

# The Empathy & Power Model ODD

J M Applegate

November 13, 2017

## Model Overview

The purpose of this model is to explore the effects of different power structures on a cross-functional team's prosocial decision making. A cross-functional team is a group of individuals (described as avatars in the model) from different functional backgrounds brought together to make one or more decisions. The power structure describes the distribution of power within the team. Empathy is characterised by the degree to which an avatar pays attention to the needs of other avatars. Are certain power distributions more conducive to the team making prosocial decisions? If so, does this tendency change over multiple decisions made by the same team?

Each decision is a choice between two options, A and B. Each avatar has a utility for each option, and the sum of the avatars' utilities for each option provides the overall team utility for each of the two options. The higher of these two values determines the prosocial option for the team.

Each avatar has a power value and will attempt to discern the utilities of all other avatars with equal or lower power values in making its own decision between A and B. Once avatars have made a decision they vote and the outcome is compared to the prosocial option. Avatars perceive the outcomes of decisions as *wins* or *losses*. If an avatar experiences a *win* in the decision making process it will increase the number of other avatars it pays attention to in the next round, while a *loss* reduces that number. Over the course of multiple decisions avatars increase and decrease their scope of attention (described by the permission variable) according to their *wins* or *losses*.

## Entities, State Variables and Scale

The model entities are two breeds of turtles: avatars and decisions. Patches are not explicitly used in this model.

Each avatar is characterised by a power value that persists across all decisions and a utility for each of the options, A and B, which changes each decision round. Power values are assigned based on a variety of distribution options specified by an interface menu: constant, uniform, normal, saddle, gamma or inverse-gamma, described in Figure 1. The number of power levels specified on the interface determines the range of power values.

Avatar's permission parameter is initially set to 0 and increases or decreases based on *wins* or *losses* in the decision process, affecting future decision making behaviour. The avatars which compose the team is specified by the `team_size` interface parameter.

A single decision is created each round with a color attribute indicating whether the decision was prosocial (green), non-prosocial (red) or a tie (yellow).

Each tick represents a decision-making round during which each

Scenario 1	1	2	3	4	5	group
utility A	8	5	4	1	7	25
utility B	3	7	2	6	2	20
decision	A	B	A	B	A	A = A
Scenario 2	1	2	3	4	5	group
utility A	2	6	8	3	1	20
utility B	1	5	7	9	6	28
decision	A	A	A	B	B	A ≠ B

Table 1: Two scenarios demonstrating the decision algorithm for a team of five avatars. In Scenario 1 option A is the prosocial outcome because the sum of the individual utilities for option A is greater than that of B, and based on individual utility values option A would be selected if individuals were voting based on their own utility, since three individuals would vote A. The vote winner matches the prosocial option. In Scenario 2 option B is now the prosocial outcome, and based on individual utility values option A would be selected if individuals were voting based on their own utility. The vote winner does not match the prosocial option.

avatar:

1. decides on an optimal choice between A and B based on a combination of their own utility values and those of fellow avatars to whom they pay attention,
2. votes in the team decision process and then
3. updates its permission value based on a perceived *win* or *loss*.

### Process Overview & Scheduling

On setup the number of avatars specified by `team_size` are instantiated (`instantiate` procedure) with utility values for each of two options A and B and a power value determined by the selected power distribution (`set_power_scores` procedure).

On go the sum of all the avatar utilities for each option are tallied and the larger of the two is determined to be the team's prosocial option for that round (`set_optimals` procedure). Two examples of the decision algorithm are given in Table 1.

Each avatar makes a decision between option A and option B based on its own utility values the utility values<sup>1</sup> for the avatars to which it pays attention (`estimate_optimal` procedure). Avatars initially pay attention to those avatars with a power value equal or higher than their own. The avatars come up with an optimal choice based on this calculation.

Perceptions of other avatars' utility values are subject to an error ranging from positive to negative `error_factor`, which is specified on the interface. An avatar's perception error is randomly generated from the specified range for each decision, and is applied to the other avatars' utilities in the optimal choice calculation.

Once all avatars have reached a decision about an optimal choice their choices are tallied as votes<sup>2</sup>, generating the team decision for that round (`tally` procedure). A decision entity is created color-coded according to whether or not the decision matches the team's prosocial option (`decide` procedure).

