StarCraft strategy classification of a large human versus human game replay dataset

Štefan Krištofík, Peter Malík
Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
Email: stefan.kristofik@savba.sk

Matúš Kasáš
Tatramed Software
Líščie údolie 9, 841 01 Bratislava, Slovakia

***Abstract*—Real-time strategy games are popular in AI research and education. Starcraft: Brood War (SCBW) is particularly well known among such games. Recently, the largest known SCBW game replay dataset STARDATA was published by Facebook. We classify player strategies used in the dataset for all three playable races and all 6 match-ups. We focus on early to mid-game strategies in matches which resolved in less than 15 minutes. By mapping the classified strategies to the replay files, we label the files of the dataset and make the labeled dataset available.**

I. Introduction

IN a competitive one on one real-time strategy (RTS) game environment, players try to outsmart and defeat their opponent by using superior strategy. This involves planning and execution of short term goals, e.g., effective military unit placement, combat or scouting, as well as long term goals, e.g., territory expansion or army composition. In turn-based strategic board games such as Chess or Go, the entire game state is always known to both players during a match. This is not the case in RTS where players can see only parts of a map near their own units and structures. Therefore, the opponent's intents and army placement can only be deducted from the limited information that is available at each time during a match and players often have to make decisions under uncertainty. Techniques used in board games can not be applied to RTS because the state space and the number of possible actions a player can make at each decision cycle is overwhelming [3]. These challenges (partial observability and huge complexity) are the main reasons why RTS are considered very challenging for AI.

The research into RTS AI began to flourish in 2003 after the initial call [1]. Since that time, it has become more clear that efficient solutions to challenges posed by RTS game environments can be helpful in many aspects of our lives. In the video gaming industry, players can have more challenging and rewarding experience. Various forecasts (weather, finance, road traffic, public transport) can be done more precisely. Combat simulation can be carried out before deploying military personnel into hostile or partially known territories which may help to make better decisions in field. In general, complex dynamic systems where agents are required to make fast real-time decisions based on incomplete information benefit the most form the in-game simulations of RTS AI agents [3], [5].

This work is supported by Slovak national project VEGA 2/0155/19.

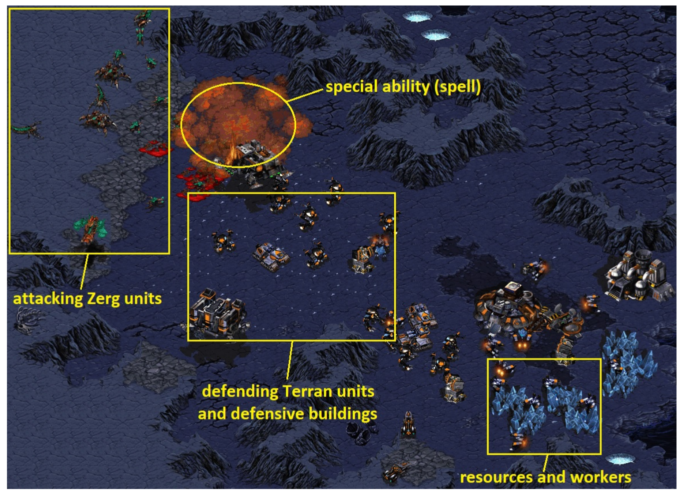

Fig. 1. StarCraft: Brood War

This research area has gained traction over the years and it has attracted many individuals and small teams, but also the tech giants Facebook, Microsoft and Google [4].

In terms of popularity of RTS AI research, the StarCraft franchise is dominating. The most successful [11] and widely known game is StarCraft: Brood War (SCBW), the first of two games, released in 1998 (Fig. 1). Despite its fairly old age, the community built around it is still very active. The competitive side of the community has been praising the unique and fair balance [10] of all three playable races (Protoss, Terran, Zerg) which is rarely accomplished in gaming and is one of the main reasons for the game's longevity. The research side of the community benefits from this longevity and over the years many tools were developed to help with AI research, including BWAPI [1].

StarCraft II is the sequel to SCBW released in 2010, but its research community has not yet grown as much as that of SCBW. Some very impressive results were achieved recently by the Google Deep Mind team [5]. However, the solution was extremely resource hungry in terms of electricity, hardware, time and personnel requirements. Such costs (estimated in tens of millions of dollars [2]) can only be justified for a one-time

[1]https://github.com/bwapi/bwapi
[2]https://www.youtube.com/watch?v=20DIg71Oma0

proof of concept type project. On top of that, the resulting AI is not able to compete at the highest level as it has not been shown that it can consistently beat the most skilled human players [3].

