

Appendix B

Validation - Betting Parameters

These test cases can be found and run in the validation module of the project.

The results are based two AI agents playing against each other, with 20,000 hands played, per test case.

Process used to determine optimal values for parameters:

- Define a default range of values for the betting parameters.
- For each parameter, test against different value for that particular parameters and obtain value which seems to be performing the best.
- Create a new set of values for the parameters from these proposed values and test against previous default parameters.

The following series of tests are the tests ran and documented to produce the set of parameters that are used within the project.

The test classes can be found in the com.saccarn.poker.testbetpassvalues package in the validation module of the project. The values can be found in com.saccarn.poker.betpassvalues package of the same module.

The recorded results include the winning player, defined in the test case, the winning margin in terms of big blinds, and the variance of this margin.

Test against random distribution of parameters to determine default set of parameters

Test Case	Test Class (to run)	Values	Result
Default Values (Player One) vs BetPassValuesTest1 (Player Two)	TestDefaultVsBetPassValues1	DefaultValues BetPassValuesTest1	Player One won: 1.372 big blinds per hand +- 0.2354
Default Values (Player One) vs BetPassValuesTest2 (Player 2)	TestDefaultVsBetPassValues2	DefaultValues BetPassValuesTest2	Player One won: 1.4839 big blinds per hand +- 0.37847

Conclusion: Default Values seem good enough to try and improve on.

Pass parameter

Test Case	Test Class (to run)	Values	Result
Default Values (Player One) vs	TestDefaultVsBetPassValues3	DefaultValues BetPassValuesTest3	Player One won: 0.14175 big blinds per

BetPassValuesTest3 (Player 2)			hand +- 0.21417
Default Values (Player One) vs BetPassValuesTest4 (Player 2)	TestDefaultVsBetPassV alues4	DefaultValues BetPassValuesTest4	Player One won: 0.07745 big blinds per hand +- 9.14908

Conclusion: Default Pass Parameter seems to produce optimal results compared to other values

Bet2 parameter

Test Case	Test Class (to run)	Values	Result
Default Values (Player One) vs BetPassValuesTest5 (Player 2)	TestDefaultVsBetPassV alues5	DefaultValues BetPassValuesTest5	Player Two won: 0.018 big blinds per hand +- 0.1268 Player One won : 0.3248
Default Values (Player One) vs BetPassValuesTest6 (Player 2)	TestDefaultVsBetPassV alues6	DefaultValues BetPassValuesTest6	Player Two won: 0.1768 big blinds per hand Player One won: 0.482 +- 0.08276
BetPassValuesTest5 (Player One) vs BetPassValuesTest6 (Player Two)	TestBetPassValues5VsB etPassValues6	BetPassValues5 BetPassValues6	Player One won 0.2526 big blinds per hand +- 0.191511.
BetPassValuesTest5 (Player One) vs BetPassValuesTest7 (Player Two)	TestBetPassValues5VsB etPassValues7	BetPassValues5 BetPassValues7	Player One won: 0.344435 big blinds per hand +- 0.29577
BetPassValuesTest6 (Player One) vs BetPassValuesTest7 (Player Two)	TestBetPassValues6VsB etPassValues7	BetPassValues6 BetPassValues7	Player Two won: 0.001 big blinds per hand +- 0.2146
BetPassValuesTest5 (Player One) vs BetPassValuesTest8 (Player Two)	TestBetPassValues5VsB etPassValues8	BetPassValuesTest5 BetPassValuesTest8	Player Two won: 0.0095 big blinds per hand. 0.3748 +- 0.2327 (Player one)
BetPassValuesTest8 (Player One) vs	TestBetPassValues8VsB etPassValues9	BetPassValuesTest8 BetPassValuesTest9	Player One won: 0.19065 big blinds per

BetPassValuesTest9 (Player Two)			hand +/- 0.2878
BetPassValuesTest8 (Player One) vs BetPassValuesTest10 (Player Two)	TestBetPassValues8VsBetPassValues10	BetPassValues8 BetPassValues10	Player One won: 0.2283 big blinds per hand. 0.058 +/- 0.1753

Conclusion: Bet2 Parameter of BetPassValuesTest8 seems to produce optimal results

Bet3 Parameter

Test Case	Test Class (to run)	Values	Result
DefaultValues (Player One) vs BetPassValuesTest11 (Player Two)	TestDefaultVsBetPassValues11	DefaultValues BetPassValues11	Player One won: 0.3951 big blinds per hand +/- 0.1649.
DefaultValues (Player One) vs BetPassValuesTest12 (Player Two)	TestDefaultVsBetPassValues12	DefaultValues BetPassValues12	Player One won: 0.0575 big blinds per hand +/- 0.20893.
DefaultValues (Player One) vs BetPassValuesTest13 (Player Two)	TestDefaultVsBetPassValues13	DefaultValues BetPassValues13	Player One won: 0.0946 big blinds per hand +/- 0.1465.

