Emotion Inspired Cognitive Architecture for Robotic Adaptive Path Planning

Zheming Zhang, Will N. Browne, and Dale A. Carnegie
School of Engineering and Computer Science, Victoria University of Wellington, New Zealand
tonyzhang@ecs.vuw.ac.nz, will.browne@ecs.vuw.ac.nz, dale.carnegie@ecs.vuw.ac.nz

Abstract

Adaptable navigation is critical to extend the range of applications for mobile robots in daily life. An ideal architecture of mobile robots should adapt its path-planning methods to various navigation scenarios. But traditional architectures are static, plus they often need hand-coded prior knowledge (e.g. tuned hyperparameters for targeted scenarios) to ensure the path-planning methods function well. This paper proposes an adaptive architecture through a hyperparameter-adjustment approach for robotic path-planning tasks. The architecture can automatically learn adaptive path-planning knowledge, termed *Planning-Action-Outcome Contingency* (*PAOC*), which is inspired by emotion theories in cognitive neuroscience. PAOC knowledge is learned in a pattern of human interrogable "if-then" rules by an *Accuracy-based Learning Classifier System* (*XCS*) algorithm. Navigation simulations of a mobile robot were conducted within 31 differing scenarios. Results show the proposed architecture achieved adaptive path-planning by automatically learning PAOC patterns in all the scenarios. These PAOC patterns also provide visual interpretations regarding what the robot perceives in the scenarios.

1 Introduction

Adaptation within a real-world navigation task is important for mobile robots as diverse scenarios require different adjustments (e.g. parameter adjustment). Navigation adaptation refers to a robot's path-planning adjustment to varing scenarios, each of which includes a navigation environment, a navigation task, and the robo t's admissible behaviours to fulfil the task in the environment. Navigation adaptation is challenged by uncertainty within scenarios such that traditional approaches require prior knowledge to handle the uncertainty. This required prior knowledge may include knowledge about uncertainty-causing factors, the effects of these factors, and mappings and contingencies between the factors and the effects. This prior knowledge that can be hand-coded tediously by an engineer is limited and thus restricts a mobile robot's navigation success.

The existing planning algorithms have limited navigation adaptation. The path-planning algorithms can be traditionally categorised into two methods: classical methods and heuristic methods [Elbanhawi *et al.*, 2013; Mac *et al.*, 2016]. The classical methods are deterministic so have little navigation adaptation. In contrast, the heuristic methods increase navigation adaptation through a path-selecting adjustment approach. In the path-selecting adjustment approach, heuristic methods could select an optimal path according to a current condition when multiple paths are generated. However, the path-selecting adjustment approach of the heuristic methods does not properly adapt to scenarios unless prior knowledge (e.g. hyperparameters) are introduced into the navigation architecture by humans.

A hyperparameter of a method is a parameter that has determining effects on the method's performance. A hyperparameter is beyond methods and implicatations that it is applied to. For example, *inflation radius* is a hyperparameter of path-planning methods, and it can be applied for A-star algorithm and other path-planning methods. This hyperparameter will directly affect the perceived map that the path-planning method works on (see Section 3.1). Traditionally, it is the engineer's responsibility to tune hyperparameters scenario by scenario to ensure the path-planning method can generate paths. When these hyperparameters are finalised, the calculation process of a path-planning algorithm is deterministic. If these hyperparameters are not fit for a new scenario, the deterministic path-planning calculation would not necessarily generate a valid path. In the worst cases, a valid path could not be generated simply because these hyperparameters are inappropriate for the new scenario. For example, in scenario No.4 (see Figure 7.b and Figure 1.c), the *inflation radius* with a value of 0.8 metres is appropriate for the wide-open area on top right corner of the local zoom map, but this value will lead to an enclosed perception of the doorway (see Section 3.1).

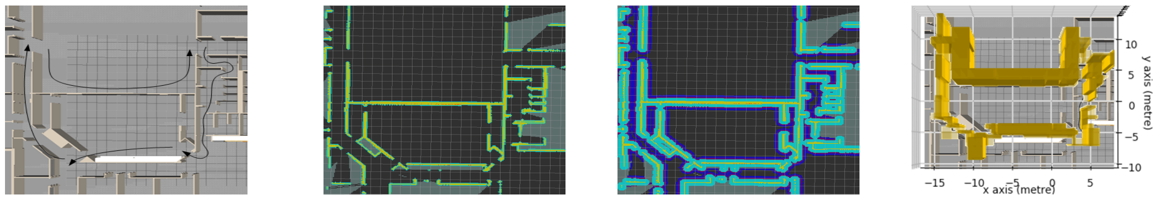
A novel hyperparameter adjustment approach is proposed to overcome the above deficiencies. The hyperparameter adjustment approach aims to guarantee valid paths in each scenario, if such paths could exist when hyperparameters are fine-tuned. The approach will adjust the hyperparameters of a path-planning method to changing or new scenarios. The adjustment requires knowledge about mappings and contingencies between hyperparameters and scenarios. The hyperparameter adjustment also requires a reasoning mechanism that can automatically learn the required knowledge.

For the hyperparameter adjustment approach, emotion theories provide inspiration for a representation of the required knowledge and a reasoning mechanism for learning this knowledge. Firstly, *Planning-Action-Outcome Contingency* (*PAOC*) is proposed as a cognitive architecture, a scenario-and-hyperparameter-encapsulated knowledge, to represent the required knowledge of the hyperparameter adjustment approach. PAOC originates from emotion theories: *appraisal theory* and *constructive theory*. The *appraisal theory* treats emotion as a reaction to events through assessments during *Planning* processes [Scherer *et al.*, 2010; Schlesinger and McMurray, 2012]. In the *Planning* processes, elements of a scenario (e.g. a goal of a task) will be assessed for emotional appraisal processes [Sequeira *et al.*, 2014]. The *Planning* is adopted as the first part of PAOC to encapsulate key elements from a scenario for assessment. On the other hand, *constructive theory* focuses on an *Action-Outcome contingency* (*AOC*) to increase flexibility. After the establishment of AOCs, actions and behaviours in AOC can be linked to "labels" and thus be considered as symbolic behaviours. The benefit of symbolic behaviour is that it may increase the flexibility of agents in the actual actions that are produced [Rolls, 2013]. Because the flexibility is necessary for the adaptation, the AOC is also adopted in PAOC to increase the flexibility and hence the adaptation of the agent, namely, a robot. Therefore, inspired by the *appraisal theory* and the *constructive theory*, this paper advocates that PAOC is an appropriate cognitive architecture that a mobile robot needs for navigation adaptation.

