Alexandru Oprisan

Analysis of Heuristic Scoring Functions For Isolation Game

As part of the Isolation Game Project, I implemented the following heuristic scoring functions:

1. Student: number_of_my_moves – 0.7x number_of_opponent_moves
2. Adversarial Optimized: number_of_my_moves – 0.7x number_of_opponent_moves but alternative implementation
3. Cooperative: number_of_my_moves + 0.5x number_of_opponent_moves. This heuristic tries to cooperate wit the opponent (Doesn't really make sense)
4. Center: number_of_my_moves + try to stay close to the center of the board
5. Margins: number_of_my_moves + try to stay close to the margins of the board
6. adv_center: number_of_my_moves – 0.7x number_of_opponent_moves + stay close to center

Performance:

Heuristic \ wins against	Random	MM Null	MM Open	MM Improved	AB Null	AB Open	AB Improved	Percentage of wins
Student	40	34	28	25	34	25	24	75.00
adv optimized	37	33	26	26	29	27	19	70.36
cooperative	38	32	26	20	31	20	22	67.50
center	38	34	27	24	29	20	12	65.71
margins	37	36	22	23	29	21	16	65.71
ID_improved	38	36	29	26	33	24	20	73.57
adv_center	38	32	29	20	27	25	18	67.50

Conclusions:

- As expected, "cooperative" does not make an improvement in adversarial search.
- Trying to bias the heuristic to staying close to the center or close to the margins does not help
- using a weight different then 1 for number_of_opponent moves can provide small improvements over ID_improved.
- I have observed significant variation in results between executions of the tournament script (for ex 5% for ID_improved). This makes me think it should be possible to improve the underlying alpha-beta algorithm (common to all heuristics) to better handle some specific situations.
- Student, Cooperative and ID improved are the only heuristics to defeat each opponent in the evaluation.

Recommendation:

I chose **Student=number_of_my_moves – 0.7x number_of_opponent_moves** as the final implementation because:

1. it is comparable to ID_Improve in performance (slightly better overall win rate)
2. consistent against every direct opponent during the tournament. Won more then 50% of the matches against every opponent
3. It takes into account both players, not just one
4. it is as intensive to compute as the Improved heuristic provided in sample_players.py