Conclusion: Default Pass Parameter seems to produce optimal results compared to other values.

All In Parameter

Test Case	Test Class (to run)	Values	Result
DefaultValues (Player One) vs BetPassValuesTest14	TestDefaultVsBetPassValues14	DefaultValues BetPassValuesTest14	Player One won: 0.015173 big blinds per hand +/- 0.14198.
DefaultValues (Player One) vs BetPassValuesTest15 (Player Two)	TestDefaultVsBetPassValues15	DefaultValues BetPassValuesTest15	Player One won: 0.1109 big blinds per hand +/- 0.157876.
DefaultValues (Player	TestDefaultVsBetPassV	DefaultValues	Player Two won:

One) vs BetPassValuesTest16 (PlayerTwo)	alues16	BetPassValuesTest16	0.07265 big blinds per hand +- 0.08768
BetPassValuesTest15 (Player One) vs BetPassValuesTest16 (Player Two)	TesDefaultVstBetPassV alues15VsBetPassValue s16	BetPassValuesTest15 BetPassValuesTest16	Player Two won: 0.223845 big blinds per hand +- 0.12841.
BetPassValuesTest14 (Player One) vs BetPassValuesTest16 (Player Two)	TestBetPassValues14Vs BetPassValues16	BetPassValuesTest14 BetPassValuesTest16	Player Two won: 0.202245 big blinds per hand. +- 0.0736

Conclusion: All in Parameter in BetPassValuesTest16 seems to produce optimal results compared to other values

Obtained values integrated together:

Test Case	Test Class (to run)	Values	Result
DefaultValues (Player One) Vs ProposedDefaultValues 1 (Player Two)	TestDefaultValuesVsPr oposedDefaultValues	DefaultValues ProposedDefaultValues	Player Two won: 0.1066 big blinds per hand +- 0.1895.

Conclusion: New Proposed values beats the previous default values and should be used in the project.

Appendix C

Validation - Testing Common Hand influencing belief vs. Common Hand Not influencing belief

These test cases can be found and run in the validation module of the project.

The results are based two AI agents playing against each other, with 20,000 hands played, per test case.

The test classes can be found in the `com.saccarn.poker.testshandpotential` and `com.saccarn.poker.testcommonhandvalues` package in the validation module of the project. The values can be found in `com.saccarn.poker.commonhandvalues` and `com.saccarn.poker.handpotentialstraightvalues` package of the same module.

The recorded results include the winning player, defined in the test case, the winning margin in terms of big blinds, and the variance of this margin.

Test Case	Test Class (to run)	Values	Result
Using CommonHand class influence (Player 1) vs NO CommonHand class influence (Player 2)	TestCommonHandTrue VsCommonHandFalse	CommonHandValues.COMMON_HAND_TRUE, CommonHandValues.COMMON_HAND_FALSE	Player Two won: 0.2657 big blinds per hand +- 0.104236

Validation - Testing Hand Potential influencing belief vs. Hand Potential Not influencing belief

These test cases can be found and run in the validation module of the project.

The results are based two AI agents playing against each other, with 20,000 hands played, per test case.

Partial Straight/Flush recognition vs no partial Straight/Flush recognition

Test Case	Test Class (to run)	Values	Result
Using HandPotential class influence (Player 1) vs NO HandPotential class influence (Player 2)	TestHandPotentialTrue VsHandPotentialFalse	HandPotentialValues.HAND_POTENTIAL_TRUE, HandPotentialValues.HAND_POTENTIAL_FALSE	Player One won: 0.28625 big blinds per hand +- 0.161786

Appendix D

Opponent Model Evaluation

These test cases can be found and run in the validation module of the project.

The results are based two AI agents playing against each other, with 20,000 hands played, per test case.

The test classes can be found in the `com.saccarn.poker.testsopponentmodels` package in the validation module of the project. The values can be found in `com.saccarn.poker.opponentmodels` package of the same module.

The recorded results include the winning player, defined in the test case, the winning margin in terms of big blinds, and the variance of this margin.

