Evaluating Ad Hoc Teamwork Performance in Drop-In Player Challenges

Patrick MacAlpine and Peter Stone

Department of Computer Science, The University of Texas at Austin

May 9, 2017

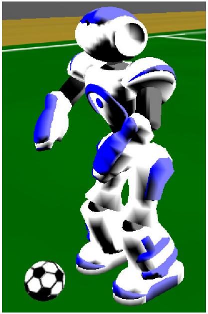
Learning Agents Research Group
The University of Texas at Austin

RoboCup Drop-In Player Challenges

- RoboCup is an international robotics competition where autonomous robots play soccer
- Games between teams consisting of different randomly chosen players from participants in the competition—pick-up soccer
- No pre-coordination between teammates, teammates/opponents unknown before start of a game
- Teams provided standard communication protocol for use during games
- Testbed for ad hoc teamwork
- Challenge held across three leagues at RoboCup competitions
  - Standard Platform League (SPL)
  - 2D Simulation League
  - 3D Simulation League

3D Simulation League

- Teams of 11 vs 11 autonomous agents play soccer
- Realistic physics using Open Dynamics Engine (ODE)
- Agents modeled after Aldebaran Nao robot
- Agents receives noisy visual information about environment
- Agents can communicate with each other over limited bandwidth channel



3D Simulation Drop-In Player Challenge

- Games are 10 vs 10 (no goalies)
- Full 10 minute games (two 5 minute halves)
- Participants contribute 2 drop-in players for a game
- Agents are provided a standard communication protocol
  - position of the ball
  - time ball last seen
  - position of the agent
  - if agent has fallen
- Score is average goal difference (AGD) across all games played

Example Drop-in Player Game

- No pre-coordination among agents

Blue: 2-3 UTAustinVilla, 4-5 Bahia3D, 6-7 Photon, 8-9 BoldHearts, 10-11 RoboCanes

Red: 2-3 magmaOffenburg, 4-5 L3MSIM, 6-7 SEUJolly, 8-9 Apollo3D, 10-11 FCPortugal

RoboCup 2015 Drop-in Player Challenge

AGD for each team in the drop-in player challenge when playing all possible parings of drop-in player games ten times (1260 games in total) and at RoboCup.

		At RoboCup (8 drop-in games played)		
Team	**AGD**	**Main Rank**	**Drop-in Rank**	**AGD**
UTAustinVilla	1.823	1	1	1.625
FCPortugal	0.340	3	3-6	-0.125
BahiaRT	0.182	4	3-6	-0.125
magmaOffenburg	-0.039	6	3-6	-0.125
FUT-K	-0.052	2	9	-0.625
RoboCanes	-0.180	7	7-8	-0.375
CIT3D	-0.361	9	2	1.125
HfutEngine3D	-0.501	10	3-6	-0.125
Apollo3D	-0.593	5	10	-0.875
Nexus3D	-0.620	8	7-8	-0.375

RoboCup 2015 Drop-in Player Challenge

AGD for each team in the drop-in player challenge when playing all possible parings of drop-in player games ten times (1260 games in total) and at RoboCup.

		At RoboCup (8 drop-in games played)		
Team	**AGD**	**Main Rank**	**Drop-in Rank**	**AGD**
UTAustinVilla	1.823	1	1	1.625
FCPortugal	0.340	3	3-6	-0.125
BahiaRT	0.182	4	3-6	-0.125
magmaOffenburg	-0.039	6	3-6	-0.125
FUT-K	-0.052	2	9	-0.625
RoboCanes	-0.180	7	7-8	-0.375
CIT3D	-0.361	9	2	1.125
HfutEngine3D	-0.501	10	3-6	-0.125
Apollo3D	-0.593	5	10	-0.875
Nexus3D	-0.620	8	7-8	-0.375

- There is a strong correlation between teams' performances in the drop-in player challenge and regular soccer
  - Spearmans's rank correlation for 2013-2015 drop-in player challenges: 0.58, 0.79, 0.73

RoboCup 2015 Drop-in Player Challenge

AGD for each team in the drop-in player challenge when playing all possible parings of drop-in player games ten times (1260 games in total) and at RoboCup.

