Experimental Results

Approach

DomsPlayers in my framework are defined by a list of tactics that work together to rank potential plays. The input from the tactics is combined in a commutative fashion, so the order of tactics is irrelevant.

Consequentially, players can be defined entirely by their tactic-list.

Tactics

- H = Simply play the highest scoring domino possible
- F = Prioritize (5,4) as the first domino to drop
- E = Look for plays that land on 59 or win outright
- B = Discourage dropping dangerous dominoes that can't be knocked off
- M = Prioritise plays exposing pips the player owns the majority of
- S = Predict the opponent's hand and what they could score in response

Selected Matchups

During the development of my tactics, I simply added each new tactic to the previously most complex player, yielding 6 players overall. These players were then benchmarked against every other player and the provided random player and their win percentage was calculated.

Each of the 49 matchups was simulated using `domsMatch` for 1,000,000 games (perhaps a bit overkill, but hey, I got to play with parallelism and optimised compilation) with a seed of 42 (naturally).

The results can be seen below. Each cell shows the percentage of the time that the row player beat the column player. I've also added a summary column that is just an average of the percentage in every column.

	Random	H	HF	HFE	HFEB	HFEBM	HFEBMS
Random	50.0225	2.981	2.8396	0.7404	0.7034	0.6713	0.275
H	97.0423	50.0138	49.1968	39.9489	39.5775	38.5602	30.3053
HF	97.1959	50.8382	50.0311	40.8585	40.5006	39.5016	30.6956
HFE	99.258	60.045	59.128	50.0287	49.6124	48.4298	36.8641
HFEB	99.2948	60.4014	59.4875	50.3954	49.9885	48.8348	37.1381
HFEBM	99.3315	61.5057	60.6219	51.7015	51.2873	50.0419	38.5675
HFEBMS	99.7275	69.871	69.4621	63.3092	63.0158	61.5742	49.9549

	Overall Win %
Random	8.319%
H	49.235%
HF	49.946%
HFE	57.624%
HFEB	57.934%
HFEBM	59.008%
HFEBMS	68.131%

Analysis

Every added tactic did improve the play of the DomsPlayer to some degree, but some of the tactics had a *much* more pronounced effect. Below is a ranking of the overall effect of the tactics (compared to the previous player):

1. H = +40.9%
2. S = +9.1%
3. E = +7.7%
4. M = +1.1%
5. F = +0.7%
6. B = +0.3%

Perhaps expectedly, choosing to play the highest scoring domino available is the most effective of the tactics.

That's not, however, to say that it cannot be improved upon; the next most important tactic was the intelligent prediction of the opponent's hand the threat that any particular move poses. This helps avoid the situation where a playing one domino allows the opponent to outscore you on the next turn.

A similarly impactful tactic was the E (end-game) tactic. While it's usually the case that more points is better, once you're within reach of 61 (or 59), the player needs to think about finishing exactly, which may mean playing a lower-scoring domino.

While it's likely not a game-changer, the M tactic (setting up the ends of the board for future plays), is one of the most "forward-looking" tactics. It's not quite as clever as S is with predictions, but takes the rest of play into account, not just the next move (as S does).

The tendency to drop (5,4) first (F) yielded almost a one percent improvement, showing that (5,4) is indeed a good way to start a game, but ultimately, playing first and drawing a (5,4) just doesn't happen often enough to see a bigger effect.

Finally, the real dud is the B tactic, which blindly avoids playing (6,6) and (5,5) in some scenarios. Without knowing what dominoes the opponent has, it's hard to determine if something is really dangerous. Note, however, this is a statistically significant effect! The self-matchups (the diagonal on the table) give us an idea of variance, as all of those matchups should average to around 50%. The biggest deviation from 50% along the diagonal is ~0.05% (thanks to the 1,000,000 game sample size), so 0.3% is certainly not just a chance effect.

Some Fun?

I mean, hey, the magic of computer programming is that you can perform really repetitive tasks with minimal effort, so why not test **every possible** tactic combination?

As it turns out, including the empty tactic list (similar to the random player), there are 64 different players that can be made with 6 tactics, which makes for 4094 matchups. I needed to drop the game number a little bit here, down to 10,000 games for each matchup, but the data is still very consistent.

You can check out the extra `collectData.hs` file for all of the code involved.

You'll obviously need to zoom this in quite a bit, or look at the separately included PDF!

Quick Analysis

Satisfyingly, many of the patterns we saw earlier are now visible as colour patterns. The most distinct pattern is the quadrant pattern which is caused by the presence or absence of the H (highest scoring) tactic.

Closely following H, most visible in the top-left quadrant are the stripes caused by the predictive S tactic. Interestingly, when two players without the H tactic are played against each other, having the S tactic leads to wins 70-92% of the time! This is a good example of the game-theory principle that what is bad for the opponent is good for the player in zero-sum games.