Test Case	Test Class (to run)	Values	Result
Default Opponent Model (Player 1) vs 'Tight' Opponent Model (Player 2)	TestDefaultVsTightOpponentModel	Default Opponent Model Tight Opponent Model	Player 2 won : 0.114848 big blind per hand +- 0.14597.
Default Opponent Model (Player 1) vs 'Loose' Opponent Model (Player 2)	TestDefaultVsLooseOpponentModel	Default Opponent Model Loose Opponent Model	Player 1 won : 0.85546 big blinds per hand +- 0.2127229.
Default Opponent Model (Player 1) vs 'Unusual' Opponent Model (Player 2)	TestDefaultVsUnusualOpponentModel	Default Opponent Model Unusual Opponent Model	Player 1 won : 0.1326 big blinds per hand +- 0.1019436
Default Opponent Model (Player 1) vs 'Cluster 1' Opponent Model	TestDefaultVsCluster1Model	Default Opponent Model Cluster 1 Opponent Model	Player 1 won : 0.3749 +- 0.16 big blinds per hand
Default Opponent Model (Player 1) vs 'Cluster 2' Opponent Model (Player 2)	TestDefaultVsCluster2Model	Default Opponent Model Cluster 2 Opponent Model	Player 1 won : 0.087 big blinds per hand +- 0.1548
Default Opponent	TestDefaultVsCluster3	Default Opponent	Player 1 won : 0.6269

Model (Player 1) vs 'Cluster 3' Opponent Model (Player 2)	Model	Model Cluster 3 Opponent Model	+ - 0.1059 big blinds per hand
---	-------	--------------------------------------	-----------------------------------

Appendix E

Validation - Preflop values

Similar strategy used to determine values as for BetPassValues.

The test classes can be found in the com.saccarn.poker.testspreflopvalues package in the validation module of the project. The values can be found in com.saccarn.poker.preflopvalues package of the same module.

PreFlop Rank Value Parameter

Test Case	Test Class (to run)	Values	Result
Default Preflop Values (Player 1) vs PreFlopValuesTest1 (Player 2)	TestDefaultVsPreFlopV aluesTest1	DefaultValues PreFlopValuesTest1	Player 1 won : 0.2908 big blind per hand +- 0.1549.
Default Preflop Values (Player 1) vs PreFlopValuesTest2 (Player 2)	TestDefaultVsPreFlopV aluesTest2	DefaultValues PreFlopValuesTest2	Player 1 won : 0.29225 big blinds per hand +- 0.2586.
Default Preflop Values (Player 1) vs PreFlopValuesTest3 (Player 2)	TestDefaultVsPreFlopV aluesTest3	DefaultValues PreFlopValuesTest3	Player 1 won : 0.06945 big blinds per hand +- 0.0898.
Default Preflop Values (Player 1) vs PreFlopValuesTest4 (Player 2)	TestDefaultVsPreFlopV aluesTest4	DefaultValues PreFlopValuesTest4	Player 1 won : 0.1838 big blinds per hand +- 0.07141563.
Default Preflop Values (Player 1) vs PreFlopValuesTest5 (Player 2)	TestDefaultVsPreFlopV aluesTest5	DefaultValues PreFlopValuesTest5	Player 1 won : 0.1838 big blinds per hand +- 0.7141563.

PreFlop RandomFold Value Parameter

Test Case	Test Class (to run)	Values	Result
Default Preflop Values (Player 1) vs PreFlopValuesTest6 (Player 2)	TestDefaultVsPreFlopValuesTest6	DefaultValues PreFlopValuesTest6	Player 1 won : 0.0851 big blinds per hand +- 0.15556.
Default Preflop Values (Player 1) vs PreFlopValuesTest7 (Player 2)	TestDefaultVsPreFlopValuesTest7	DefaultValues PreFlopValuesTest7	Player 1 won : 0.02899 big blinds per hand +- 0.15367.
Default Preflop Values (Player 1) vs PreFlopValuesTest8 (Player 2)	TestDefaultVsPreFlopValuesTest8	DefaultValues PreFlopValuesTest8	Player 1 won : 0.16905 big blinds per hand +- 0.248.
Default Preflop Values (Player 1) vs PreFlopValuesTest9 (Player 2)	TestDefaultVsPreFlopValuesTest9	DefaultValues PreFlopValuesTest9	Player 1 won : 0.13625 big blinds per hand +- 0.10431.
Default Preflop Values (Player 1) vs PreFlopValuesTest10 (Player 2)	TestDefaultVsPreFlopValuesTest10	DefaultValues PreFlopValuesTest10	Player 1 won : 0.13755 big blinds per hand +- 0.15600.
Default Preflop Values (Player 1) vs PreFlopValuesTest11 (Player 2)	TestDefaultVsPreFlopValuesTest11	DefaultValues PreFlopValuesTest11	Player 1 won : 0.150755 big blinds per hand +- 0.15942.