		At RoboCup (8 drop-in games played)		
Team	**AGD**	**Main Rank**	**Drop-in Rank**	**AGD**
UTAustinVilla	1.823	1	1	1.625
FCPortugal	0.340	3	3-6	-0.125
BahiaRT	0.182	4	3-6	-0.125
magmaOffenburg	-0.039	6	3-6	-0.125
FUT-K	-0.052	2	9	-0.625
RoboCanes	-0.180	7	7-8	-0.375
CIT3D	-0.361	9	2	1.125
HfutEngine3D	-0.501	10	3-6	-0.125
Apollo3D	-0.593	5	10	-0.875
Nexus3D	-0.620	8	7-8	-0.375

- There is a strong correlation between teams' performances in the drop-in player challenge and regular soccer
  - Spearmans's rank correlation for 2013-2015 drop-in player challenges: 0.58, 0.79, 0.73
- Considerable noise makes it hard to evaluate agents after only a few games

Questions

- How to best measure/evaluate/score ad hoc teamwork?

- How to get more meaningful results in only a few games?

Questions

- How to best measure/evaluate/score ad hoc teamwork?
  - Instead of using AGD that rewards agents for being better skilled at individually playing soccer, try and isolate agents' ad hoc teamwork performance from skill level.

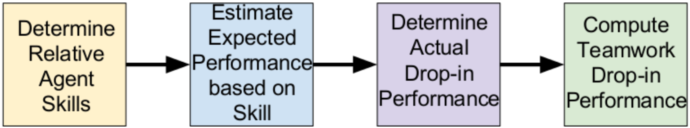
- How to get more meaningful results in only a few games?

Questions

- How to best measure/evaluate/score ad hoc teamwork?
  - Instead of using AGD that rewards agents for being better skilled at individually playing soccer, try and isolate agents' ad hoc teamwork performance from skill level.

- How to get more meaningful results in only a few games?
  - Predict scores of unplayed games based on results of games played to estimate results of all possible team permutations of games.

- How to best measure/evaluate/score ad hoc teamwork?
  - Instead of using AGD that rewards agents for being better skilled at individually playing soccer, try and isolate agents' ad hoc teamwork performance from skill level.



Skill Levels

- Walking speed of agents are limited to different percentages of maximum walking speed
- Everything else about agents are the same

Agents with different skill levels (maximum allowed walking speeds) running across the field

Normal (Good) Teamwork

- Only go to ball if closest member of team to ball

Agents displaying normal (good) teamwork

Poor Teamwork

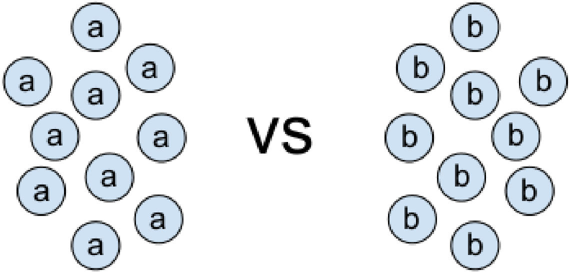
- Will go to ball even if another unknown teammate is closer to ball
- Unknown teammate = teammate who is not the exact same agent type—not having the same skill level and normal/poor teamwork attribute

Agents displaying poor teamwork

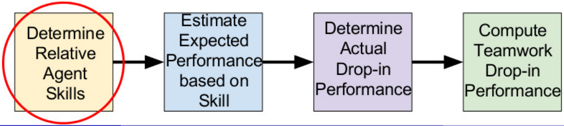
Determine Relative Skill Levels of Agents

Use AGD performance of two agents *a* and *b* playing against each other in drop-in player games with teams consisting entirely of their own agent as proxy for relative skill level between agents

$$\texttt{relSkill}(a, b)$$



Play round robin tournament of all agents against each other to determine `relSkill` of all agent pairs

