Report CSCE 631-600, Fall 2025

Hugh Coleman
Texas A&M University

I. Motivation

Super Auto Pets (SAP) is a stochastic auto battler created by Team Wood Games in 2022. The game consists of various pets you can select who iteratively battle other players' teams for you. This paper focuses on analyzing single battles, in the first round of SAP. The core motivation is that SAP is a competition, and players want to perform better than others to avoid losing lives. SAP can be played in a battle royale format, or head to head against another player. The first turn is particularly important, as the player begins with considerable control over their strategic direction, with randomness steeply increasing at the start of the second turn.

II. Formal Definition

A. Sets and Structures

- Let $\mathcal{A} = \mathcal{H} = \mathbb{N}$ be the sets of possible base attack and base health values for pets.
- Let Π be a finite set of ability triggers (e.g., $\Pi = \{\text{start\_battle}, \text{faint}, \text{knockout}, \text{attack}, \dots\}$).
- Let $\mathcal{E}$ be the finite set of abilities, such as dealing damage to a random opponent, summoning a friend, or adding stats to a friend. These may be stochastic functions.
- The set of possible abilities and their trigger times is $\mathcal{B} = \mathcal{E} \times \Pi$

Definition II.1 (Pet)**.** A pet is a triple $p = (a, h, b) \in \mathcal{P} := \mathcal{A} \times \mathcal{H} \times \mathcal{B}$, encoding the base attack a, base health h, and behavior b of a pet.

Definition II.2 (Team)**.** A team is an ordered sequence of at most 5 pets

$$T := \bigcup_{k=0}^{5} \mathcal{P}^k,$$

where $T = (p_1, p_2, \dots, p_k)$ with $k \leq 5$, p_1 the front (first target/attacker), and p_k the back.

Definition II.3 (Outcome Space)**.** Let $\Omega = \{W, L, D\}$ be the outcome set of a battle from the perspective of the first team:

- W: Win – All pets on the second team have fainted, while ≥ 1 pet on the first team remains alive.
- L: Loss – All pets on the first team have fainted, while ≥ 1 pet on the second team remains alive.
- D: Draw – Both teams' pets all faint in the same round.

Let $\Delta(\Omega) := \{(p_W, p_L, p_D) \in [0,1]^3 \mid p_W + p_L + p_D = 1\}$ be the 2-simplex of outcome distributions.

Definition II.4 (Combat Function)**.** The combat function is

$$c : T \times T \to \Delta(\Omega),$$

where $c(T_1, T_2)[\omega]$ is the probability that a simulated battle between T_1 (Player 1) and T_2 (Player 2) ends in outcome $\omega \in \Omega$.

Definition II.5 (Battle State)**.** A battle state is a tuple $S = (T_1, T_2, Q, r)$ where:

- $T_1, T_2 \in T$ are the current teams for Player 1 and Player 2
- $Q \subseteq \mathcal{P} \times \{1, 2\} \times \mathcal{E}$ is an ordered queue of pending ability activations, where each element (p, i, e) represents pet p from team i triggering ability e
- $r \in \mathbb{N}$ is the current round number

Definition II.6 (Attack Function)**.** Let $\alpha : \mathcal{P} \times \mathcal{P} \to \mathcal{P}^* \times \mathcal{P}^*$ be the attack function where $\alpha(p_1, p_2)$ returns the resulting states of both pets after simultaneous attack. For $p_i = (a_i, h_i, b_i)$:

$$\alpha(p_1,p_2) \text{ where } \begin{array}{l} p_1' = \begin{cases} (a_1,\, h_1 - a_2,\, b_1) & \text{if } h_1 > a_2, \\ \varepsilon & \text{otherwise} \end{cases} \\ p_2' = \begin{cases} (a_2,\, h_2 - a_1,\, b_2) & \text{if } h_2 > a_1, \\ \varepsilon & \text{otherwise} \end{cases} \end{array}$$

Definition II.7 (Ability Triggering)**.** When a pet $p = (a, h, (e, \pi)) \in T_i$ experiences event $\pi' \in \Pi$, if $\pi = \pi'$, then the tuple (p, i, e) is enqueued to Q. Each ability $e \in \mathcal{E}$ is a (possibly stochastic) function $e : \mathcal{S} \to \mathcal{S}$ that modifies the battle state and may cascade additional triggers.