After each decision, avatars assess whether or not they achieved an individual *win* in the decision process by comparing the decision to their own originally preferred option, the option for which their own utility is higher (`adjust` procedure). Any decision that matches the avatar's own preference is considered a win, likewise any decision that does not match is considered a loss. A *win* will increase the avatar's permission value by 1 which means the avatar will consider the next lower power level during the following decision round.<sup>3</sup> A *loss* means the avatar will only consider its own utility next decision

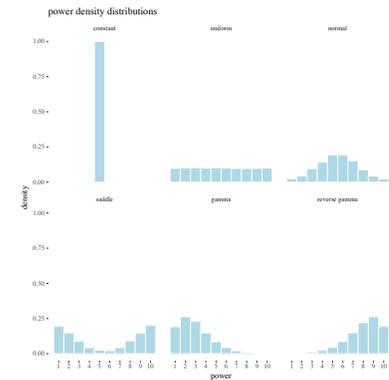


Figure 1: The model allows for the exploration of six power distributions: constant, uniform, normal, saddle, gamma and inverse-gamma. The distribution above are based on ten power levels.

<sup>1</sup> the calculation is either a power-weighted or direct accounting of other avatars utilities depending on the utility option set on the interface.

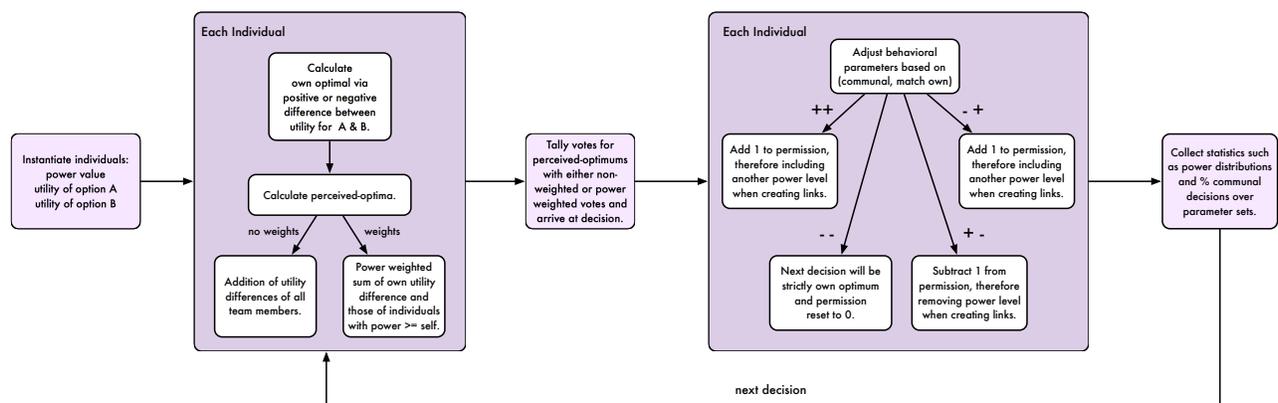
<sup>2</sup> Votes are tallied either directly or weighted by power depending on the voting option selected on the interface.

<sup>3</sup> Over multiple decision rounds it is possible that an avatar's permission value grows to include all other avatars on the team until it encounters a *loss*.

round and if the team decision was not prosocial its permission value is reset to 0.<sup>4</sup> A *loss* that resulted in a prosocial decision is a mitigated *loss* and permission is simply decreased by 1 rather than reset to 0.<sup>5</sup> Figure 2 demonstrates the payoff matrix that determines changes to avatar permission values. These behavior rules describe adaptive behavior in a formal power structure.<sup>6</sup> An informal power structure, such as one based on expertise, would suggest different adaptive rules.

The model continues to run until the specified number of decisions have been made.<sup>7</sup>

The model functionality is summarized in the flowchart in Figure 3.



<sup>4</sup> If the memory option is selected the avatar will encounter a negative permission value in the case of a *loss*.

<sup>5</sup> The model can be run with and without the permission functionality by setting the permissivity option to *off*.

<sup>6</sup> Joe C Magee and Adam D Galinsky. 8 social hierarchy: The self-reinforcing nature of power and status. *Academy of Management annals*, 2(1):351–398, 2008

<sup>7</sup> The interface presents an option to run experiments over multiple teams setups for a given parameter set.

Figure 3: Empathy Power Model Flow Diagram

### Design Concepts

**EMERGENCE** The trends in prosocial decision making are not determined analytically or a priori, rather these trends emerge through the repeated interactions of avatars in a repeated decision making process.

**ADAPTATION** Avatars will change their decision making behaviour based on the results of the decision making process. This change is captured in the avatar’s permission variable which changes over the course of repeated interactions.