Nowadays, it is pretty common for many games to have annual AI competitions [11]. SCBW is no exception. Each year tournaments are organized to compare SCBW agents, e.g., "Student StarCraft AI Tournament and Ladder (SSCAIT) [4], "BASIL Ladder" [5], "AIIDE StarCraft AI Competition" [6], "IEEE CoG StarCraft AI Competition" [7]. This is also common in other AI disciplines, such as image classification [9] or object detection [8]. Based on the results of recent competitions, it can be concluded that SCBW AI agents are currently still not yet able to consistently defeat expert level human players and even can struggle against lower tier players [4]. Current goal of ongoing SCBW AI research is the continuous improvement of agents and ultimately overcoming human expert players on a consistent basis.

One of possible approaches to agent improvement is machine learning from the past matches. SCBW allows archiving of played matches in the form of replay files. Over the years, a vast amount of such data was accumulated. However, it is scattered among many sources with various levels of quality. In case of SCBW, for machine learning purposes a dataset of game replay files should meet multiple requirements to be considered viable [2], [10]. Recently, the largest known SCBW replay dataset called STARDATA which adheres to these requirements was published by the Facebook team [2]. It contains 65646 human versus human game replays and is the largest dataset compiled to date. Other datasets with similar purposes containing up to 7649 replays were introduced previously [6], [7], [8]. Multiple requirements of STARDATA were checked in the original work. Recently, also the strategy diversity requirement of the dataset was verified [10] which further proves its quality.

Following are the main contributions of this work. We thoroughly analyze the STARDATA dataset, classify strategies used by both players in each match and label the replay files with identified strategies. We make the labeled dataset available [9]. This may be helpful for future machine learning attempts.

II. Related Work

Multiple attempts at strategy classification from SCBW replays were conducted. Weber [6] extracts 6 strategies per race from 5493 replays, but does not consider each race in different match-ups and identifies only global strategies independent on the match-up. Cho [7] expands on [6] by enlarging the dataset by 570 more replays and also experiments with the fog of war information with moderate results. Synnaeve [8] extracts from 7649 replays, but focuses mainly on short term tactical aspects of the game. Lin [2] also attempts strategy extraction, but only few strategies were identified for only one race.

Krištofík [10] analyzes STARDATA and classifies 30 strategies (10 for each opposing race), but only for the Terran race. However, the main focus is on building, training and testing of an AI agent using identified strategies. In this work, we expand the idea introduced in [10]. We now consider and classify strategies of all three playable races in all 6 match-ups.

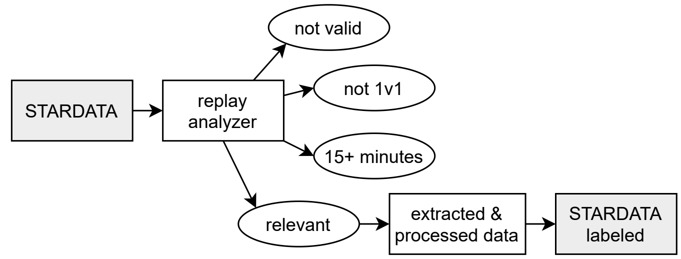

Fig. 2. Strategy classification and labeling

III. Strategy Classification

This work deals with strategy classification from the SBCW replay dataset STARDATA. Strategy classification was identified as one of the tasks which the dataset is suitable for [2]. Having such information could be helpful for future machine learning attempts and improvement of SCBW AI agents. We provide strategy information in the form of labeled files. The overall classification and labeling process is shown in Fig. 2. The following sections will describe the process step by step.

A. Dataset

We use STARDATA for strategy classification. It is a high quality collection of 65646 replays of human versus human matches gathered from various sources (websites). It meets many requirements of a good base for machine learning, including diversity, universality and validity [2], [10]. Initially, only a small portion of the dataset is partially labeled. 3321 (5 %) of replay files indicate the match-up, e.g., *TL_TvZ_GG20105.rep* indicates the match-up is Terran versus Zerg (TvZ). No other useful information can be gathered directly from file names.

B. Dataset cleaning and filtering

In this work, we are interested in valid competitive 1v1 matches and early to mid-game strategies. The first step is to clean the possible invalid replays from the dataset. Next step is to filter out non-competitive and overly long matches from the remaining valid data. Then the remaining relevant data will be processed further. For the match length threshold, we chose the value of 15 minutes for same reasons as described in [10].