Definition II.8 (Battle Execution)**.** A battle proceeds through rounds via phase transitions. Each round r consists of:

1) **Start Phase**: If $r = 1$, enqueue all abilities with trigger start_battle. For all r, enqueue abilities with trigger start_round.
2) **Attack Phase**: Let p_1 and p_2 be the frontmost pets of T_1 and T_2. Compute $(p_1', p_2') = \alpha(p_1, p_2)$ and update teams. For each fainted pet, enqueue its faint ability. For the surviving attacker(s), enqueue any knockout or attack abilities.
3) **Ability Resolution Phase**: Process Q in FIFO order. Each ability e modifies S and may add new elements to Q. Continue until $Q = \emptyset$.
4) **Terminal Check**: Evaluate terminal condition. If satisfied, return outcome $\omega \in \Omega$. Otherwise, increment $r \leftarrow r + 1$ and repeat.

Definition II.9 (Terminal Condition)**.** Let $\tau : \mathcal{S} \to \Omega \cup \{\perp\}$ determine if state $S = (T_1, T_2, Q, r)$ is terminal:

$$\tau(S) = \begin{cases} W & \text{if } Q = \emptyset \text{ and } |T_1| > 0, |T_2| = 0 \\ L & \text{if } Q = \emptyset \text{ and } |T_1| = 0, |T_2| > 0 \\ D & \text{if } Q = \emptyset \text{ and } |T_1| = 0, |T_2| = 0 \\ \perp & \text{Continue} \end{cases}$$

III. Methodology

To compute the probabilities, repeat simulation was used to simulate the battles. There are 9 pets a player can have on their team, allowing for 729 unique teams. We simulated 1000 battles between each of the 729 teams, allowing us to analyze the stochastic elements of SAP. For data analytics, Python scripts were used on the extracted dataset to find relevant characteristics. To find the Nash Equilibrium (NE), we modeled SAP as a zero-sum game. We solved the linear programming by using SciPy linprog to minimize the linear objective function. For brevity, the explanations are purposefully kept concise. The code for this paper, and the implementation of each method is hosted on GitHub for viewing.

IV. Data Results

Each of these is a script that analyses the simulated battle dataset made by the simulator. The names of pets are abbreviated to their first letter (e.g., Ant → A, Mosquito → M, Otter → O, . . .)

A. Unbeaten Teams

The ideal situation for this project would be for one team to dominate others, never lose, and be clearly better than ever other team. This is not the case for SAP, there is no team that cannot take a loss. Furthermore, there is no perfect strategy that will always guarantee a win or even a tie in the first round.

B. Total Win Count

The total win count was calculated for each team as a metric for their performance. The top 10 teams with the most wins tallied are shown in the table below. These teams garner the most wins, but wins do not necessarily mean they are the best first round choice.

Rank	1	2	3	4	5	6	7	8	9	10
Team	A,A,C	F,A,C	A,A,F	A,F,C	M,A,C	A,F,F	A,M,C	F,A,F	A,A,O	A,A,M
Wins	1,399,386	1,363,923	1,305,266	1,296,239	1,279,992	1,272,926	1,258,910	1,252,869	1,250,868	1,241,624

TEAM COMPOSITIONS RANKED BY TOTAL WIN COUNT

C. *Total Loss Count*

The total loss count was calculated for each team as a metric for their performance. The top 10 teams with the least losses tallied are shown in the table below. These teams avoid taking losses, but again this does not necessarily imply that they are the best choice.

Rank	1	2	3	4	5	6	7	8	9	10
Team	F,A,F	A,A,F	F,A,M	A,A,O	F,F,F	A,A,M	F,A,C	A,F,F	F,A,B	A,A,C
Losses	2,147	3,789	16,089	16,923	16,947	17,300	18,580	18,796	19,996	21,655

TEAM COMPOSITIONS RANKED BY TOTAL LOSS COUNT

D. *Never Swept*

To avoid being swept in a given battle, a team needs at least some chance of winning or tying a battle. There are 30 teams that are never swept by another team, all of which possess high volatility. Every team that is not swept has at least one pet with stochastic randomness in its ability. There is at least one Mosquito in 22 of the 30 unswept teams. There is at least one Ant in 25 of the 30 unswept teams.

E. *Always One Win*

Analysis was done to find teams that have a chance of winning every battle. Precisely 3 teams have the unique designation of going into every battle with non-zero probability of winning. These are **(A, M, C)**, **(A, M, M)**, and **(A, A, M)**. It is worth noting that all three of these teams are entirely composed of members with abilities, $8/9$ of the pets have an ability with a stochastic element. These pets exhibit volatile performance characteristics, demonstrating imperfect reliability while perpetually maintaining non-zero win probabilities.

F. *Unique Losses*

We found it interesting to determine which teams had the least teams that it could lose to. For the most part, there is a strong correlation between total losses and unique losses. There is a notable outlier in **(F, A, M)** having the 3rd lowest total losses, yet being 10th for unique losses.