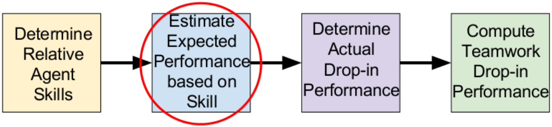


Compute Expected Skill AGD Across All Drop-in Games

Compute the expected AGD for each agent across all possible drop-in player game team pairings based on agents' relative skill levels.

$$\texttt{skillAGD}(a) = \frac{1}{K(N-1)} \sum_{b \in Agents \setminus a} \texttt{relSkill}(a, b)$$

where N is number of agents and K is number of agents per team



relSkill and skillAGD Values of Agents

AGD of agents when playing 100 games against each other. Number at end of agents' names refers to their maximum walk speed percentages. Positive goal difference means that row agent is winning.

	Agent60	Agent70	Agent80	Agent90
Agent100	1.73	1.36	0.78	0.24
Agent90	1.32	0.94	0.45	
Agent80	0.71	0.52		
Agent70	0.16			

Skill values (skillAGD) for agents.

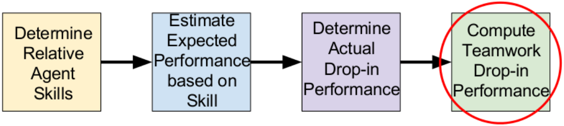
Agent	**skillAGD**
Agent100	0.183
Agent90	0.110
Agent80	0.000
Agent70	-0.118
Agent60	-0.174

- Agents with higher walk speed percentages have higher skillAGD

Isolate Ad Hoc Teamwork Performance from Skill Level

Subtract expected AGD based on agent's skill (`skillAGD`) from actual AGD across all permutations of drop-in player games (`dropinAGD`) to isolate adhoc teamwork performance (`teamworkAGD`).

$$\texttt{teamworkAGD}(a) = \texttt{dropinAGD}(a) - \texttt{skillAGD}(a)$$



`teamworkAGD` Values of Agents

`dropinAGD` values computed from playing total number of possible drop-in team combinations $\left(\left(\binom{10}{5} * \binom{5}{5}\right)/2 = 126\right)$ ten times for a total of 1260 games. PTAgents are those with poor teamwork.

Agent	`skillAGD`	`dropinAGD`	`teamworkAGD`
Agent100	0.183	0.204	0.021
Agent90	0.110	0.123	0.013
PTAgent100	0.183	0.109	-0.074
Agent80	0.000	0.087	0.087
Agent70	-0.118	0.017	0.135
PTAgent90	0.110	-0.018	-0.128
Agent60	-0.174	-0.055	0.119
PTAgent80	0.000	-0.101	-0.101
PTAgent70	-0.118	-0.169	-0.051
PTAgent60	-0.174	-0.196	-0.022

- Same speed agents have same `skillAGD` regardless of teamwork as functionally same when playing with all agents of same type

teamworkAGD Values of Agents

dropinAGD values computed from playing total number of possible drop-in team combinations $\left(\left(\binom{10}{5} * \binom{5}{5}\right)/2 = 126\right)$ ten times for a total of 1260 games. PTAgents are those with poor teamwork.

Agent	skillAGD	dropinAGD	teamworkAGD
Agent70	-0.118	0.017	0.135
Agent60	-0.174	-0.055	0.119
Agent80	0.000	0.087	0.087
Agent100	0.183	0.204	0.021
Agent90	0.110	0.123	0.013
PTAgent60	-0.174	-0.196	-0.022
PTAgent70	-0.118	-0.169	-0.051
PTAgent100	0.183	0.109	-0.074
PTAgent80	0.000	-0.101	-0.101
PTAgent90	0.110	-0.018	-0.128

- teamworkAGD ranks all agents with poor teamwork below other agents

teamworkAGD Values of Agents with Wider Skill Range

dropinAGD values computed from playing total number of possible drop-in team combinations $\left(\left(\binom{10}{5} * \binom{5}{5}\right)/2 = 126\right)$ ten times for a total of 1260 games. PTAgents are those with poor teamwork.

