**A COMPREHENSIVE REVIEW ON ARTIFICIAL INTELLIGENCE BASED**

**MACHINE LEARNING TECHNIQUES FOR DESIGNING INTERACTIVE CHARACTERS**

* Paper reviews current literature based on the challenges in real time strategy game and explores the tasks in a real-time environment.
* To achieve optimal performance many techniques have been used along with the neural networks, which are prepared end to end in tasks such as object recognition video games and board games.
* The comparative analysis is made on different AI based machine learning techniques to comprehend the performance issues and other challenges faced while trying to obtain optimal performance.
* Paper contains discussion on the advantages and disadvantages of each technique to understand and compare the efficiencies with the existing techniques based on the strategies.
* Techniques used for review of the artificial intelligence-based machine learning techniques for designing interactive characters naive Bayes classifier, support vector machines, genetic algorithms and neural networks.
* The review techniques were used to create a new intelligent, serious game and to provide the players with a real experience.
* These techniques were compared and it was observed that each technique has its own pros and cons whereas neural networks are better compared to other techniques and thus help in designing of the player characters in video games

[P.S – waste paper, has grammatical errors in it -\_-]

**EFFICIENT REINFORCEMENT LEARNING IN ADVERSARIAL GAMES**

* Algorithm: MiniMax
* Paper presents a learning approach based on the Least-Squares Policy Iteration (LSPI) algorithm that overcomes MiniMax optimality criterion or under optimization limitations by focusing on learning a state-action evaluation function.
* The approach to learning in a variety of two-player, turn-taking, tactical adversarial games (Eg: Backgammon, Othello/Reversi, Chess, Hex, etc) consists of updating some state evaluation function usually in a Temporal Difference (TD) sense either under the MiniMax optimality criterion or under optimization against a specific opponent.
* Limitations of MinMax:
  + updates to the evaluation function are incremental
  + stored samples from past games cannot be utilized
  + quality of each update depends on the current evaluation function due to bootstrapping.
* Advantage of LSPI : agent can make batch updates to the evaluation function with any collection of samples, can utilize samples from past games, and can make updates that do not depend on the current evaluation function since there is no bootstrapping.
* Things from paper that we can use:
  + For any given state [board, player, parity] of the game, we compute the following:
    - mobility: Number of available moves for the player in the current board.
    - stability: Number of player’s discs in the current board whose color cannot change in the rest of the game.
    - frontier: Number of player’s discs in the current board adjacent to empty squares.
    - square(i, j): Content (player disc, opponent disc, or empty) of square (i, j) in the current board.
* Experimental results and graphs for Mukesh

**EVALUATING A REINFORCEMENT LEARNING ALGORITHM WITH A GENERAL INTELLIGENCE TEST**

* Paper applies recent notion of anytime universal intelligence tests to the evaluation of a popular reinforcement learning algorithm, Q-learning.
* Show that a general approach to intelligence evaluation of AI algorithms is feasible.
* Simple terms: Evaluates progress of AI, that is, does performance and intelligence test.
* Evaluation is better when the problem or task is understood completely (more time understanding the nuts and bolts)
* Paper uses “notion of universal distribution and the related algorithmic information theory” to “define a universal distribution of tasks for a given AI realm, and sort them according to their (objective) complexity.”

(Personally, I do not know what this means)

* The experiment run in this paper portrays a view about how an implementation of the intelligence test using the environment class can be used to evaluate AI systems.
* The goal of the paper was not to analyse some well-known properties of Q- learning (such as convergence, state overloading, etc.) or to designate a `winning' algorithm. The goal of the paper, rather, was to show that a top-down (theory-derived) approach for evaluating AI agents can work in practice.

**GENERAL VIDEO GAME EVALUATION USING RELATIVE ALGORITHM PERFORMANCE PROFILES**

* Paper explores the idea of characterising game quality through playing a game with different controllers and comparing performance.
* It does so by investigating the relative performance of different general game-playing algorithms.
* Seven game-playing algorithms was used to play several hand-designed, mutated and randomly generated VGDL game descriptions.
* The seven controllers were used to play through a set of example-, mutated and randomly generated games.
* Result show that win rates suggest a relationship between intelligent controllers' success and better game design; for better designed games, the relative performance of different types of algorithms differ more.
* Average wins for generated, mutated and example games are graphically represented
* Paper hypothesised that the performance difference between good and bad game-playing algorithms is higher on well-designed games, and therefore can be used as at least a partial proxy for game quality.
* The results corroborate the paper’s initial conjecture, showing a clear distinction between results of more and less intelligent controllers for human-designed games but not for random games.
* Images from game for explanation

**REAL-TIME CONNECT 4 GAME USING ARTIFICIAL INTELLIGENCE\***

* Problem statement: The study presented a design that converted connect 4 game into a real-time game by incorporating time restraints.
* Approach: The design used Artificial Intelligence (AI) in implementing the connect 4 game. The AI for this game was based on influence mapping.
* Results: A waterfall-based AI software was developed for a Connect 4 game.
* Conclusion: A real time connect 4 game was successfully designed and implanted with GUI using C++ programming language.
* The concept of the Connect 4 game is to get, before your opponent, four chips in a row, arranged either diagonally, vertically, or horizontally.
* Algorithms considered are minimax, minimax with alpha-beta pruning, A\* and influence maps.
* Time was used to check performance
* Multiplayer
* Explanation and flowchart

**HEURISTIC SEARCH WHEN TIME MATTERS**

* Paper propose an approach – Bugsy. It incorporates the utility function directly into the search, obviating the need for a separate termination policy.
* It is based on off-line parameter tuning and a novel benchmark domain for planning under time pressure based on platform-style video games.
* Bugsy does not require any off-line training.
* First paper to apply anytime monitoring to anytime heuristic search.
* Paper presents a very simple portfolio-based method that estimates a good parameter to use for a bounded-suboptimal search algorithm to optimize a given utility function.
* Bugsy is a best-first search algorithm that accounts for the user’s preference between search time and solution cost.
* Difference between Bugsy and most other methods for trading-off deliberation time and solution cost is that Bugsy considers the trade-off directly in the search algorithm, whereas previous techniques, such as those based on anytime algorithms, only consider the trade-off externally to the actual search algorithm.
* Background research on heuristic and suboptimal searches and utility functions are done for the paper
* Paper is huge, has 4 pages worth of references.