Rank	1	2	3	4	5	6	7	8	9	10
Team	F,A,F	A,A,F	F,F,F	F,A,C	F,F,B	A,A,O	F,A,B	F,A,A	F,A,O	F,A,M
Unique Losses	6	10	22	27	28	29	32	36	36	38

TEAM COMPOSITIONS RANKED BY UNIQUE LOSS COUNT

G. *Bradley Terry Rankings*

Bradley-Terry (B-T) rankings provide a model for pairwise comparisons of strength between SAP teams. While we have the benefit of a full extensive simulation with all head to head results computed, B-T helps provide a traditional team strength rankings to this paper. The top 3 for total win count and B-T are the same, providing strong evidence for these teams strength.

Rank	1	2	3	4	5	6	7	8	9	10
Team	A,A,C	F,A,C	A,A,F	A,F,F	F,A,F	A,F,C	A,A,O	A,A,M	M,A,C	F,F,F
B-T Strength	87.85	60.32	41.36	29.86	29.35	28.98	26.64	25.33	24.51	22.98

TEAM COMPOSITIONS RANKED BY BRADLEY-TERRY STRENGTH

V. NASH EQUILIBRIUM

We computed a NE for the first round of SAP in a head to head match. It is optimal to play a strategy consisting of **(F, A, C)**, **(A, A, F)**, and **(F, A, F)**. The ratio is displayed in the table below.

Team	F,A,C	A,A,F	F,A,F
Probability	38%	38%	24%

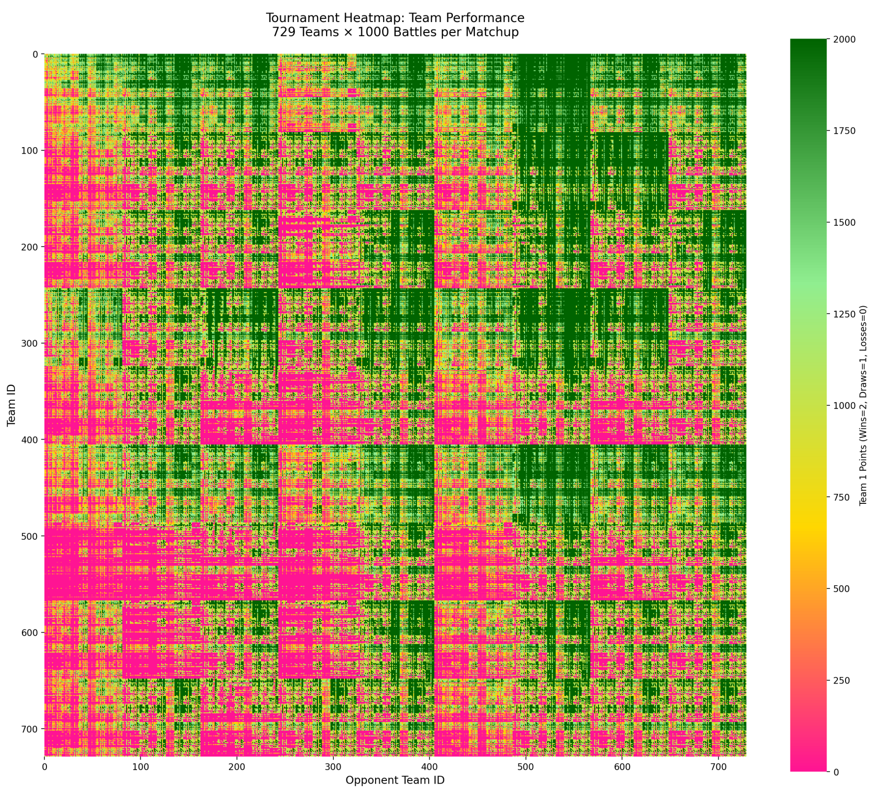
NASH EQUILIBRIA FOR SUPER AUTO PETS

First we noted that every one of these teams is an extremely high performer in the previously analyzed metrics. It is especially notable that the teams comprising the NE have the 1st, 2nd, and 4th lowest unique loss counts.

Team	F,A,C	A,A,F	F,A,F
Total Wins	2nd	3rd	8th
Total Losses	7th	2nd	1st
Never Swept			X
Unique Losses	4th	2nd	1st
B-T	2nd	3rd	5th

NE SUPPORT PERFORMANCES IN ANALYZED CATEGORIES

VI. HEAD-TO-HEAD IMAGE



CODE

code available at **github.com/hjc2/sap-final**

ACKNOWLEDGMENTS

Thanks to Team Wood Games for making the game Super Auto Pets, and for keeping it free and cross-platform.