Agent	skillAGD	dropinAGD	teamworkAGD
Agent40	-0.710	-0.270	0.440
Agent50	-0.226	-0.129	0.097
Agent55	-0.142	-0.081	0.061
Agent100	0.412	0.416	0.004
PTAgent50	-0.226	-0.230	-0.004
Agent90	0.296	0.259	-0.037
Agent70	0.028	-0.005	-0.033
Agent85	0.245	0.176	-0.069
PTAgent70	0.028	-0.179	-0.207
PTAgent90	0.296	0.043	-0.253

- teamworkAGD no longer ranks all agents with poor teamwork below other agents

Normalized `teamworkAGD`

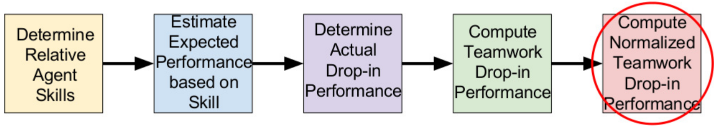
Add offset value to `teamworkAGD` to normalize same teamwork agents

$$\texttt{normTeamworkAGD}(a) = \texttt{teamworkAGD}(a) + \texttt{normOffset}(a)$$

For set of agents A with the same teamwork, and for every agent $a \in A$,

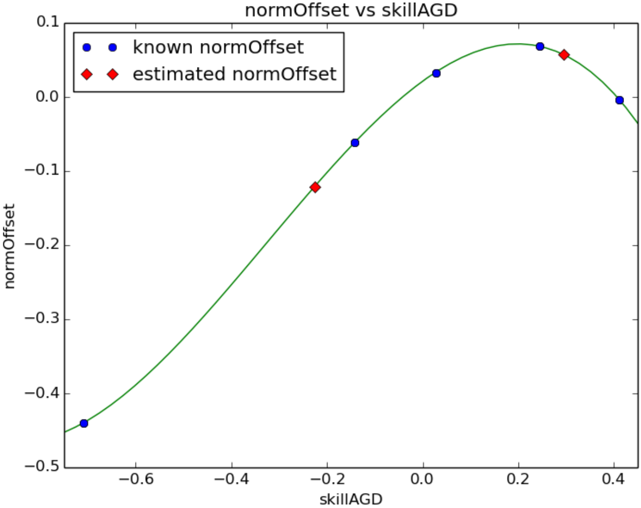
$$\texttt{normOffset}(a) = -\texttt{teamworkAGD}(a)$$

All agents with the same teamwork have the same `normTeamworkAGD` value = 0



Estimate `normOffset` for Other Agents

Plot and fit curve of `normOffset` vs `skillAGD` of same known teamwork agents to estimate `normOffset` values for other agents



Normalizing `teamworkAGD` to 0 for agent walk speeds 100, 85, 70, 55, 40
Estimating `normOffset` for agent walk speeds 50, 90

normTeamworkAGD Values of Agents

PTAgents are those with poor teamwork.

Agent	teamworkAGD	normOffset	normTeamworkAGD
Agent90	-0.037	0.057	0.020
Agent55	0.061	-0.061	0.000
Agent40	0.440	-0.440	0.000
Agent100	0.004	-0.004	0.000
Agent70	-0.033	0.033	0.000
Agent85	-0.069	0.069	0.000
Agent50	0.097	-0.121	-0.024
PTAgent50	-0.004	-0.121	-0.125
PTAgent70	-0.207	0.033	-0.174
PTAgent90	-0.253	0.057	-0.196

- normTeamworkAGD ranks all agents with poor teamwork below other agents

Results with Few Games

- How to get more meaningful results in only a few games?
  - Predict scores of unplayed games based on results of games played to estimate results of all possible team permutations of games.