The second inspiration from emotion theories relates to how a reasoning mechanism works, e.g. *instrumental learning theory*. The *instrumental learning theory* indicates that a reasoning mechanism should work through interactions between an agent and its environment. As a result, the agent would learn to associate their behaviour with reinforcers or outcomes [Rolls, 2013].

Learning Classifier Systems (*LCSs*), a machine learning technique, learns knowledge through interactions as the instrumental learning theory suggests. An *LCS* is a rule-based technique that learns "if-then" rules through interactions between the learning agent and its environment. The *LCS* rules are updated based on a *reinforcement learning* process that assigns rewards from the environment. Based on various update criteria of the *reinforcement learning* process, the *LCS*s have two major categories: *Accuracy-based Learning Classifier System* (*XCS*) and *Strength-based Learning Classifier System* [Urbanowicz and Moore, 2009; Urbanowicz and Browne, 2017]. *XCS* agents update the worth of the rules based on the accuracy of the prediction of these rewards. In contrast, *Strength-based Learning Classifier System* agents' updates are based on the value of the rewards. A *XCS* agent is to be adapted to generate accurate, maximally general rules to represent PAOC knowledge such that the learned rules could apply the same hyperparameters to similar scenarios, where they are appropriate. However, the *standard approach* of *XCS* tends to generate overgeneral rules (see Section 3.5). Therefore, an amendment to introduce an additional *accuracy pressure*, termed the *mitosis approach*, is needed for a better algorithm (see Section 4).

The aim of this paper is to develop an emotion-inspired cognitive architecture for adaptive path-planning tasks for a mobile robot. The first objective of this paper is to provide a cognitive architecture, PAOC knowledge, that will be beneficial to adaptive path-planning tasks through the hyperparameter adjustment approach. PAOC knowledge will guarantee a valid path generation from a path-planning module. The learned PAOC knowledge, which is to be achieved by a reasoning mechanism, will replace hand-coded mappings that could introduce human bias. The second objective is to apply an accuracy pressure to *XCS* in the reasoning mechanism for the robot to learn PAOC knowledge. The novel *mitosis approach* (see Section 4) will introduce accuracy pressure into the standard approach of the *XCS* algorithm to amend over-generalization pressure during the algorithm's evolutionary processes. The mitosis approach will boost the *XCS* algorithm's learning ability in learning PAOC knowledge in terms of prediction accuracy, pattern robustness and pattern accuracy. The third objective is to interpret the learned, qualified PAOC knowledge. After visualisation, the qualified PAOC knowledge can provide insight into what the robot perceived in the navigation scenario.



(a) Ground Truth Map (b) Robotic Perception Map 1 (c) Robotic Perception Map 2 (d) Representation from Learned Knowledge

Figure 1: Different Representations of Willow Garage Office Environment.
(a) Environment Viewed in *Gazebo* (targeted navigation path is illustrated by black arrows); (b) Inflation Radius (IR) = 0.1 m; (c) IR = 0.8 m, (maximum value in this paper.) (IR = 5.5 m is the default value of the navigation package); (d) Learned IR Patterns (the yellow cuboids, see Section 5.3), for target path (a).

2 Related Work

Emotions have been shown to be beneficial to robotic navigation scenarios due to the similarity between robots and biological agents in the decision-making process and the reward-assignment process [Moerland *et al.*, 2018; Sequeira *et al.*, 2014]. Lee-Johnson introduced artificial emotional modules into a mobile robot's hybrid architecture[Lee-Johnson and Carnegie, 2010]. The classes of artificial emotions are pre-set and hand-coded as *Fear*, *Anger*, *Happiness* and *Sadness*. Artificial emotions affected an obstacle distance function, which directly impacted the navigation performance. The robot had fewer collisions and more exploration, yet longer path times and slightly slower speed. However, this approach introduced human bias into the artificial emotional modules by hand-coding functions.

Tsankova proposed a robotic navigation architecture with a combination of artificial immune networks and emotion mechanisms [Tsankova, 2009]. *Fear* is introduced by the immune networks into a short-term memory. Experiments showed that the architecture increases the probability of negotiating some closed-loop situations. This approach also requires hand-coding emotion mechanisms.

Williams proposed an emotion inspired adaptive path-planning cognitive architecture without introducing human bias [Williams *et al.*, 2015; Williams, 2016]. The architecture automatically learned mappings between perception states, emotional states, and action states through a reinforcement learning approach. The learned mappings replaced the hand-coded mappings that could introduce human bias. After training, a robot would move faster in a "happy" state, and the robot would have fewer collisions in a "fear" state. However, the number of learned heterogenic emotion states was limited.

3 Preliminaries

3.1 Experimental Scenarios

Experiments were conducted in the Willow Garage Offices (see Figure 1.a), which is a simulation environment sustained by a *Robotic Operation System* backend engine, the *Gazebo* simulator [Williams *et al.*, 2015]. A mobile robot navigates along a targeted path (see Figure 1.a), which is separated into 31 segments. Each segment is contained in a 3m by 3m square area, also termed a *PAOC window*. Although it is not a necessary requirement, each PAOC window is the same size as the local perception map in the *ROS* path-planning module to facilitate the analysis of the results. A PAOC window is the navigation environment of a scenario (see definition of scenario in Section 1). Thus, the segmentations ensured that the 31 scenarios contain various territories (see Figure 1), such as wide-open areas, door-way areas, long narrow corridors and irregular territories.