**OBJECTIVES** Avatars seek to maximize their perceived utility by choosing one of two options.

**LEARNING** This model uses behaviour modification but not learning per se.

**SENSING** Avatars sense their environment through seeking information about the option utilities for multiple team members. Sensing is accomplished within a specified error.

		communal?	
		yes	no
match?	yes	win	win
	no	mitigated loss	loss

Figure 2: Avatars compare team decision to their own option preferences to determine whether there is a match, and update their permission value based on perceived *wins* or *losses*. Any decision in which the team decision matches the avatar’s own preference is considered a *win* regardless of whether that decision is prosocial or not.

**INTERACTION** Avatars do not explicitly interact with each other beyond sensing each other's option utilities.

**STOCHASTICITY** The model is stochastic in that `power_score` values are determined randomly based on selected distribution, utilities for options are determined randomly based on the uniform distribution, and perception error is also randomly determined within a specified range.<sup>8</sup>

**COLLECTIVES** The model is based on the idea of a collective, the team, consisting of the avatars. The collective behavior is of primary interest.

**OBSERVATION** From the collective level we observe the trends in prosocial decision making over time. Avatar specific information such as `optimal_choice` and `permission` may be captured at each timestep in order to explore the voting trends in specific power values in various distributions. The main interface graphic shows avatars with permission levels by color and with links from each avatar demonstrating which other avatars' utility values are being considered. The interface also shows timeseries for prosocial voting and permission over decision rounds as well as the power distribution for the current team.

<sup>8</sup>The effects of this particular application of stochasticity are explored in the Appendix.

## *Details*

### *Initialisation & Inputs*

The model does not use any external input files. The following input parameters and options are set on the interface with the exception of `permission` which is initialised at 0 on setup.

### *Submodels*

#### SETUP PROCEDURES & ALGORITHMS

**instantiate** Create avatars according to the value of `team_size`.

**normalise** Select normalised value for runs based on numerical solutions (see Appendix). These values are used to produce the normalised decision timeseries plot on the interface.

**power\_scores** Assign `power_score` to avatars based on selected distribution, redraw avatars with sizes according to power value, and create links between avatars with greater or equal power to indicate scope of attention.

#### GO PROCEDURES & ALGORITHMS

**set\_optimals** Set avatars' utility values for options A and B via `random_float 10` and set `own_diff` equal to the difference between utility values. The variable `own_optimal` is set to A if `own_diff > 0`,

Parameter	Inputs
team_size, number of avatars in the functional team	set on interface
rounds, number of team decisions	set on interface
power_level, the number of power levels used in the simulation	set on interface
power_distribution, distribution used to assign power values	set on interface
error_factor, the maximum error in determining other utility values	set on interface
permissivity, option to use permission in the model	toggle on interface
permission, the scope of attention toward other avatars of lesser power	0 initially
weighted_utility, option to weight utility calculation by power	toggle on interface
voting_algorithm, option to have vote weighted by power	toggle on interface

otherwise own\_optimal is set to B. Sum all avatars' own\_diff and if  $> 0$  set team optimal to A, otherwise B.

**estimate\_optimal** Set perception error as random-float value between  $[-error\_factor, error\_factor]$ . If the weighted\_utility option is selected, calculate utility by  $own\_diff \times power + \sum(\text{error} \times \text{power difference between self and neighbor} \times \text{neighbor's diff})$  for all neighbors. Else utility is  $own\_diff + \sum(\text{error} \times \text{neighbor's diff})$  for all neighbors. If perceived utility is  $> 1$  the perceived optimal choice is A, otherwise it is B.

**decide** If  $own\_optimal = optimal$  the avatar has a match between its own option preference and the team prosocial option. If an avatar's perceived optimal matches the team optimal the yeas variable is incremented by 1 for unweighted voting or by its power value for weighted voting.

**tally** The decision criteria is majority rule using either the weighted or unweighted value for yeas depending on the voting\_algorithm selection. With criteria representing the team median, if  $yeas > criteria$  a star-shaped decision is created with the color green, if  $yeas < criteria$  a decision is created with color red, and if  $yeas = criteria$  a decision is created with color yellow. The variables communal, non-communal and ties are incremented appropriately.

**adjust** For each avatar, if decision is prosocial and the avatar has a match between own\_option and team prosocial option, permission is incremented by 1 unless permission is already at max value.

If decision is prosocial but there is not a match between own\_option and team prosocial option, permission is decreased by 1 unless permission is already at min value.