Predict Scores of Unplayed Drop-in Player Games

Model drop-in player games as system of linear equations

Given two drop-in player teams A and B, `score(`A, B`)` is modeled as the sum of strength coefficients S,

$$\sum_{a \in Agents} S_a * \begin{cases} 1 & \text{if } a \in A \\ -1 & \text{if } a \in B \\ 0 & \text{otherwise} \end{cases}$$

teammate coefficients T,

$$\sum_{a \in Agents, b \in Agents, a < b} T_{a,b} * \begin{cases} 1 & \text{if } a \in A \text{ and } b \in A \\ -1 & \text{if } a \in B \text{ and } b \in B \\ 0 & \text{otherwise} \end{cases}$$

opponent coefficients O,

$$\sum_{a \in Agents, b \in Agents, a < b} O_{a,b} * \begin{cases} 1 & \text{if } a \in A \text{ and } b \in B \\ -1 & \text{if } a \in B \text{ and } b \in A. \\ 0 & \text{otherwise} \end{cases}$$

Predict Scores of Unplayed Drop-in Player Games

Solve for the $N + 2\binom{N}{2}$ coefficients using least squares regression

$$\sum S_1 + \sum T_1 + \sum O_1 = \texttt{score}(A_1, B_1)$$
$$\vdots$$
$$\sum S_n + \sum T_n + \sum O_n = \texttt{score}(A_n, B_n)$$

Need enough games for all coefficients to be multipled by non-zero value

Predicted `dropinAGD`

`dropinAGD` from all drop-in team pairing combinations compared to `dropinAGD` from half the team pairing combinations ($\frac{1}{2}$ `dropinAGD`), and predicted `dropinAGD` from half the team pairing combinations (Pred. `dropinAGD`). Difference (error) from true `dropinAGD` values shown in parentheses. PTAgents are those with poor teamwork.

Agent	`dropinAGD` 1260 games	$\frac{1}{2}$ `dropinAGD` 630 games	Pred. `dropinAGD` 630 games
Agent100	0.416	0.454 (0.038)	0.436 (0.020)
Agent90	0.259	0.356 (0.097)	0.296 (0.037)
Agent85	0.176	0.203 (0.027)	0.201 (0.025)
PTAgent90	0.043	0.105 (0.062)	0.048 (0.005)
Agent70	-0.005	-0.019 (0.014)	-0.016 (0.011)
Agent55	-0.081	-0.168 (0.087)	-0.132 (0.051)
Agent50	-0.129	-0.121 (0.008)	-0.098 (0.031)
PTAgent70	-0.179	-0.241 (0.062)	-0.173 (0.006)
PTAgent50	-0.230	-0.238 (0.008)	-0.241 (0.011)
Agent40	-0.270	-0.330 (0.060)	-0.323 (0.053)

- MSE: $\frac{1}{2}$`dropinAGD` $= 3.076 \times 10^{-3}$, Pred. `dropinAGD` $= 9.068 \times 10^{-4}$

RoboCup 2015 `normTeamworkAGD` Values of Agents

Values for `skillAGD` computed from every agent playing 100 games against each of the other agents with teams consisting of all the same agent. `dropinAGD` values computed using a prediction model built from the results of playing 1000 out of 378,378 possible drop-in player games.

Agent	skillAGD	dropinAGD	teamworkAGD	normOffset	normTeamAGD
UTAustinVilla	0.932	1.178	0.246	0.129	0.375
FCPortugal	0.384	0.262	-0.122	0.267	0.145
magmaOffenburg	0.038	-0.047	-0.085	0.139	0.054
Agent100	1.095	1.031	-0.064	0.064	0
Agent80	0.772	0.577	-0.195	0.195	0
Agent65	0.355	0.091	-0.264	0.264	0
Agent50	-0.278	-0.129	0.149	-0.149	0
Agent30	-1.456	-0.437	1.019	-1.019	0
BahiaRT	0.328	-0.029	-0.357	0.260	-0.097
RoboCanes	0.178	-0.199	-0.377	0.216	-0.161
FUT-K	0.520	0.029	-0.491	0.263	-0.228
Apollo3D	-0.533	-0.506	0.027	-0.465	-0.438
HfutEngine3D	-1.124	-0.470	0.654	-1.100	-0.446
CIT3D	-0.574	-0.589	-0.015	-0.519	-0.534
Nexus3D	-0.676	-0.763	-0.087	-0.653	-0.740