The targeted mobile robot is *Pioneer* [Williams *et al.*, 2015], which is equipped with a *LIDAR*, front and rear bump sensors, and sonars. Three sources of Gaussian noise are added into the LIDAR, the sonars, and the location perception of the Pioneer to simulate a real-world environment. The common *ROS* navigation stack [Williams, 2016] was applied as the path-planning module. The navigation task of a scenario is to require the path-planning module to generate a path to a goal position. Although the architecture is capable of operating in the physical world, repetition for statistical analysis is too time-consuming, so tasks were conducted in the simulation environment. Because the physical world may contain different noise and uncertainty, robustness is considered in the performance analysis (see Section 5.3).

The path-planning module's admissible behaviours largely rely on hyperparameters, as described in Section 1. These hyperparameters include *inflation radius*, *cost factor*, *publishing frequencies* and so on [Eitan, 2018]. Among these hyperparameters, *inflation radius* has a dominant effect on admissible behaviours of the path-planning method. The *inflation radius* specifies an obstacle-spreading distance from obstacles [Eitan, 2018], and this hyperparameter is critical to the results of the path-planning method: whether the path-planning method can generate a path within the current scenario or not. This is because different values of the inflation radius will result in different perceived environments of the same scenario, especially in narrow spaces (see Figure 1.b, 1.c and 1.d). An overlarge value of inflation radius (e.g. 0.8 metre) will fill space around obstacles with large inflation (see Figure 1.c). Hence, narrow spaces, where the path-planning method should be able to generate a path through, will be considered as being occupied by an obstacle as well. Door-way areas in the scenario No.4 and No.24 (see Figure 7.b and 7.h) are such narrow spaces where the path-planning method could fail to generate a valid path through these areas (see Figure 1.c).

3.2 Knowledge-encapsulated *PAOC*

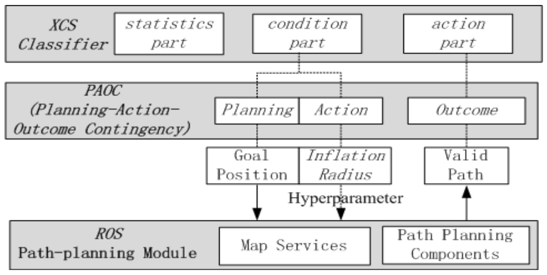


Figure 2: Relationship between *XCS Classifier*, *PAOC*, and *ROS* Navigation Package.

PAOC knowledge seeks to encapsulate the contingences between hyperparameters and scenarios based on its semantic origin (see Section 1). A PAOC is a four-tuple containing different elements of hyperparameters and scenarios (see Figure 2). The first and second attributes of a PAOC are for the *Planning* part, which contains the goal position of the navigation task. Because a goal position is kept in an orthogonal coordinate system (an x-axis-and-y-axis plane), the two attributes are needed. The third attribute is for the *Action* part, which is assigned to a hyperparameter, inflation radius. This hyperparameter is scaled by a modifier axis. The fourth attribute is for the *Outcome* part, which represents the result of a path-planning method The value of one represents a success of generating a valid path to the goal (the first and second attributes) under the value of chosen inflation radius (the third attribute). Otherwise, the value of zero represents a failure. Therefore, a PAOC represents a specific contingency between a value of inflation radius and a specific scenario. For example, a PAOC of (1, 3, 0.3, 1) means that the path-planning module can successfully generate a path to the goal position of (1 metre, 3 metres) if the inflation radius is 0.3 metre.

In addition, a four-tuple of a PAOC can be visualised in a three-dimensional space, termed a *PAOC space* (see examples in Figure 6 and 7). The axes of the first three attributes (the x-axis, y-axis and modifier axis) construct the *PAOC space*. The fourth attribute of a tuple is indicated by the colour of the points. Blue is for the value of one and red is for the value of zero. Therefore, a PAOC is visualised as a spatial, colourful point in the *PAOC space* (see examples in Figure 6.c and Figure 6.e).

PAOC datasets are needed for a reasoning mechanism to generate a more generalized scenario-and-hyperparameter-encapsulated knowledge than a specific PAOC. For each PAOC window, a collected PAOC dataset contains 800 PAOCs, which can be adjusted to fit the granularity required in the domain. As shown in Figure 6.c and Figure 6.e, PAOC dataset contains inaccurate PAOCs, which is a challenge to *XCS* algorithms as they aim to generate accurate rules.

3.3 *PAOC* Application of XCS Algorithm

The rule-based *XCS* algorithm is applied to learn rules, produce generalized, accuracy knowledge and detect patterns from PAOC datasets. Each *XCS* rule contains a *condition-action* pair, and the rule is termed a *classifier* when complemented by *statistics* regarding its worth to the system. Initially, the *condition* part of a classifier covers the *Planning* and *Action* of PAOC, and the *action* part predicts the *Outcome* of the PAOC (see Figure 2 and Section 3.2). During training, classifiers will receive feedback from the environment. The feedback includes reward, if their *action* parts correctly predict the *Outcome* of a PAOC instance. Otherwise, classifiers will receive a punishment. Classifiers should evolve toward accurate, maximally general rules, which cover as many PAOC instances as possible and also predict the *Outcome* accurately. Therefore, the condition parts of these accurate, maximally general classifiers contain accurate, maximally general PAOCs knowledge, termed *PAOC patterns* or *Inflation Radius (IR) patterns* (see Section 5.3). Because the PAOC patterns can be visualised in the *PAOC space* as cuboids, a qualified PAOC pattern could represent what a robot perceives in the environment, e.g. space occupied by an open path or obstacles (see Section 5.3).

