Turn Based War Chess Model

Introduction :

A turn-based strategy game (TBS) is a strategy game where players take turns when playing. This is distinguished from real-time strategy, in which all players play simultaneously. TBS games have become the second most famous type of game after RPGs (role-playing games). TBS games will have greater potential for development in the areas of touch-screen operation , lightweight , fragmented time and so on. The content of a TBS game generally comprise of two levels :

Strategic coor-dination
Tactical battle control

The latter level, whose rules are similar to those of board games, for example, moving pieces on the
board, beating a specified enemy target for victory, and turn-based orders, is called the turn-based war chess game (TBW). There is traditional and classic games like Go , chess, backagammon etc . Which require normal alpha-beta search algorithms and MCTS algorithm wherease advance AI is required in TBW. AI of TBW is an important component of all TBS games. So improving and adding new techniques to AI of TBW is the need and neccesity of TBS games. A precise attention is paid to TBW's unit moves, attacks and presentation of war ground . The tactical module has to decide where to move the units and plan the attacks. Reinforcement learning is used to learn rule weights (that influence decision making), while evolutionary learning is used to evolve good rule sets. It increases the flexibilty of a game. TBW's have many

similiarities with structural chess moves and tactics . A TBW game is essentially the compound of combinational optimization laterally and game tree search vertically, which can be regarded as a programming problem of multiagent collaboration in stages and can be seen as a tree search problem with huge branching factor. Two types of search algorithms for a single round from different certain angles could be put :
Dictionary sequence enumeration method and recursive enumeration method.

Some famous TBS games are :
PANZER CORPS 2

UNITY OF COMMAND 2

Game Module :

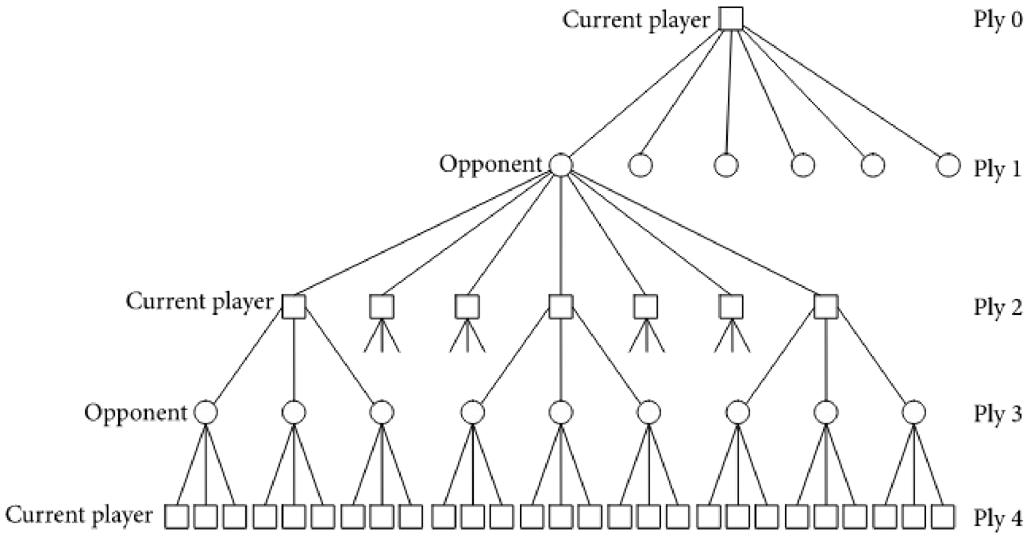
TBW is played on a square, hexagonal, or octagonal tile-based map. Each tile is a composite that can consist of terrain types such as rivers, forests, and mountains or built up areas such as bridges, castles, and villages . Each tile has a movement cost for the unit on that tile.The movement of the unit and movement cost of the tile depends upon the type of terrain for.e.g. if the terrain has mountians or rivers then movement cost of the tile is undefined which means a unit or a player could not cross that tile. Each tile is occupied by only one unit at the same time. Each player in TBW controls a number of units which in some games is its army. Each unit has different movement , attack ranges and power. All of the tiles the unit can travel to compose a union of them called movement range , including the tile occupied by the unit itself. The movement range can generally be calculated by some algorithm such as breadth first search. If it has free pass or same army people , he can move and if he finds an enemy in the path , it could either attack or change path as per the strategy. In addition to the movement point, each unit has its own health point (Hp) and attack power (ATK), which are numerical values and are various among the different units. Like movement range, a unit's attack range is another union of tiles to which the unit can attack from its current tile. A unit's attack range can be identified by the attack technique. For e.g. cavarly or swordsmen could only attack adjacent players, archers could attack for a given range and could also attack across rivers and terrains , cannonballs even farther attack and bigger damage etc. If a unit attacks another unit, it forfeits all of its movement points and cannot take any further actions that turn; therefore, if a unit needs to be moved to a different tile, it must perform the move action prior to performing an attack action. A unit also has the option not to take any attack action after its movement or even not to take any action and stay on its current tile. Each unit attacked by its enemy must deduct its Hp by the attacking unit’s ATK, which indicates the damage. When a unit’s Hp is deducted to or below 0, this indicates that it is dead and must be removed from the board immediately. The tilt it occupied becomes empty and can be reached by other following units. A game of TBW is the sequence of turns for each of the player moves and giving the correct sequence of attack and movement. TBW is composed of the board and pieces (units). The board is considered as an undirected graph G(V, E), where V is the set of vertices (tilts) and E is the set of edges that connect the neighboring tilts. A game tree search theory could be used to reserach the AI of TBW. In game theory, a game tree is a directed graph whose nodes are positions in a game and whose edges are moves. In the game tree , nodes express states of the game board. Branches derived from nodes express selections of the move method. The root node is the current state, and the leaf nodes are end states whose depths are specifically expanded from the root. Both sides take turns. The most common game tree search are Alpha-Beta search , Monte Carlo Tree Search etc. But they cant be applied directly to the TBW’s. TBW is huge and common search algorithm would generate timeout.