3D Simulation Drop-In Player Challenge Strategy (UT Austin Villa)

- Attempt to beam (teleport) in to take kickoff
- Go to ball if closest player otherwise stay behind ball in support role
- Evalute communicated information from teammates to determine if they're trustworthy

Blue player 2 and 3 from UTAustinVilla

Summary

- Possible to isolate players' skills from their teamwork in drop-in player challenges

Summary

- Possible to isolate players' skills from their teamwork in drop-in player challenges

- Assuming we have multiple agents with the same teamwork but different skill levels, we can use them to normalize the measure of agents' teamwork

Summary

- Possible to isolate players' skills from their teamwork in drop-in player challenges

- Assuming we have multiple agents with the same teamwork but different skill levels, we can use them to normalize the measure of agents' teamwork

- Can build a model from drop-in player game results to predict the scores of all unplayed team combinations of drop-in player games

Summary

- Possible to isolate players' skills from their teamwork in drop-in player challenges
- Assuming we have multiple agents with the same teamwork but different skill levels, we can use them to normalize the measure of agents' teamwork
- Can build a model from drop-in player game results to predict the scores of all unplayed team combinations of drop-in player games
- Combining `teamworkAGD` and a prediction model allows for evaluating adhoc teamwork in drop-in player challenges with only needing to play a small number of drop-in player games

Related Work

- Barrett, S., Stone, P.: An analysis framework for ad hoc teamwork tasks. In AAMAS, 2012
- Barrett, S., Stone, P., Kraus, S.: Empirical evaluation of ad hoc teamwork in the pursuit domain. In AAMAS, 2011
- Bowling, M., McCracken, P.: Coordination and adaptation in impromptu teams. In: AAAI, 2005
- Genter K., Laue T., and Stone P. Benchmarking robot cooperation without pre-coordination in the robocup standard platform league drop-in player competition. In: IROS, 2015
- Genter K., Laue T., and Stone P. Three years of the robocup standard platform league drop-in player competition: Creating and maintaining a large scale ad hoc teamwork robotics competition. In: JAAMAS, 2016
- Jones, E., Browning, B., Dias, M.B., Argall, B., Veloso, M.M., Stentz, A.T.: Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In: ICRA, 2006
- Liemhetcharat, S., Veloso, M.: Modeling mutual capabilities in heterogeneous teams for role assignment. In: IROS, 2011
- MacAlpine P., Genter K., Barrett S., and Stone P. The RoboCup 2013 drop-in player challenges: Experiments in ad hoc teamwork. In: IROS, 2014
- Stone, P., Kaminka, G.A., Kraus, S., Rosenschein, J.S.: Ad hoc autonomous agent teams: Collaboration without pre-coordination. In: AAAI, 2010
- Tambe, M.: Towards flexible teamwork. Journal of Artificial Intelligence Research 7, 1997
- Wu, F., Zilberstein, S., Chen, X.: Online planning for ad hoc autonomous agent teams. In: IJCAI, 2011

More Information

UT Austin Villa RoboCup 3D Simulation Homepage:
http://www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/