We use our own tool called BWAPI replay analyzer to automatically process the original replay files of STARDATA. It utilizes BWAPI which is an open source API for SCBW that allows to interact with the game engine and is useful

[3] https://www.nature.com/articles/d41586-019-03298-6
[4] https://sscaitournament.com
[5] https://basil.bytekeeper.org
[6] http://www.cs.mun.ca/~dchurchill/starcraftaicomp/index.shtml
[7] https://cilab.gist.ac.kr/sc_competition
[8] https://www.kaggle.com/c/global-wheat-detection
[9] www2.fiit.stuba.sk/~kristofik/STARDATA_labeled.zip

TABLE I
DATASET CLEANING AND FILTERING

Replays	Amount
STARDATA Total	65646
Not valid (BWAPI incompatible)	31
Not competitive (not 1v1)	1075
Longer than 15 minutes	30429
Relevant for classification	**34111**

TABLE II
MATCH-UP DISTRIBUTION. P-PROTOSS, T-TERRAN, Z-ZERG

Dataset	PvP	PvT	PvZ	TvT	TvZ	ZvZ
STARDATA [2]	7015	17385	18016	2550	14531	6149
this work	4561	6797	9011	984	7410	5348

for AI research. It allows to setup matches between two AI agents, but can also play back a replay file and gather useful information from the match. Each replay file is sequentially run through BWAPI.

1) Validation: First, each replay is checked for validity; if it can run correctly in BWAPI. Invalid replay files either cause the engine to crash or the replay is corrupted and runs indefinitely. Such files are then manually removed from the relevant file pool and excluded from further processing.

2) Filtering: Next, if a replay is valid it is checked for competitiveness and length. We remove from the relevant file pool matches that have more than 2 players and/or ale longer than 15 minutes. Those files are excluded from further processing.

The results of the dataset cleaning and filtering process are summarized in Table I. 34111 replays remained in the pool of replays relevant for strategy classification. Match-up distribution of relevant replays is shown in Table II as well as the distribution in the entire dataset for comparison. From the table it can be observed that the variety of match-ups is kept intact in this work.

C. Raw information extraction

To extract raw detailed information from the relevant replay files and store in into json files (one json for each replay file), we created a modified version of the replay extractor for the Terran race introduced in [10]. In addition to all functionality of the original version, the modified version has the following new features:

- Extracts information from all 6 match-ups.
- Automatically categorizes extracted information by match-up.
- Maps extracted information to replay files.

D. Information processing

We further process the raw information stored in json files and prepare it for strategy classification. SCBW strategies can be characterized mainly by [10] a) build orders, which are sequences of building construction. For example, to be able to produce Zealots, a Protoss player has to first build a Pylon, then a Gateway; and b) army compositions, which are lists of unit types that form the backbone of the player's army, i.e., the most used unit types. For example, one of the more popular Terran army compositions against Zerg opponents is Marines with Medics with the support of few Siege Tanks and Science Vessels [10].

With the above in mind, from each json file, we extract following information:

- Basic information: file name, player names and races, match length [11], map name, winner [10]. Needed for mapping of replay files to identified strategies (main goal of this work) and is also useful for collecting various statistics.
- For both players: count of all structure types. Needed for build orders.
- For both players: relative count of all unit types per minute. This is computed as total count divided by match length in minuts. Needed for unit frequency statistics (explained in III-E).
- For each structure and unit type: timestamp [12] of first occurrence. Needed for build orders and unit frequency statistics.
- Auxiliary information: unit upgrades, tech. Currently not used.

The information is consolidated and stored into 6 csv files (one for each match-up). Each file stores information about one match per line.

E. Strategy classification

We classify strategies of all three races in all 6 match-ups. Because each race would use different set of strategies against different opponents, we divide strategies into 9 categories, summarized in Table III. For example, the Protoss race would use different set of strategies against Terran (PT in Table III) than against Zerg (PZ in Table III). The table also shows summary numbers on how many players used the strategies from each category.

Based on our domain knowledge and experience with the game, we have selected a set of most important structures and units for each race which will be used to define various strategies. The list is shown in Table IV. Each race can create more structure and unit types, but those were not selected for different reasons: a) being mandatory to progress in a match (created always), e.g., workers, Zerg larvae, supply limit increasing structures [13]; b) being unpopular and created very rarely, e.g., Protoss Scout, Zerg Devourer; c) being used in late game, e.g., Terran Battlecruiser, Zerg Ultralisk.

Considering only structures listed in Table IV, we compute for each match in each csv file the build order in the following way:

[10]https://www.youtube.com/watch?v=qyixL9J7-B8

[11]Measured in game frames. Competitive SC:BW games are played at 23.81 frames per second.

[12]In frames.

[13]Supply value represents the current size of the player's army.