3.4 Main Loop of Standard Approach of *XCS* algorithm

An *XCS* algorithm agent learns classifiers by its interactions with an environment in its iteration loops: perception of a current state from the environment, execution of a chosen action to the environment, and receiving a reward from the environment (see Figure 3). At the beginning of an iteration loop, the *XCS* agent perceives a current state, which is a niche from the environment, as a perception. Then, the *XCS* agent initiates its *matching* method to choose classifiers from the *population* ([*P*]). The niche-coverage of the condition part of the chosen classifiers must be able to cover the perception. If there is no classifier in $[P]$ that meets this requirement, the *covering* method will be activated to generate a perception-covered classifier. The chosen classifiers or the perception-covered classifier form a *match set* ($[M]$). As an iteration loop progresses, the *selection* method chooses an *action* from the available *actions*, which come from the classifiers in $[M]$. Classifiers with the chosen *action* in $[M]$ form an *action set* ($[A]$). After the execution of the chosen *action*, the agent will receive a reward from the environment. Based on the reward, classifiers in $[A]$ are updated and returned to the $[P]$ at an end of the iteration loop. In addition to these methods, a *Genetic Algorithm* (*GA*) method will be activated conditionally within $[A]$ to generate new classifiers.

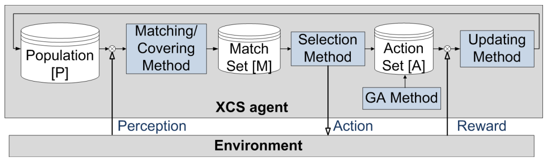


Figure 3: Standard Approach in *XCS* algorithm's Learning Processes.

3.5 Overgeneralized Tendency of *XCS* Algorithm

XCS algorithms are applied as the reasoning mechanism for this study. However, the *standard approach* of an *XCS* suffers from an over-generalization tendency. That is, the condition part of a classifier tends to extend its niche-coverage. This tendency is rooted in Wilson's *generalization hypothesis* [Wilson, 1995]. The *generalization hypothesis* suggests that *XCS* algorithms have a *generalization pressure*, which is an intrinsic tendency to evolve accurate, maximally general classifiers. Butz also confirmed the existence of the generalization pressure on Boolean Multiplexer Problems [Butz *et al.*, 2004]. In addition, Butz noted that the *generalization pressure* originates from interactions between *niched evolution* (i.e. *GA* in $[A]$) and *panmictic deletion* (i.e. removal of excess rules from $[P]$). In *niched evolution*, the classical *GA* of the standard *XCS* creates a much stronger *generalisation pressure* than *specification pressure*. As a result, the over-generalization tendency causes the following problems:

1 An accurate classifier lacks robustness to maintain its accurate state;

2 Classifiers tend to be stuck at a local optimum, where those classifiers stop evolving toward maximal general and accurate classifiers.

The overgeneralized tendency of *XCS* algorithms needs an amendment to improve the achievement of maximal general and accurate classifiers.

4 Method

As described in the section 3.5, the overgeneral pressure tends to produce overgeneral classifiers, which mix accurate niche-coverage with inaccurate niche-coverage. The XCS algorithm lacks the ability to distinguish overgeneral classifiers from other inaccurate classifiers. The standard approach also lacks the ability to distinguish accurate niche-coverage from inaccurate niche-coverage in overgeneral classifiers. As a consequence, qualified classifiers are rare, especially in a noisy environment.

To amend *the overgeneral pressure* that frequently dominates the evolutionary processes of *XCS*, this paper proposes a novel *accuracy pressure*. The *mitosis approach* is inspired by the biological cell's mitosis procedure. In the biological mitosis procedure, an original cell divides into two new cells, and mother chromosomes replicated are separated into two daughter nuclei. Similarly, the proposed artificial mitosis method will generate children classifiers aiming to inherit accurate niche-coverage from a parent classifier but abandon inaccurate niche-coverage. As training progresses, the accurate children classifiers will eventually replace their inaccurate parents in a *classifier population*. As a result, the mitosis approach is anticipated to increase the accuracy of the entire classifier population.

4.1 *Mitosis* Method

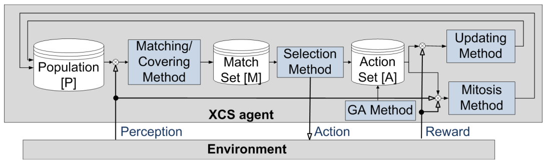


Figure 4: Mitosis Approach in *XCS* algorithm's Learning Processes.

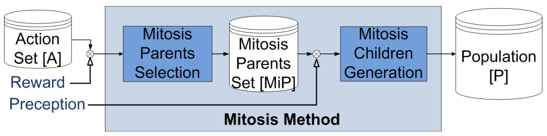

Action Set [A]
Reward
Preception
Mitosis Parents Selection
Mitosis Parents Set [MiP]
Mitosis Children Generation
Population [P]
Mitosis Method

Figure 5: Mitosis' Two-step Procedure.

The mitosis method is added into *XCS* for the mitosis approach (refer Figure 4, and see the standard counterpart in Figure 3). Structurally, the mitosis method is parallel to the updating method, which updates the population according to a reward. Yet the mitosis method needs to be activated before an execution of the updating procedure. This is because the updating method will update and affect accuracy elements of overgeneral classifiers, from which the mitosis method could generate more accurate classifiers.

A classifier's generation process takes two steps in the *mitosis* method (see Figure 5). Firstly, the *mitosis-parents-selection procedure* chooses qualified overgeneral classifiers from $[A]$. All the qualified overgeneral classifiers are denoted as *mitosis parents* and are preserved in *mitosis parent set* ($[MiP]$) waiting for the second step. The second step, *mitosis-children-generation procedure*, will generate new classifiers (denoted as *mitosis children*), and thus insert them into $[P]$. The procedure removes inaccurate niche-coverage from the *mitosis parents* and preserves accurate niche-coverage and other accurate elements for the *mitosis children.* These accurate *mitosis children* in $[P]$ create the accurate pressure to amend the overgeneralized pressure in $[A]$.

