Robot Vacuum Cleaners

Coursework Report - COMP3004 Designing Intelligent Agents

The core idea of this project is to (1) investigate how two types of agents: wandering bots (WB) and planning bots (PB) compare in the task of vacuuming an environment. The agents are autonomous drivers which move across an environment mapped with dirt via wheels in a differential drive format, collecting dirt that they pass by. The WB move randomly, using some low-level reactive intelligence to switch between random angles of direction. The PB are agents with memory and intention, storing a map of the environment and using it to influence their behaviour by calculating the best A\* path through the map from point (9,9) to (0,0) that will allow them to collect the most amount of dirt. Previous studies have shown that robots which plan out their task (PB) perform more efficiently than those which execute the task randomly (WB), however, these PB also come at a much higher cost due to computational complexity, etc. (Galceran and Carreras, 2013). Nevertheless, more recent studies have highlighted that bots which follow an optimal path will complete their task requiring less computational power and time due to high coverage rates and low repetition rates (Pham and Lam, 2019). This project will reveal whether the improved performance of the PB is significant enough to compensate for the financial aspects of producing and purchasing them. The project also aims to (2) understand how the performance of the two agents is affected when a battery functionality and a bin functionality are introduced to the system, i.e., when the agents are limited by the amount of moves they can take before needing to charge, and dirt they can store before needing to unload at a bin. Introducing these two new functionalities will convert the agents to subsumption agents who keep internal battery and storage states and react to them by allowing them to trigger different behaviours: vacuuming, seeking a charger, seeking a bin.

To experiment with questions (1) and (2) defined above, an environment had to be designed for the vacuuming task to be executed in and for the bins and chargers to be added to. This environment consisted of a simple quadratic area, divided into 10x10 sections with a random distribution of dirt mapped onto each section, some sections containing more dirt than others. Furthermore, for the experiments where the agents were limited by their levels of battery and storage, the chargers and bins were placed in random locations on the map. The performance of the agents was then measured as the amount of dirt they collected in total in each run. Both agents were tested with all possible combinations of using and not using batteries and bins, as shown in Table 1 below. In the experiments with the battery functionality, the bots were initialised with a battery-level of 1000, which was decremented as they moved, and the bots would start seeking a charger after their internal battery state had dropped below 600. If it had dropped below 0, they would come to a halt. In the experiments with the bin functionality, the bots were initialised with a storage-level of 50, which was decremented as they collected dirt, and the bots would start seeking a bin after their internal storage state had dropped below 35. If it had dropped below 0, they would continue moving but without collecting anymore dirt until they had unloaded at a bin.

### Table 1: Combinations tested for usage of batteries and bins

Table

Description automatically generated

The system was developed using Python with the tkinter library for graphics, the matplotlib.pyplot library for data visualisation, and the scipy.stats library for performing statistical tests. Some challenging aspects of the development included (A) randomly distributing the dirt in a reproducible manner and (B) designing the experimental setup to test insightful parameters with meaningful values.

A solution to the first issue (A) was seeding the runs of the experiments since the seed would ensure that the random distribution of dirt remained the same. The seed would also ensure that the results would be reproducible. This is because WB with the same seed will chose the same random movements, and PB in environments with the same random distribution of dirt will pick the same best path from point (9,9) to (0,0) on the map since they use the deterministic A\*- algorithm to find this path.

For the second issue (B), every possible combination of using and not using a battery and bin functionality was experimented with for both bots (as explained previously and as shown in Table 1), to ensure that the results were comprehensive and conclusive. For each combination, several different random distributions of dirt were tested to be able to calculate insightful aggregate values (mean values), and t-tests were performed to discover the statistical significance of the experiments. Moreover, each experiment was built with a number of static variables which were set by trial-and-error testing aiming to reveal the values which would give the most meaningful results. Examples of limitations that the trial-and-error testing aimed to overcome include:

* The number of bots in each run had to be limited to 1 to create a fair comparison between the two types of bots. This was because 2 WB in the same environment would move randomly and differently and collect more dirt than just 1 WB. However, 2 PB in the same environment would choose the same path as they would both calculate it using the deterministic A\* algorithm, thus collecting the same amount of dirt.
* Time was not explored as a dynamic parameter as it can be assumed that the more time there is, the better the PB will perform since the WB will most probably re-visit sections of the environment. The less time there is, the more equally the agents will perform as the PB’s strategy won’t have time to give them a significant enough advantage since most areas will be un-visited in the beginning. The static amount of time chosen in the end therefore had to be balanced between both extremes. It also had to ensure that the system was stopped before the PB finished its path to make for a fair comparison between the bots, otherwise the WB would continue moving and collecting dirt after the PB had finished its path and stopped.
* The charging limit had to be low enough for the bots to reach it – otherwise, the introduction of a battery functionality would not affect the performance of the bots as they would never have to seek the charger, i.e., switch behaviour.
* The logic above can be applied to the unloading limit as well.