TABLE III
STRATEGY CATEGORIZATION AND USAGE

Abbr.	Race	Match-up	Players
PP	Protoss	PvP	9122
PT	Protoss	PvT	6797
PZ	Protoss	PvZ	9011
TT	Terran	TvT	1968
TP	Terran	PvT	6797
TZ	Terran	TvZ	7410
ZZ	Zerg	ZvZ	10696
ZP	Zerg	PvZ	9011
ZT	Zerg	TvZ	7410

TABLE IV
SELECTED STRATEGY DEFINING STRUCTURES AND UNITS

Race	Structures	Units
Terran	Academy Armory Command Center Comsat Station Control Tower Engineering Bay Factory Machine Shop Science Facility Starport Refinery	Marine Vulture Goliath Siege Tank Wraith Medic Firebat
Protoss	Nexus Cybernetics Core Gateway Forge Templar Archives Stargate Robotics Facility	Dragoon Zealot High Templar Dark Templar Carrier
Zerg	Hatchery Lair Spawning Pool Hydralisk Den Spire	Zergling Hydralisk Lurker Mutalisk Scourge

- Assign value 1 to the first structure type to be constructed by a player during a match.
- Assign value 2 to the second, 3 to third, etc., up to the number of selected structures from Table IV.
- Assign value 1 higher than the number of selected structures from Table IV to all structure types never constructed during a match.

Example: A Zerg player built Hatchery as first, Spawning Pool as second and Hydralisk Den as third. They never built Lair nor Spire. The assigned values will be as follows: Hatchery 1, Spawning Pool 2, Hydralisk Den 3, Lair 6, Spire 6.

Considering only units listed in Table IV, we also compute for each match in each csv file the unit frequency statistics in the following way:

- Assign values from 1 up to the number of selected units from Table IV depending on the relative frequency during a match. Unit types created with higher frequency get lower values and those created with lower frequency get higher values.
- Assign value 1 higher than the number of selected units from Table IV to all unit types never created during a match.

Example: A Protoss player created many Dragoons during a match. They also created slightly less Zealots and about the same number of Dark Templars as Zealots. They also created a small number of High Templars. They never created Carriers. The assigned values may be roughly as follows: Dragoon 1, Zealot 2, Dark Templar 2, High Templar 5, Carrier 6.

For strategy classification, we treat STARDATA as unlabeled data because strategies used by both opponents are unknown. We perform classification on these unlabeled data by the K-Means clustering algorithm, inspired by [10]. The goal is to identify some regularities in the data. By grouping similar data into clusters, we can differentiate between various strategies. Each cluster will represent a distinct strategy. The number of desired clusters needs to be specified beforehand. We chose 10 for each match-up. This number resulted in sufficient diversity of clusters and also sufficient abundance of replays per cluster. The algorithm produces differently sized clusters. The more popular a strategy is the larger the cluster representing it will be.

This resulted in a total of 90 strategies, 30 for each race, 10 per category from Table III. The results are summarized and discussed in the next section.

IV. RESULTS

Classified strategies for Protoss are summarized in Fig. 3, for Terran in Fig. 4 and for Zerg in Fig. 5. Strategy distributions are shown in Fig. 6.

A. *Strategy descriptions*

Strategies in Figs. 3-5 are named based on categories from Table III and the cluster number assigned by the K-Means algorithm. For example, TP6 is a Terran player strategy used against Protoss with the cluster number 6 assigned by K-Means. The amounts of players that used each strategy are shown in column *count*. Columns *average structure order* show the build order values (explained in III-E) averaged over all the matches in each cluster. Columns *average unit frequency* show the unit frequency values (explained in III-E) averaged over all the matches in each cluster.

Column *brief description* gives a short verbal description of the strategies. Descriptions focus on different aspects of strategies. In general, we are interested in the following information about each strategy.

1) Most used units: Examples: For Protoss, *DZ* means the most used units in matches are Dragoons and Zealots (e.g., PP9). For Zerg, *M* means the most used unit is Mutalisk (e.g., ZT7).

2) Other used units: Example: For Terran, *often WMV* means Wraiths, Marines and Vultures are used very often in matches, but are not the most used units (e.g., TT4).

3) Significant structures: If the player has built some particular structures, this may indicate they are going to produce some specific units excluded from the selected list in Table IV. Examples: If a Protoss player builds Robotics Facility, it might indicate the intent to produce Reaver units (excluded) later in the game (e.g., PP0). If a Zerg player builds Lair (e.g., ZP6), it might indicate the intent to produce more advanced late game units like Ultralisks or Defilers (both excluded).