A very common Game Module :

Complexity Analysis :

A game of TBW consists of a sequence of turns. During each turn, every player gets their own turn to perform all of the actions for each of their side’s units, which is the most important feature of TBW. Compared with TBW games, other board games (such as chess, checkers, etc.) only require

selecting a unit to perform an action in a single round, which not only results in fewer single-round action plans but also makes the number of plans linear with increasing numbers of units. In TBW , the branching is extensive and increasing explosively. A TBW game is essentially the compound of combinational optimization laterally and game tree search vertically. It can also be seen as a tree search problem with huge branching.



Single Round Search Algorithm

a) Dictionary Sequence Enumeration Algorithm

Determining an action sequence of n A units requires a permutation algorithm. There are some famous permutation algorithms, such as the recursive method based on exchange, the orthoposition trade method, the descending carry method, and the dictionary sequence method. Enumerate all of the plans of units' actions in a particular order. Because the search depth is limited (equal to the number of units), depth-first search is an effective method. Because the depth is not great, realizing the depth-first search by the use of recursion requires smaller space overhead, which leads to the sequential enumeration algorithm withpermutation and recursion.

b) Recursive Enumeration Algorithm

On the basis of this feature, we switch to the recursive permutation algorithm to achieve the arrangement so that the recursive algorithm combines with the recursive depth-first search algorithm for the purpose of removing the redundant computation, which is the improved algorithm called the recursive enumeration algorithm. n is the size of our sequence. With respect to the predefined procedure, we generate the permutations from the ith to the last unit in the sequence by calling the function recursive permutation(i). The latter is realized using the subpermutations from the i + 1th to the last unit in the sequence, which are generated by calling the function recursive permutation(i + 1) recursively. The index j points to the unit swapped with the ith unit in every recursive call, after which the two units must resume their orders , for the next step. By initializing the sequence Ord and running the function recursive permutation(1), we can obtain a full permutation of all the elements.

Now lets compare both the algotithms :

The time consumption of the recursive enumeration algorithm lies in an n times loop and an $n - 1$ times recursion, such that the time complexity is $O(n(n-1)(n-2) \cdot \cdot \cdot 1) = O(n!)$. It is the same as the time complexity of the dictionary sequence enumeration algorithm. Moreover, the states searched by the two algorithms are also the same.

On the premise that the search states are exactly the same, Recursice Enumeration Algorithm is better than Dictionary Sequence Enumeration Algorithm regarding the consumption of ops1 and actual running time.

Conclusion :

Turn-Based War Chess Game (TBW) is the most important part of turn-based strategy game, and the research on its AI is hard but has a great significant impact not only on video games but also on applied mathematics.

In the current era of digital entertainment, TBW games have broad application prospects. They also have a profound theoretical research value.