Parallel Computing

2022/2023 1st semester

Exercise Sheet #9 2022-12-07

1. The program `nim.py` simulates a simplified version of the *NIM* game. In this version, there are n pieces in only one row at the beginning. The two players move alternatingly, and during each move have to take either 1, 2 or 3 pieces away. The winner is the one who manages to take the last piece.

   Try out the program and analyse how it works.

   Supply a final evaluation at the end of the program that, knowing the correct strategy of the game, verifies until which position the automatic algorithm already managed to learn the correct move to take at this position in order to win.

   Detect this limit for the cases of simulating 10, 100, 1000, 10000 games with 1000 inicial pieces.

2. Whenever the choice of the best move is ambiguous at a certain position, i.e. there is more than one move with the same highest score, the program simply uses the first option. This creates a certain bias to the choice of the moves to be taken, especially at the beginning when all scores are still at 0.

   Implement a mechanism that, for these situations, uses a random choice between the possible moves with the same highest scrore instead.

   Does this change the efficiency of the learning process?

3. Use the same mechanism of machine learning in order to analyze the classical *NIM* game. This game also has two players, but starts with four rows containing 1, 3, 5 and 7 pieces, respectively. In each move, the player has to take away at least one piece and can take up to all pieces of one single row. The winner is the one who manages to take the last piece.

   Use the program to discover the winning strategy for this version of the game, or rather to conclude that there is none.

4. Discuss if this algorithm can be parallelized and how, or why not (submit a text file in response).

Submit the codes of your programs (1 to 4) at `https://trixi.coimbra.lip.pt/cap` until the beginning of the next lectures.