4) Economic expansion strategies: These are such kind of strategies that try to expand economically very early and gain an income advantage over the opponent. This means building Nexus for Protoss, Command Center for Terran and Hatchery for Zerg. Examples: *fast exp* indicates this type of strategy while *late exp* does not indicate it.

5) Rush strategies: These are such kind of strategies that try to end the match as soon as possible by attacking the opponent very early and catching them unprepared. Examples: *Cannon rush* for Protoss means building offensive structures (Photon Cannons) near the opponent's base (e.g., PZ3), *Z rush* for Zerg, creating many cheap fast units to attack and overwhelm the opponent (e.g., ZP1).

B. Discussion

The results for each of three races show the good variety among identified strategies. Not only 'normal' strategies are represented, but also some rush as well as economic strategies are in the mix.

However, it might indicate some of the selected strategy defining structures or units were probably chosen incorrectly. For example, Carriers for Protoss are almost never used in any strategy and contribute almost no useful information to strategy classification, as it is more of a late-game unit and we have a 15 minute threshold for classification. On the other hand, for example, Spawning Pools for Zerg are almost always built in all strategies and also contribute no useful information to strategy classification. This may be caused by the fact that it is the mandatory building for the Zerg race and without it, a player's options are very limited.

In conclusion, it is worth noting that the selection (Table IV) and subsequent clustering are an easily configurable and repeatable processes. The most time consuming process is the initial raw data extraction (Section III-C). Once those data are available, the group of selected strategy defining structures and units could be modified easily and then additional experiments could be carried out to try to find better group.

C. Strategy distribution

The results in Fig. 6 prove the variety of identified strategies for all three races is good. For each race, few favorite strategies are clearly visible as well as those less popular.

V. Dataset labeling

We label the original STARDATA replay files by adding the following information:

- Strategies for both players.
- Match-up (can be inferred from the strategies).
- Winner flag.

Example original file: *bwrep_0xi84.rep*.

Labeled file: *bwrep_0xi84_TZ7_ZT2W.rep*.

The original unique replay ID number is preserved. Player 1 was Terran and used strategy TZ7 (Fig. 4. Player 2 was Zerg and used strategy ZT2 (Fig. 5). The winner was Player 2, indicated by the symbol W after their strategy.

VI. Conclusion

We classify player strategies used in StarCraft: Brood War replay files from the largest known unlabeled dataset called STARDATA and label the replay files. The replay files in the labeled version now offer information about match-up and strategies used by both players and also identify the winning player. While original STARDATA may be used for unsupervised learning, in machine learning, it is always beneficial to have more options. Having the above information available makes the labeled version useful for supervised learning. We make the labeled dataset available for future machine learning attempts for StarCraft AI agent training and improvement.

References

[1] M. Buro, "Real-time strategy games: A new ai research challenge," *International Joint Conferences on Artificial Intelligence, IJCAI 2003,* pp. 1534-1535.

[2] Z. Lin, J. Gehring, V. Khalidov and G. Synnaeve, "STARDATA: A StarCraft AI Research Dataset," *13th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2017,* pp. 50–56, arXiv:1708.02139.

[3] S. Ontañon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill and M. Preuss, "A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft," *IEEE Transactions on Computational Intelligence and AI in games, IEEE Computational Intelligence Society, 2013,* 5(4), pp. 1–19, doi: 10.1109/TCIAIG.2013.2286295.

[4] Mi. Čertický, D. Churchill, K.-J. Kim, Ma. Čertický and R. Kelly, "StarCraft AI Competitions, Bots and Tournament Manager Software," *IEEE Transaction on Games, 2018,* 11(3), pp. 227–237, doi: 10.1109/TG.2018.2883499.

[5] O. Vinyals, I. Babuschkin et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature, 2019,* 575, pp. 350–354, doi: 10.1038/s41586-019-1724-z.

[6] B. G. Weber and M. Mateas, "A data mining approach to strategy prediction," *IEEE Symposium on Computational Intelligence and Games, 2009,* pp. 140-147, doi: 10.1109/CIG.2009.5286483.

[7] H. C. Cho, K. J. Kim and S. B. Cho, "Replay-based strategy prediction and build order adaptation for StarCraft AI bots," *IEEE Conference on Computational Intelligence in Games (CIG), 2013,* pp. 1-7, doi: 10.1109/CIG.2013.6633666.

[8] G. Synnaeve and P. Bessière, "A Dataset for StarCraft AI & an Example of Armies Clustering," *Artificial Intelligence in Adversarial Real-Time Games, 2012,* arXiv:1211.4552.

[9] W. Gong, X. Zhang, B. Deng and X. Xu, "Palmprint Recognition Based on Convolutional Neural Network-Alexnet," *Federated Conference on Computer Science and Information Systems, FedCSIS 2019,* 18, ACSIS, pp. 313–316, doi: 10.15439/2019F248.