4.2 *Mitosis-Parents-Selection Procedure*

algorithm 1 *Mitosis Parents Selection*

```
Inputs: Reward $R$, Action set $[A]$, Mitosis Parents
    Set $[MiP]$, a classifier $cl$, a classifier's accuracy $cl.acc$,
    a classifier's predicted reward $cl.prd$, a preset difference
    threshold of predicted-reward $threshold.prd.diff$.
Outputs: Select qualified mitosis parents.
 1: function SELECT MITOSIS PARENTS($R$, $[A]$)
 2:     $[MiP] \leftarrow empty$
 3:     for $cl$ in $[A]$ do
 4:         if $cl.acc > threshold.acc$ and $cl.prd > threshold.prd$ then
 5:             if $|cl.prd| < |R|$ and $|cl.prd - R| > threshold.prd.diff$ then
 6:                 $[MiP] \leftarrow cl.id$
 7:             end if
 8:         end if
 9:     end for
10:     return $[MiP]$
11: end function
```

The *mitosis-parents-selection procedure* identifies mitosis parents because they represent the overgeneralized pressure around optimum solutions. The mitosis parents must meet three requirements (see Algorithm 1). Firstly, mitosis parents are classifiers that make the current incorrect prediction in the current iteration. This will directly separate inaccurate classifiers from accurate classifiers. Secondly, mitosis parents must have achieved the maximum accuracy (*threshold.acc*) before this iteration. This requirement will filter overgeneral classifiers from the rest of the incorrect classifiers. Thirdly, mitosis parents have to reach the value of the maximum reward (*threshold.prd*). This requirement attempts to identify qualified overgeneral classifiers that are around optimum solutions. If a classifier's absolute value of a predicted reward (*cl.prd*) has achieved the value of the maximum reward, this classifier is more likely to reach the global optimum solution than those who have not. If $[MiP]$ is empty, the next procedure, the *mitosis-children-generation procedure*, will not be activated. Therefore, the *mitosis-parents-selection procedure* potentially increases the efficiency of the mitosis method by focusing on qualified overgeneral classifiers. In addition, without this procedure to filter out unqualified classifiers, the mitosis method risks the introduction of an undesirable over-specified pressure into $[P]$.

4.3 *Mitosis-Children-Generation Procedure*

algorithm 2 *Mitosis Devision*

```
Inputs: An Individual Mitosis Parent $cl$, Stituation $\sigma$,
    New Generation's Gene Set $[Gene]$
Outputs: Target: Give birth to mitosis children
    through mitosis process
1: function MITOSIS EXECUTION PROCESS($cl$, $\sigma$ )
2:     $[Gene] \leftarrow$ CHROMOSOME MORPHING($cl$, $\sigma$)
3:     $[NewBorn] \leftarrow$ TELOPHASE($cl$, $[Gene]$ )
4:     return $[NewBorn]$
5: end function
```

The *mitosis-children-generation procedure* passes accurate elements from mitosis parent to children classifiers. If there are qualified mitosis parents in $[MiP]$, the *mitosis-children-generation procedure* will activate a mitosis-children-generation loop. In this loop, each mitosis parent in $[MiP]$ generates its own children classifiers by a procedure, termed the *mitosis-division procedure* This procedure contains two sub-procedures: *chromosome-morphing procedure* and *telophase procedure* (see Algorithm 2).

The *chromosome-morphing procedure* generates an accurate condition part for children classifiers by detecting (and thus removing) inaccurate niche-coverage within the mitosis parent. The detecting is based on overlaps between the current *perception* and the niche-coverage of the condition part of mitosis parent. Because the mitosis parent is inaccurate under the current *perception*, these overlaps must contain an inaccurate niche-coverage. Thus, condition parts that have removed one of the overlaps have a more accurate niche-coverage for the mitosis child.

The *telophase procedure* combines every part of a classifier into the new *mitosis child.* The condition part has been generated by the *chromosome-morphing procedure.* The action part directly inherits from the mitosis parent. The *statistics* of children classifiers also inherit from their parent, except *numerosity* and *experience*. The *numerosity* specifies the number of copies of a classifier. The *experience* tells how many times a classifier has been activated. In this procedure, these two statistics are reset to their initial values of one.

5 Result and Discussion

5.1 Result

The experiments show that the emotion inspired cognitive architecture was able to increase navigation adaptation of the mobile robot. The cognitive architecture of learned *PAOC patterns* was able to generate valid paths in all 31 navigation scenarios (see Figure 1.d and Section 5.2). *PAOC patterns*, the knowledge-encapsulated cognitive architecture, were visualised for interpretation (see Section 5.3). Experiments also showed that the emotion inspired reasoning mechanism applied by *XCS* algorithms was capable of learning PAOC knowledge in the 31 navigation scenarios. The *mitosis approach* of *XCS* algorithm performed better than the *standard approach* on almost all of the 31 navigation scenarios in terms of *prediction accuracy* (see Section 5.2), *pattern robustness* (see Section 5.3), and *pattern accuracy* (see Section 5.3).

5.2 Quantitative Analysis

Quantitative analysis is based on the *prediction accuracy* of learned PAOCs. The *prediction accuracy* measures the ratio of the number of correctly predicted PAOC instances to the number of all instances in a PAOC dataset. The standard XCS approach provides the benchmark for the mitosis approach proposed. A *one-hidden-layer neural network* is included to demonstrate potential interferences from the PAOC datasets, because these datasets of various scenarios contain different numbers of inaccurate instances.

The quantitative analysis of *prediction accuracy* highlighted that the mitosis approach achieved better prediction accuracy than the standard approach (see Table 1). The mitosis approach had statistically significant improved prediction accuracies in 23 of the 31 navigation scenarios based on two-tailed T-test with 0.05 P-Value (see Table 1, each navigation scenario run 30 trails). In the remaining eight navigation scenarios, *prediction accuracies* were not significantly different between the two approaches, yet the mitosis approach achieved a higher value of the mean of *prediction accuracy* in seven of these eight scenarios and a slightly worse performance in one scenario. As the mitosis approach had overall better *two-tailed performances*, this also indicated that the accuracy pressure that was introduced by this method could effectively amend the overgeneralized tendency of the standard *XCS* algorithm.

