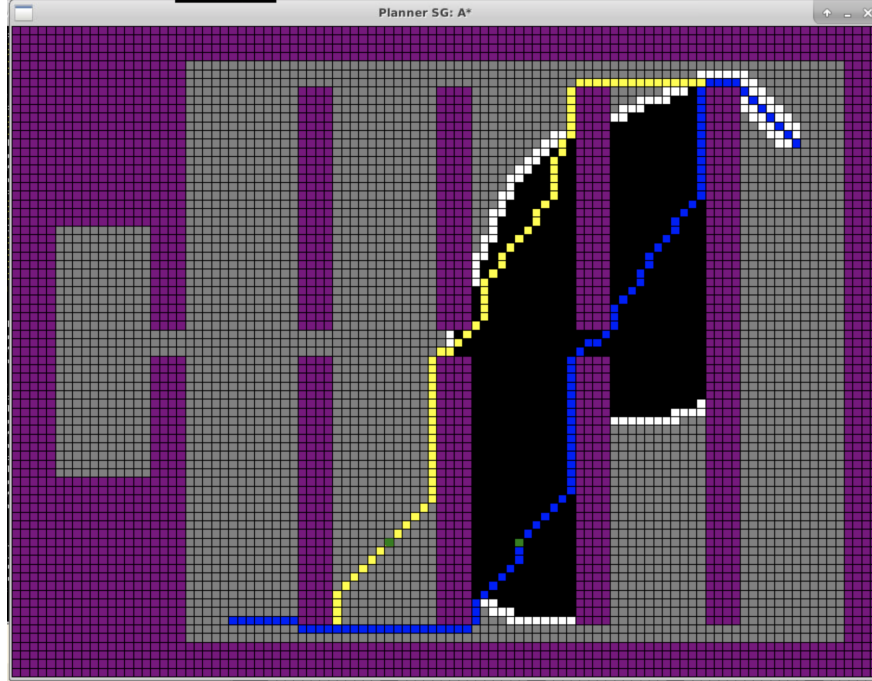
We use the function *planPathToGoalViaAisle()* to plan the paths via aisle B and via aisle C. We only consider the two paths here due to that there is only a single active obstacle and it always presents in aisle B. The travel cost is obtained by using the provided path property *path.travelCost*. We then calculate the threshold expected $T_W (= \frac{\text{pathCost}_C - \text{pathCost}_B}{L_W})$ and the corresponding $\lambda_B (= p/\mathbb{E}(T_W))$ and output the values to terminal. With the threshold values, the waiting time is then input by users from terminal to determine which path the robot will choose. If the waiting time greater than or equal to the threshold value then the robot will go down the aisle C.

```

pathCost_B: 114.911688245
pathCost_C: 115.497474683
threshold expected  $T_w (= p / \lambda_B)$ : 0.292893218813
threshold  $\lambda_B$ : 2.7313708499

```

(a) Data



(b) The search grid. The yellow path is the path via aisle B, and the blue one is via aisle C.

Fig. 3. Result of a single run of our implementation

Fig.3(a) shows the values of travel cost of the two paths of a single run(excluding the waiting cost if it encounters an obstacle at aisle B) and the values of the threshold of the waiting-time-related parameters. There is little difference between the travel cost of the two paths if we do not take the waiting time into consider. The threshold expected T_W tells that if the waiting time is greater than or equal to 0.2928, then the robot will go directly down aisle C. So the maximum value of λ_B is 2.7313.