[10] Š. Krištofík, P. Malík, M. Kasáš, Š. Neupauer, "StarCraft agent strategic training on a large human versus human game replay dataset," *Federated Conference on Computer Science and Information Systems, FedCSIS 2020,* 21, ACSIS, pp. 391–399, doi: 10.15439/2020F178.

[11] M. Świechowski, "Game AI Competitions: Motivation for the Imitation Game-Playing Competition," *Federated Conference on Computer Science and Information Systems, FedCSIS 2020,* 21, ACSIS, pp. 155–160, doi: 10.15439/2020F126.

		average structure order							average unit frequency					
strategy	count	Nexus (exp)	CyberneticsCore	Gateway	Forge (F)	Templar Archives	Stargate (S)	Robotics Facility (R)	Dragoon (D)	Zealot (Z)	High Templar (HT)	Dark Templar (DT)	Carrier (C)	brief description
PP0	2222	3,85	2,01	1,01	8,00	7,82	7,99	3,14	1,03	2,21	6,00	6,00	6,00	DZ, fast R, exp
PP1	891	6,16	2,04	1,00	5,10	3,29	7,99	7,15	2,15	2,50	5,93	1,91	6,00	DT rush, often DZ, late exp
PP2	269	6,27	7,01	1,26	2,14	7,97	7,99	7,97	5,99	1,15	6,00	6,00	6,00	Cannon rush, Z, late exp
PP3	759	4,06	2,40	1,07	3,86	3,88	7,99	6,52	2,28	2,18	2,51	3,52	6,00	DZ HT DT, exp
PP4	682	2,25	3,30	1,31	3,32	7,36	7,96	6,71	1,58	1,91	5,85	5,64	5,96	DZ few HT DT, fast exp
PP5	872	7,38	2,03	1,00	6,90	7,86	7,99	8,00	1,43	2,11	6,00	6,00	5,99	DZ, no exp
PP6	1030	8,00	2,00	1,00	8,00	7,93	8,00	3,00	1,08	2,22	6,00	5,99	6,00	DZ, fast R, no exp
PP7	530	7,38	8,00	1,06	8,00	8,00	8,00	8,00	6,00	1,02	6,00	6,00	6,00	Z rush, no exp
PP8	866	3,88	2,04	1,00	6,25	4,99	7,99	3,41	1,69	2,29	3,53	3,75	6,00	DZ, often HT DT, exp
PP9	1001	4,71	2,06	1,00	4,27	7,56	7,98	3,53	1,13	2,05	5,97	5,99	6,00	DZ, R, F, exp
PT0	546	3,33	2,03	1,02	7,68	7,89	7,80	4,22	1,01	6,00	5,99	6,00	5,90	D, R, fast exp
PT1	427	4,76	2,02	1,00	7,40	3,80	7,72	3,99	1,58	3,79	5,90	1,63	5,97	DT rush, often D, exp
PT2	1055	3,22	2,23	1,13	4,70	7,23	7,54	3,90	1,51	1,55	5,81	5,95	5,95	DZ, R, F, fast exp
PT3	424	8,00	2,00	1,00	7,87	7,93	7,97	3,01	1,09	3,15	6,00	6,00	5,99	D, often Z, fast R, no exp
PT4	706	3,00	2,21	1,15	7,28	5,89	5,88	3,92	1,52	1,66	5,58	5,98	5,45	DZ, few C, R, fast exp
PT5	581	4,95	2,09	1,03	6,93	3,24	7,26	7,93	2,03	3,33	5,89	1,64	5,93	DT rush, often DZ, exp
PT6	161	7,07	7,46	1,48	5,33	8,00	7,93	8,00	6,00	1,59	6,00	6,00	6,00	Z rush, F, late exp
PT7	900	3,43	2,18	1,10	6,63	4,49	6,73	4,39	1,88	1,54	5,51	2,79	5,97	DZ, DT, R, exp
PT8	1631	3,15	2,17	1,13	8,00	8,00	8,00	3,88	1,24	1,78	6,00	6,00	6,00	DZ, R, fast exp
PT9	366	8,00	2,07	1,00	7,50	7,68	7,98	8,00	1,39	3,26	6,00	6,00	6,00	D, often Z, no exp
PZ0	824	2,28	3,84	2,43	1,50	7,16	5,96	7,23	1,44	1,65	6,00	5,77	5,98	DZ, fast F, fast exp
PZ1	672	6,26	2,07	1,02	4,81	4,59	3,13	7,53	3,48	1,51	3,63	3,34	6,00	Z, often D HT DT, fast S, late exp
PZ2	2144	2,00	3,98	2,76	1,28	5,85	5,62	6,92	1,63	1,65	2,82	4,68	6,00	DZ HT DT, fast F, often S, fast exp
PZ3	586	3,37	7,94	4,11	1,37	8,00	8,00	8,00	6,00	4,14	6,00	6,00	6,00	Cannon rush, exp
PZ4	521	2,20	3,92	2,57	1,37	5,89	5,46	7,53	4,94	1,70	5,51	1,38	6,00	DT rush, Z, fast F, often S, fast exp
PZ5	785	7,04	2,09	1,04	5,66	7,92	5,63	5,50	2,00	1,59	5,98	5,99	5,99	DZ, often FSR, late exp
PZ6	644	7,78	7,84	1,12	5,73	8,00	8,00	8,00	6,00	1,05	6,00	6,00	6,00	Z rush, no exp
PZ7	1314	2,07	3,96	2,64	1,37	5,86	5,55	7,39	5,60	1,42	1,83	3,73	6,00	Z HT DT, fast F, often S, fast exp
PZ8	1133	2,40	3,88	2,40	1,50	7,37	5,62	7,60	6,00	1,03	6,00	6,00	5,98	Z, fast F, often S, fast exp
PZ9	388	5,79	2,14	1,03	3,94	3,63	7,97	6,93	3,37	1,34	3,74	3,65	6,00	late exp