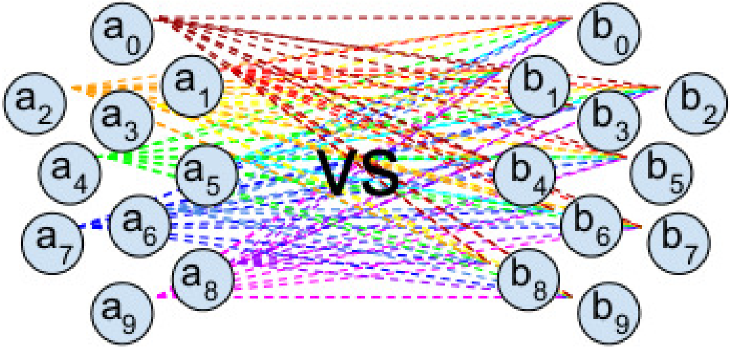
UT Austin Villa Code Release: https://github.com/LARG/utaustinvilla3d

Email: patmac@cs.utexas.edu

Compute Expected Skill Goal Differences for Mixed Agent Team Games

Estimate the goal difference of any mixed agent team drop-in player game by summing and then averaging the `relSkill` values of all agent pairs on opposing teams

$$\texttt{score}(A,B) = \frac{1}{|A||B|} \sum_{a \in A, b \in B} \texttt{relSkill}(a,b)$$



Compute Expected Skill AGD Across All Drop-in Games

Example: compute `skillAGD` of agent a for drop-in player challenge with agents $\{a, b, c, d\}$ and two agents on each team.

First determine the `score` of all drop-in game permutations involving agent a (`rS` used as shorthand for `relSkill`):

$$\texttt{score}(\{a,b\},\{c,d\}) = \frac{\texttt{rS}(a,c) + \texttt{rS}(a,d) + \texttt{rS}(b,c) + \texttt{rS}(b,d)}{4}$$

$$\texttt{score}(\{a,c\},\{b,d\}) = \frac{\texttt{rS}(a,b) + \texttt{rS}(a,d) + \texttt{rS}(c,b) + \texttt{rS}(c,d)}{4}$$

$$\texttt{score}(\{a,d\},\{b,c\}) = \frac{\texttt{rS}(a,b) + \texttt{rS}(a,c) + \texttt{rS}(d,b) + \texttt{rS}(d,c)}{4}$$

Compute Expected Skill AGD Across All Drop-in Games

Example: compute `skillAGD` of agent a for drop-in player challenge with agents $\{a, b, c, d\}$ and two agents on each team.

First determine the `score` of all drop-in game permutations involving agent a (`rS` used as shorthand for `relSkill`):

$$\texttt{score}(\{a,b\},\{c,d\}) = \frac{\texttt{rS}(a,c) + \texttt{rS}(a,d) + \texttt{rS}(b,c) + \texttt{rS}(b,d)}{4}$$

$$\texttt{score}(\{a,c\},\{b,d\}) = \frac{\texttt{rS}(a,b) + \texttt{rS}(a,d) + \texttt{rS}(c,b) + \texttt{rS}(c,d)}{4}$$

$$\texttt{score}(\{a,d\},\{b,c\}) = \frac{\texttt{rS}(a,b) + \texttt{rS}(a,c) + \texttt{rS}(d,b) + \texttt{rS}(d,c)}{4}$$

Averaging all scores to get `skillAGD`(a), and as

$$\texttt{rS}(a,b) = -\texttt{rS}(b,a),$$

`relSkill` values not involving agent a cancel out such that

$$\texttt{skillAGD}(a) = \frac{\texttt{rS}(a,b) + \texttt{rS}(a,c) + \texttt{rS}(a,d)}{6}.$$

Compute Expected Skill AGD Across All Drop-in Games

Based on `relSkill` values canceling each other out when averaging over all drop-in game permutations, the general simplified form is

$$\texttt{skillAGD}(a) = \frac{1}{K(N-1)} \sum_{b \in Agents \setminus a} \texttt{relSkill}(a, b)$$

where N is number of agents and K is number of agents per team

- Don't need to compute `score` for all possible $\left(\binom{N}{K} * \binom{N-K}{K}\right)/2$ drop-in player mixed team game permutations for an agent
- Only need `relSkill` values