5.3 *Inflation Radius Pattern* Analysis

A major advantage of the two *XCS* approaches over the applied neural network is that the learned knowledge of the *XCS* approaches is easy to interpret. The neural network learns knowledge through matrix modules, in which mappings and weights usually require extra efforts to interpret. In contrast, *XCS* preserves knowledge in rules, where the "if-then" structure is comprehensible. In addition, because PAOC knowledge is directly encoded into *XCS* rules (see Section 3.3 and Figure 2), learned, qualified PAOC knowledge, termed *PAOC patterns*, can be conveniently harvested from a qualified classifier for visualisation.

PAOC patterns were visualised as cuboids in the three-dimensional *PAOC Space*(see Section 3.2) for interpretation (see Figure 6 and 7). A shape of a cast from a pattern to the x-axis-and-y-axis plane covers the targeted area within the navigation environment that these patterns can apply. The length of cast from a pattern to the modifier axis indicated an inflation radius value, which is what the path-planning module applies to the targeted area. Therefore, patterns provide more generalized knowledge than specific PAOCs. Besides, the PAOC patterns also are also called *Inflation Radius (IR) patterns*, because these patterns describe the learned inflation radius distribution on the x-axis-y-axis plan. Ideally, four types of IR patterns could be generated by *XCS* algorithm (see Table 2). Specifically, *TP* and *TN pattern* advocate values of inflation radius for open space, and *FP* and *FN pattern* work for space that contains

Table 1: Statistics of The Two Approaches and The Standard One-hidden-layer Neural Network

Scenario ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mean 1	0.836	0.909	0.902	0.823	0.818	0.875	0.777	0.798	0.826	0.814	0.847	0.805	0.812	0.894	0.940	0.886
Mean 2	0.826	0.901	0.892	0.814	0.808	0.866	0.760	0.779	0.813	0.816	0.834	0.770	0.791	0.860	0.940	0.868
Mean 3	0.800	0.851	0.840	0.732	0.721	0.787	0.644	0.706	0.641	0.630	0.614	0.688	0.611	0.858	0.836	0.807
P value	0.005	0.009	0.004	0.117	0.159	0.022	0.002	0.000	0.005	0.704	0.002	0.000	0.006	0.000	0.786	0.000

Scenario ID	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Mean 1	0.909	0.926	0.923	0.925	0.886	0.918	0.911	0.836	0.805	0.819	0.837	0.800	0.772	0.805	0.857
Mean 2	0.904	0.918	0.908	0.914	0.882	0.912	0.900	0.818	0.785	0.805	0.824	0.782	0.762	0.797	0.838
Mean 3	0.897	0.904	0.922	0.914	0.841	0.877	0.827	0.747	0.633	0.764	0.703	0.681	0.766	0.690	0.796
P Value	0.090	0.009	0.000	0.000	0.071	0.005	0.019	0.014	0.026	0.003	0.014	0.001	0.070	0.154	0.002

[1] Mean 1: mean value of prediction accuracies of the *mitosis approach* for 30 trials.
[2] Mean 2: mean value of prediction accuracies of the *standard approach* for 30 trials.
[3] Mean 3: mean value of prediction accuracies of the standard one-hidden-layer neural network for 30 trials.
[4] P value: P-value of prediction accuracies the *mitosis* and *standard approach* by the two-tailed T-test among 30 trials.

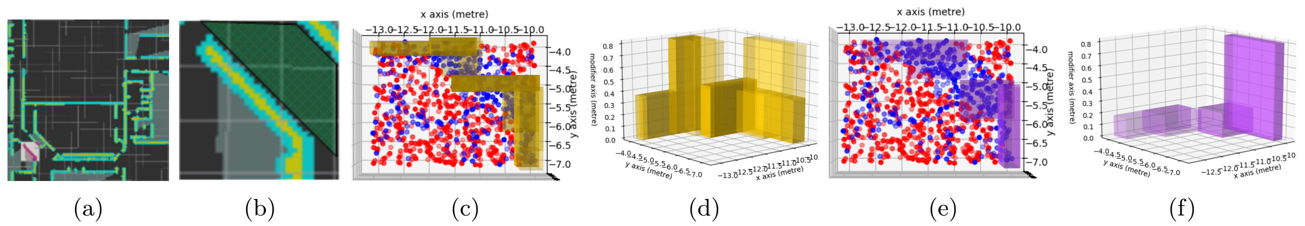


Figure 6: Two Types of Discovered IR Patterns - Scenario No.7, Mitosis Approach
Global Perception Map: a; Local Zoom Map: b (targeted area shown with the green grid); *TP pattern*: yellow cuboids, *TN pattern*: purple cuboids; plan view: c, e; diagonal view: d, f.

obstacles.

Table 2: Confusion Matrix of *IR Patterns*

		True Occupancy	
		Open Spaces (True)	Obstacles (False)
Classifier Prediction	Open Spaces (Positive)	TP Pattern (True Positive)	FP Pattern (False Positive)
	Obstacles (Negative)	FN Pattern (False Negative)	TN Pattern (True Negative)

Experiments show that the proposed cognitive architecture was capable of being applied to various territories. The 31 navigation scenarios contained various territories, such as wide-open areas, long narrow corridors, door-way areas and irregular territories. Four navigation scenarios are selected as examples for three typical territories in Figure 6 and Figure 7. Navigation scenario No.7 was for a long corridor with 45-degree orientation (see Figure 6.a, and green grid in Figure 6.b.). Scenario No.28 was for an irregular territory (see Figure 7.b). Scenario No.4 (see Figure 7.h) and No.24 (see Figure 7.n) were for door-way areas.