Fig. 3. Protoss strategies. Structure order: 1=built as first, 8=never built. Unit frequency: 1=always created, 6=never created

		average structure order											average unit frequency							
strategy	count	Academy (A)	Armory	Command Center (exp)	Comsat Station	Control Tower	Engineering Bay	Factory	Machine Shop	Science Facility	Starport	Refinery	Marine (M)	Vulture (V)	Goliath (G)	Siege Tank (S)	Wraith (W)	Medic (E)	Firebat (F)	brief description
TT0	203	10,78	9,23	3,17	11,77	11,96	10,76	2,17	3,80	12,00	11,51	1,15	2,50	2,15	8,00	3,19	7,98	8,00	8,00	V, often MS, fast exp
TT1	357	5,79	4,94	3,45	7,55	10,92	11,54	2,19	3,76	11,76	9,44	1,19	3,77	3,00	2,81	2,09	7,13	8,00	7,96	S, often GVM, few W, fast exp
TT2	192	11,89	10,61	4,39	12,00	8,03	9,82	2,17	4,03	11,98	3,94	1,16	3,27	4,04	8,00	1,77	2,74	8,00	8,00	S, often WMV, no A, exp
TT3	60	10,85	12,00	10,35	11,12	12,00	12,00	12,00	12,00	12,00	12,00	6,45	2,50	8,00	8,00	8,00	8,00	7,30	7,92	M rush, few E, late exp
TT4	241	6,36	10,17	4,95	8,74	6,78	10,46	2,22	4,10	11,89	4,08	1,21	3,28	4,18	7,90	1,83	2,54	7,94	7,92	S, often WMV, exp
TT5	354	8,63	5,88	5,05	10,01	6,90	9,36	2,12	4,11	11,66	4,74	1,11	4,08	4,11	1,81	2,14	5,42	8,00	7,99	GS, often WMV, exp
TT6	211	6,26	5,00	4,43	8,11	11,27	6,45	2,09	3,50	11,83	10,09	1,08	3,67	3,63	2,31	2,02	7,33	7,98	7,97	GS, often MV, few W, exp
TT7	99	11,60	10,38	12,00	11,84	12,00	10,33	2,02	5,65	12,00	12,00	1,00	2,21	2,67	8,00	5,05	8,00	7,95	8,00	MV, often S, no exp
TT8	84	11,48	11,67	11,56	11,88	8,68	10,92	2,01	6,17	12,00	3,36	1,01	2,98	2,60	7,92	5,27	2,62	8,00	8,00	WMV, often S, no exp
TT9	167	11,67	4,25	6,17	12,00	11,98	9,26	2,04	4,46	12,00	10,95	1,03	3,47	3,54	1,59	3,62	7,84	8,00	8,00	G, often MVS, exp
TP0	124	11,35	12,00	10,57	12,00	11,94	11,85	2,19	11,08	12,00	11,93	1,09	1,32	5,48	8,00	8,00	8,00	7,90	7,85	M rush, often V, few F, late exp
TP1	1083	5,48	11,99	4,35	7,60	11,78	5,66	2,16	3,30	11,92	11,41	1,11	2,84	1,20	8,00	2,14	7,92	7,83	7,95	VS, often M, few E, exp
TP2	808	6,29	6,84	3,52	7,84	11,02	5,92	2,22	3,39	11,30	9,77	1,19	3,92	1,45	2,39	2,39	7,84	7,93	7,97	VGS, often M, fast exp
TP3	948	9,22	10,63	6,13	10,44	6,00	7,57	2,06	3,29	11,62	4,50	1,05	3,07	1,46	7,46	2,20	6,58	7,93	7,98	VS, often M, few GW, exp
TP4	993	11,90	11,18	3,81	12,00	12,00	4,68	2,16	3,30	11,99	11,76	1,11	2,47	2,78	7,97	1,95	7,96	8,00	8,00	SM, often V, fast exp