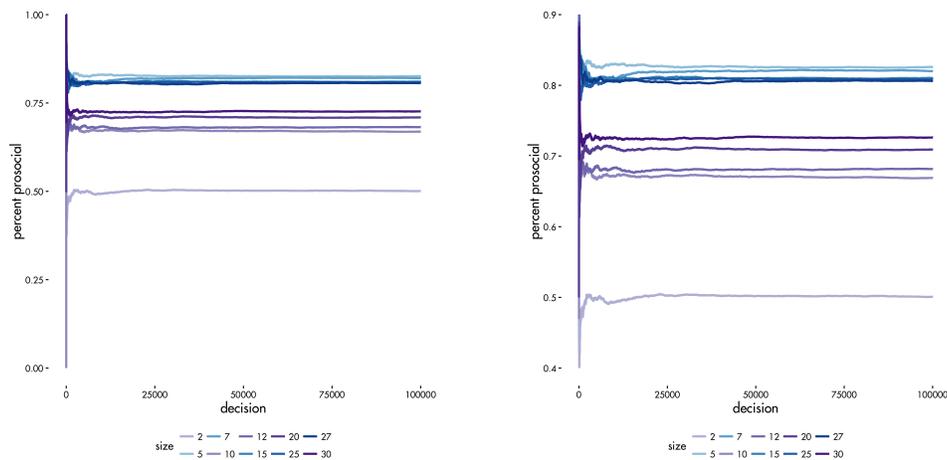
If decision is not prosocial but there is a match between own\_option and the team prosocial option, permission is incremented by 1 unless permission is already at max value.

If decision is not prosocial and there is not a match between own\_option and team prosocial option, permission = 0 and the vote\_own flag is toggled so that avatar will vote strictly own\_option next round. If memory is activated then permission = power - (power\_level + 1) and will become negative.<sup>9</sup>

<sup>9</sup> The following customised reporters allow extracting avatar-level data through BehaviorSpace: all\_who, all\_power, all\_own, all\_perceived, all\_permission and all\_links.

*Appendix: Baseline Numerical Solutions to Decision Algorithm*

Given that the prosocial option has a sum of utilities greater than the alternative option, it is more likely that the individual utilities will be greater, thus skewing decisions toward the prosocial outcome. What is the probability that the winner of the vote based on individual utility matches the prosocial option? Figure 4 shows the values at which prosocial decisions converge for various team sizes.



Numerical solutions to this question are given for group sizes from one to 30 in Table 2, with an even number of group members allowing for the possibility of a tie. The following figures show the baseline prosocial decisions, where the individual vote matches the prosocial outcome in the absence of any consideration of the utility

Figure 4: Decision baselines for selected group sizes. Even sizes are represented in purple, odd sizes are represented in blue. The figure on the left shows the baselines in the context of possible prosocial outcomes from 0 to 1, while the figure on the right shows the values in the restricted range of .4 to .9.

of other group members, for selected odd and even values of group size. As expected, a group size of two would end up with a prosocial decision 50% of the time.<sup>10</sup>

<b>1</b>	<b>1</b>	<b>11</b>	0.81364	<b>21</b>	0.80999
<b>2</b>	0.50091	<b>12</b>	0.68158	<b>22</b>	0.71415
<b>3</b>	0.83948	<b>13</b>	0.80987	<b>23</b>	0.80988
<b>4</b>	0.58467	<b>14</b>	0.69155	<b>24</b>	0.71889
<b>5</b>	0.82578	<b>15</b>	0.81031	<b>25</b>	0.80850
<b>6</b>	0.62554	<b>16</b>	0.69541	<b>26</b>	0.72113
<b>7</b>	0.82004	<b>17</b>	0.81160	<b>27</b>	0.80620
<b>8</b>	0.65295	<b>18</b>	0.70421	<b>28</b>	0.72518
<b>9</b>	0.81566	<b>19</b>	0.81047	<b>29</b>	0.80609
<b>10</b>	0.66891	<b>20</b>	0.70907	<b>30</b>	0.72626

Table 2: Baselines prosocial outcome values for groups of size 1 to 30.

<sup>10</sup> The normalise procedure reads the normalised value for the given team size from a table. This value is used to produce the normalised decisions plot on the interface, as well as running the optional baseline procedure which obtains these normalised values.

## References

Joe C Magee and Adam D Galinsky. 8 social hierarchy: The self-reinforcing nature of power and status. *Academy of Management annals*, 2(1):351–398, 2008.