Details of the final experimental setup can be seen in Table 2 below. In total, there are 4 types of experiments for each of the 2 types of bots (WB and PB) w.r.t battery and bin usage (TT, TF, FT, and FF, as shown in Table 1), resulting in 8 experiments in total. Each experiment was repeated 70 times, each time with a new seed and thereby a new random distribution of dirt – a list of 70 random seeds was stored to be reused for each experiment for reasons explained earlier w.r.t experiment-reproduction. Thus, the table of results included 560 data points in total, with 70 rows and 8 columns. For clarity, a sample of the first 10 rows of this table may be seen in Table 3 below (the complete table of results may be seen in the file ‘data.xlsx’ in the folder ‘coursework.zip’).

### Table 2: Experimental setup

A picture containing scatter chart

Description automatically generated

### Table 3: Results for the first 10/70 runs of each of the 8 experiments

A picture containing calendar

Description automatically generated

The results in Figure 1 below show the data distribution of all 70 runs of each of the 8 experiments. Each distribution has been annotated with three variables, e.g., ‘w,T,T’, where:

* The first variable represents the strategy (i.e., type of bot) used: wandering *w* vs. planning *p*
* The second variable represents the usage of a battery functionality: true *T* vs. false *F*
* The third variable represents the usage of a bin functionality: true *T* vs. false *F*

According to Figure 1, both bots seem to have an equally large and consistent distribution of data points with relatively few outliers – only 4 with particularly high values were observed in the experiments without a battery functionality. The trend in boxes shows that WB predominantly collect around 10 to 20 pieces of dirt while the PB interval is the same for experiments with a bin functionality but increases to around 20-30 for the experiments without it. This observation indicates that the introduction of a bin functionality drastically lowers the performance of the PB, which is ordinarily much higher than that of the WB.

### Figure 1: Experiment-distributions

Chart, box and whisker chart

Description automatically generated

Figure 2 below shows the mean values of each of the 8 experiments, calculated from the results of all their 70 runs. This aggregate function shows the trend in the agents’ behaviour. Both bots collect the *least* amount of dirt when using both the battery *and* the bin functionality (T,T), which is to be expected since this setting limits the bots the most as they are forced to seek both a charger and a bin in the run, thus removing the focus from the task of collecting dirt and possibly forcing them to revisit cleaned areas. Since the PB aim to complete the task optimally instead of just randomly, they are the most affected by using both functionalities – essentially demoting them from agents with intention and memory to reactive state agents. This is why the performance of the PB is lowered to the same level of the WB for this experiment.

In alignment with the statement above, the bots’ second-worst ranked experiment was initially expected to be the ones where *one* limitation was used (i.e., FT or TF: using either a bin or a battery system). However, Figure 2 showed that the two functionalities seemingly affected the bots’ performance differently – evidently, introducing just the bin functionality (FT) lowered the bots’ performance by more than introducing just the battery functionality (TF) did. This may be because charger-seeking bots can still collect more dirt on their way to the charger, but bin-seeking bots can’t collect anymore dirt on their way to a bin if their available storage level has dropped below 1. Admittedly, charger-seeking bots can’t collect anymore dirt on their way to a charger if their battery level has dropped below 1 either, but this case is comparatively infrequent. Therefore, limiting bots’ storage-level has a more negative effect on their performance than limiting their battery-level. This phenomenon may be observed to a greater magnitude with the PB since they have more ‘intelligence’ which is interfered with in these cases, explaining the distributions in Figure 1.

Previous outcomes have created an expectation for the bots to perform the best with no limitations at all. This was true for the WB, however, the PB unexpectedly performed even better (by a small margin) in the experiment with just the battery functionality added (comparing ‘p,F,F’ with ‘p,T,F’). This may be because the PB looks for the best path from the bottom right corner of the map to the top left corner but regions far from those may have a lot of dirt which the PB may only discover when forced to seek out a charger which may be anywhere on the map. The same effect was not observed with just the bin functionality due to the weaknesses of it explained earlier. Regardless, the PB perform almost equally as effectively in the experiment without either functionality since they already know the best path which allows them to collect the most dirt and any interference with this behaviour forcing them to deviate from this path, will likely not increase their performance. This is because they won’t recalculate the best path from their new location after charging or unloading, but instead go back to where they deviated from the old path and continue along it which may not be the most optimal path anymore. That may be why adding limitations to the WB didn’t significantly lower their performance as their movement wasn’t meaningful to begin with. However, the limitations might still cause the WB to waste time re-visiting areas around the charger/bin if they keep having to go there to charge/unload, explaining why their performance is still lowered by a small amount with the introduction of limitations.