Experiments show that the mitosis approach performed better than the standard approach to learn valid IR patterns in terms of *pattern robustness* and *pattern accuracy*. Firstly, patterns were more frequently achieved by the mitosis approach than the standard approach. The *pattern robustness* refers to the frequency of valid IR patterns which can be harvested at the end of a learning process. An example is their performance comparison in navigation scenario No.7 (see Figure 6). Patterns aim to cover the corridor without a gap. The 45-degree orientation of the corridor creates a challenge for two approaches to generate accurate patterns, which will have an accurate coverage. The coverage requirement is similar to filling a trapezoidal space with cubes that can be skewed to align the axes. Thus, gaps between the cubes are difficult to avoid, and connected patterns without a gap are more accurate than patterns with gaps. Based on the performances of *TP pattern* in the total 30 trials repetition, the mitosis approach overcame this challenge 15 times, and the standard approach made it 7 times.

Secondly, patterns that were achieved by the mitosis approach are more accurate than that of the standard approach. *Pattern accuracy* can be judged by how accurately a pattern covers the targeted area in the perception map, hence a global optimum solution is patterns that can accurately cover the targeted area. The mitosis approach was able to achieve a global optimum solution in complex territories, e.g. Scenario No.4, No.24 and No.28, where the standard approach failed (see discussions in the next two paragraphs). This suggests that the mitosis approach overcame two problems caused by the overgeneralization tendency of the standard *XCS* algorithm (see Section 3.5):

1 Mitosis approach provides an accuracy pressure that keeps accurate classifiers in their accurate states;

2 This accuracy pressure drives the evolutionary methods toward maximally general and accurate classifiers.

The first point was shown in irregular territories, such as Scenario No.28. In Scenario No.28, TN patterns of the mitosis approach reached the global optimum solution because of their accurate, full coverage (see Figure 7.c). In contrast, TN patterns of the standard approach only reached local optimum solutions because of their partial coverage (see TN patterns learned by the standard approach in three different trials in Figure 7.d, 7.e, and 7.f respectively). It is worth noticing that the combination of three local optimum solutions of the standard approach could provide the global optimum solution as in the mitosis approach. This also suggests that the standard approach has the potential to generate all the accurate rules that are necessary to represent a globally optimum solution. But, the standard approach failed to maintain some of these accurate rules in the later learning processes (see Section 3.4), hence this approach only achieved the local optimum solutions in this scenario. In contrast, the mitosis approach was able to maintain accurate classifiers because of the introduction of the accuracy pressure.

The second point was highlighted in doorway areas, such as Scenario No.4 and No.24. Again, the mitosis approach reached the global optimum solution and the standard approach failed (see (i-l and o-r) in Figure 7). Specifically, TN patterns of the mitosis approach con-

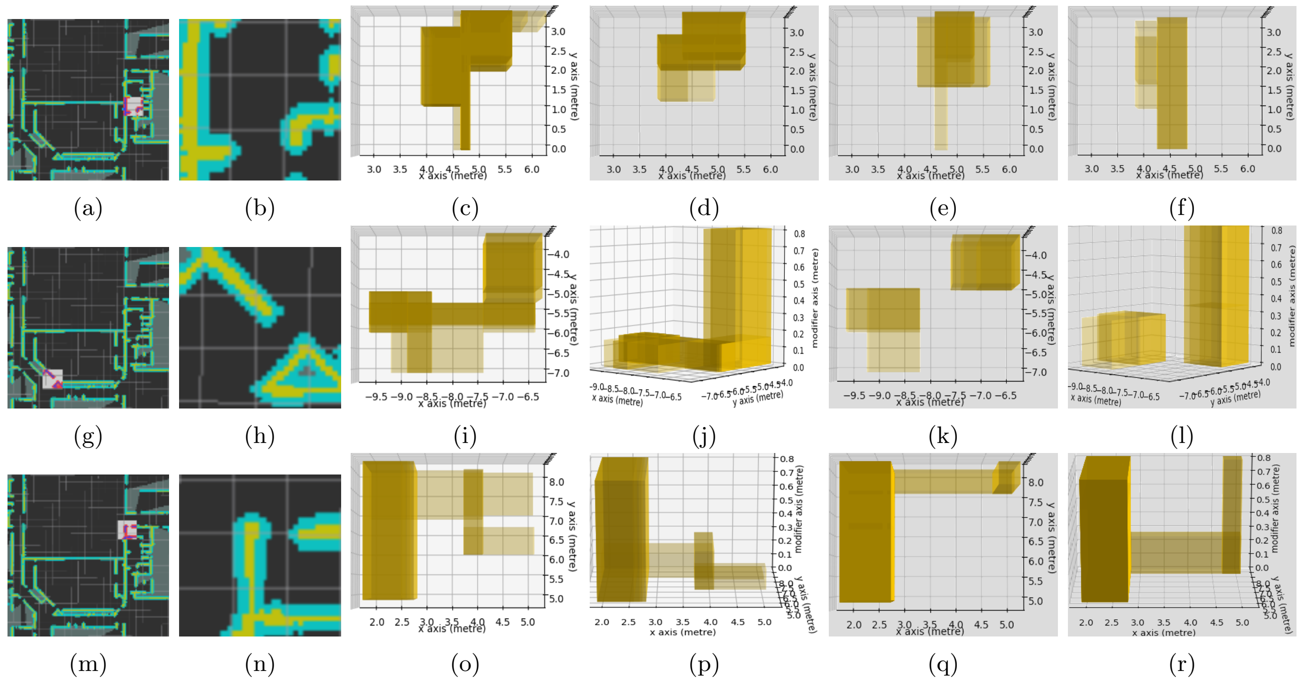


Figure 7: Discovered IR Patterns - Scenarios No.28, No.4 and No.24 per row respectively.
Note: unresolved gaps in paths learned by the standard approach compared with the novel approach, e.g. k vs i.
Mitosis: white background; Standard: grey background. Row 1 shows single optimal path vs three local paths.
Global Perception Map: Col. 1; Local Zoom Map: Col. 2; All columns plan views, except diagonal view: j, l, p, r.

tained the narrow doorways that the standard approach was less likely to identify them (see comparisons between (i) and (k), (j) and (l), (o) and (q), and (p) and (r) in Figure 7). These suggest that the generalized pressure in the standard approach is insufficient to learn accurate classifiers from such territories. These also suggest that the accurate pressure introduced by the mitosis approach can drive classifiers evolving toward maximally general and accurate classifiers.