Interestingly, there is no experiment where the two agents perform equally, meaning that for all the possible settings in this set of experiments, there is not even one in which the performance of the PB is lowered to the level of the performance of the WB.

### Figure 2: Mean values of all experiments

Chart, bar chart

Description automatically generated

To validate the experimental results in Figures 1 and 2 above – rejecting the null hypothesis claiming that the results were obtained by chance – a set of 4 t-tests was performed: one for each experiment. The t-test is a statistical test aimed to discover the significance of the difference in results between two groups: WB and PB in this case. The results of the t-tests can be seen in the grouped bar chart below, Figure 3, where the blue bars represent the t-value for each test and the orange bars represent the p-value. The higher the t-value is, the more different the results for the two bots are in that experiment. In this study, the results of the t-tests indicate that the WB performs ~1-2 times as differently from the PB in the experiments with a bin functionality and ~5-7 times as differently in the ones without it since the removal of the bin functionality significantly increases the performance of the PB, as explained earlier. These t-values indicate that the results are relatively repeatable for the experiments with the bin functionality and highly repeatable for the ones without it (the same trends may be visualised if re-running the experiments with new values). The lower the p-value is, the more significant the difference between the performance of the bots is – the p-value is namely the probability that the results occurred by chance. The results in Figure 3 indicate that the experimental results of this study are not significant for the experiment with both functionalities (TT), which is to be expected due to the high level of interference with the bots’ intelligence in that experiment. Furthermore, the results show that the experiment with just the storage-limitation (FT) had somewhat significant results and the experiment without it (regardless of the battery-limitation) (TF and FF) had extremely significant results. As a note, p-values shown as 0 in the bar chart are simply rounded up from extremely small values, and the threshold used for significant p-values is values below 0.05 which is a widely used threshold in the field of statistics. Generally, the results of the t-test showed that the experiments with the bin functionality resulted in smaller differences between the two bots and less significant results, which is in alignment with the results from Figures 1-2 and the limitations of this functionality: interfering too much with the bots’ behaviour and especially the intention of the PB.

### Figure 3: T-test results per type of experiment

Chart, bar chart

Description automatically generated

In conclusion, the experimental results showed that A\* planning bots performed better in the task of vacuuming an environment, compared to randomly wandering bots, for all environmental circumstances. Furthermore, the introduction of functionalities in the form of battery and storage limits seemed to interfere with the agents’ behaviour and lower their performance in the task. The storage limitation was revealed to have the most drastic effect of that sort, particularly w.r.t the PB. In summary, PB are more effective and efficient robot vacuum cleaners and their computational complexity and higher prices are worth it.

To make the system more realistic and the bots more reactive, the bots should be able to take moving objects (such as pets or people) in the environment into consideration and avoid bumping into them. Furthermore, considering how realistic the system is, the bots should also disregard the toroidal geometry currently implemented, where the bot joins the other edge if it has run off the edge, and instead use their stored map of the environment to curve to the side whenever they’re approaching an edge. Currently, the number of bots in the experimental setup is hard coded as 1 due to the limitations of introducing more than 1 PB to the system, as mentioned earlier. As an improvement for scalability purposes, the PB could coordinate their movements by sending feedback to each other, ensuring they won’t visit revisit locations already covered in other bots’ paths. Furthermore, a new solution should ensure that the PB recalculate the best A\* path if they have had to deviate from the old path to go to a charger or bin due to previously mentioned limitations in this regard. Critical decision-making should also be implemented for the bots to allow them to prioritise dirt collection despite possible low levels of available battery or storage: if there’s enough battery and storage to complete the current vacuuming session, the bots should not switch to charger or bin seeking behaviour. Realistically, this is a better solution from a user-perspective since the bots can always be charged or unloaded after they’ve finished vacuuming. As a future extension, one could investigate how the PB are affected by data noise by randomly changing the *actual* map of dirt in the environment by different percentages from the map stored in the PB (the map which they use to calculate their A\* path from), and measure how much this affects their performance.

## Sources

Galceran, E. and Carreras, M., 2013. A survey on coverage path planning for robotics. Robotics and Autonomous Systems, [online] 61(12), pp.1258-1276. Available at: <https://www.sciencedirect.com/science/article/pii/S092188901300167X> [Accessed 20 June 2021].

Pham, H. and Lam, T., 2019. A new method using knowledge reasoning techniques for improving robot performance in coverage path planning. International Journal of Computer Applications in Technology, [online] 60(1), p.57. Available at: <https://www.inderscienceonline.com/doi/abs/10.1504/IJCAT.2019.099503> [Accessed 20 June 2021].