Patterns also highlighted a complementary relationship. TP and TN patterns approach to the same ground truth pattern in open scenarios in opposite directions. A TP pattern approaches the ground truth pattern from its “upper boundary”, because it tends to include neighbour spaces, which belong to boundaries of two types of spaces (see Figure 6.c and 6.d). Therefore, a TP pattern can be considered as a maximally general, accurate pattern. In contrast, a TN pattern becomes a “lower boundary” of the ground truth pattern, because it tends to exclude such neighbour spaces (see Figure 6.e and 6.f). Therefore, a TN pattern can be considered as a minimally specific, accurate pattern. Both of a maximally general, accurate pattern and a minimally specific, accurate pattern can be represented as the robot’s perceptions of the environment.

However, both the mitosis approach and the standard approach did not generate robust FP and FN patterns, which predict space that is occupied by obstacles. FP and FN patterns were relatively rarely generated in all the 31 navigation scenarios. This is because of inaccurate instances within scenarios. In these scenarios, the obstacle-occupied space contains a high proportion of inaccurate instances that advocate the open space. Due to the lack of a sufficient amount of the accurate instances, FP and FN patterns are much less frequently generated compared with TP and TN patterns.

A success real-world application of this work depends on the noise and uncertainty in the world. If the noise has the same or less magnitude and the similar distribution as in the simulations, the system can function well. If the noise has larger magnitude or different distributions, then as long as the number of inaccurate instances that are caused by the noise is under a known accuracy limit, the system can still function because of PAOC pattern robustness. Once the known accuracy limit is exceeded, further experiments will be needed to investigate the effects of the noise.

6 Conclusion

This paper proposed an emotion inspired cognitive architecture for robotic adaptive path-planning without hand-coded prior knowledge. Firstly, *PAOC* knowledge is the basis of the proposed cognitive architecture, which is beneficial to navigation adaptation. Experiments showed that the architecture was able to generate valid paths in all the 31 navigation scenarios. Secondly, the PAOC patterns were automatically learned by the novel algorithm, the *mitosis approach*, instead of hand-coded hyperparameters for targeted scenarios as prior knowledge. Learned *PAOC patterns* (*IR patterns*) are plain and interpretable for humans. The visualisations of *TP pattern* and *FP pattern* showed that these patterns accurately describe open space within the navigation scenarios. Finally, our mitosis approach performed better than the standard approach in terms of *prediction accuracy*, *pattern robustness* and *pattern accuracy* as the novel accuracy pressure amended the overgeneralized tendency of the standard *XCS* algorithm.

References

[Butz *et al.*, 2004] Martin V Butz, Tim Kovacs, Pier Luca Lanzi, and Stewart W Wilson. Toward a theory of generalization and learning in XCS. *IEEE Trans. Evolutionary Computation*, 8(1):28–46, 2004.

[Eitan, 2018] Dave Hershberger Eitan, David V Lu. Wiki ros: Costmap2d package summary. `http://wiki.ros.org/costmap_2d`, 2018.

[Elbanhawi *et al.*, 2013] Mohamed Elbanhawi, Milan Simic, and Reza Jazar. Autonomous robots path planning. In *Applied Mechanics and Materials*, volume 373, pages 246–254. Trans Tech Publ, 2013.

[Lee-Johnson and Carnegie, 2010] Christopher P Lee-Johnson and Dale A Carnegie. Mobile robot navigation modulated by artificial emotions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(2):469–480, 2010.

[Mac *et al.*, 2016] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, and Robin De Keyser. Heuristic approaches in robot path planning. *Robotics and Autonomous Systems*, 86:13 – 28, 2016.

[Moerland *et al.*, 2018] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Emotion in reinforcement learning agents and robots: a survey. *Machine Learning*, 107(2):443–480, 2018.

[Rolls, 2013] Edmund T Rolls. *Emotion and decision making explained.* Oxford University Press, 2013.

[Scherer *et al.*, 2010] Klaus R Scherer, Tanja Bänziger, and Etienne Roesch. *A Blueprint for Affective Computing: A sourcebook and manual.* OUP, 2010.

[Schlesinger and McMurray, 2012] Matthew Schlesinger and Bob McMurray. The past, present, and future of computational models of cognitive development. *Cognitive Development*, 27(4):326–348, 2012.

[Sequeira *et al.*, 2014] Pedro Sequeira, Francisco S Melo, and Ana Paiva. Learning by appraising: an emotion-based approach to intrinsic reward design. *Adaptive Behavior*, 22(5):330–349, 2014.

[Tsankova, 2009] Diana D Tsankova. Emotional intervention on an action selection mechanism based on artificial immune networks for navigation of autonomous agents. *Adaptive Behavior*, 17(2):135–152, 2009.

[Urbanowicz and Browne, 2017] Ryan J Urbanowicz and Will N Browne. *Introduction to learning classifier systems.* Springer, 2017.

[Urbanowicz and Moore, 2009] Ryan J Urbanowicz and Jason H Moore. Learning classifier systems: a complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*, 2009:1, 2009.

[Williams *et al.*, 2015] Henry Williams, Christopher Lee-Johnson, Will N Browne, and Dale A Carnegie. Emotion inspired adaptive robotic path planning. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 3004–3011. IEEE, 2015.

[Williams, 2016] Henry Williams. *Human inspired robotic path planning and heterogeneous robotic mapping.* PhD dissertation, Victoria University of Wellington, 2016.

[Wilson, 1995] Stewart W Wilson. Classifier fitness based on accuracy. *Evolutionary computation*, 3(2):149–175, 1995.