TP5	220	11,35	12,00	10,27	12,00	12,00	11,73	12,00	12,00	12,00	12,00	8,79	1,83	8,00	8,00	8,00	8,00	7,95	7,94	M rush, late exp
TP6	731	11,59	11,89	12,00	12,00	12,00	8,83	2,01	3,02	12,00	11,59	1,00	2,30	2,17	7,95	2,65	7,89	7,99	8,00	MVS, no exp
TP7	471	11,59	11,67	3,55	12,00	12,00	12,00	2,14	3,20	12,00	11,71	1,11	2,12	3,00	7,90	2,46	7,94	8,00	8,00	MS, often V, fast exp
TP8	296	3,66	11,74	5,18	5,63	11,64	6,85	4,19	5,94	11,81	11,20	1,41	1,18	7,11	7,95	3,59	7,91	3,58	6,74	M, often ES, few VF, exp
TP9	1123	5,96	7,26	3,71	7,73	11,59	5,51	2,18	3,31	11,74	10,68	1,13	3,01	1,21	8,00	1,96	7,87	7,89	7,96	VS, often M, fast exp
TZ0	222	10,44	12,00	1,05	12,00	12,00	11,04	11,55	12,00	12,00	12,00	7,06	1,66	7,97	8,00	8,00	8,00	8,00	7,70	M rush, few F, fast exp
TZ1	589	5,52	11,29	7,93	8,05	5,89	7,62	2,10	8,01	9,38	3,34	1,05	1,28	4,56	7,84	5,13	4,39	3,04	7,07	M, often EWSV, few Fexp
TZ2	1125	3,04	12,00	1,49	5,19	12,00	4,75	9,90	11,87	12,00	12,00	1,91	1,11	7,96	7,99	8,00	8,00	2,43	4,61	ME, often F, fast exp
TZ3	917	8,79	4,96	4,45	10,06	10,36	6,91	2,31	4,17	11,45	9,21	1,26	3,26	3,09	1,80	4,09	7,57	7,54	7,95	G, often MVS, few WE, exp
TZ4	506	11,96	12,00	12,00	12,00	12,00	11,65	12,00	12,00	12,00	12,00	10,72	1,21	8,00	8,00	8,00	8,00	8,00	8,00	M rush, no exp
TZ5	475	2,33	11,90	10,55	4,17	9,55	5,78	3,59	6,67	10,11	7,67	1,08	1,06	7,27	7,95	4,11	7,73	2,68	4,57	ME, often FS, few V, late exp
TZ6	374	1,97	11,98	12,00	5,86	12,00	7,64	10,94	12,00	12,00	12,00	1,16	1,17	7,90	7,98	8,00	8,00	3,18	4,22	M, often EF, no exp
TZ7	2005	3,18	11,88	1,30	4,80	9,44	4,20	5,75	7,39	10,01	7,77	1,94	1,06	7,11	7,95	2,70	7,70	2,83	5,42	M, often SEF, few V, fast exp
TZ8	390	10,86	10,12	10,33	11,91	8,52	9,85	2,04	7,84	11,54	5,24	1,03	2,28	2,88	6,82	6,67	4,29	8,00	7,99	M, often VW, few GS, late exp
TZ9	807	3,22	11,72	1,43	4,98	9,85	4,21	5,64	10,16	10,25	7,35	1,90	1,04	7,55	7,97	8,00	7,47	2,27	5,53	ME, often F, few VW, fast exp

Fig. 4. Terran strategies. Structure order: 1=built as first, 12=never built. Unit frequency: 1=always created, 8=never created

		average structure order					average unit frequency					
strategy	count	Hatchery (exp)	Lair (A)	Spawning Pool	Hydralisk Den	Spire	Zergling (Z)	Hydralisk (H)	Lurker (L)	Mutalisk (M)	Scourge (S)	brief description
ZZ0	816	5,22	2,00	1,00	6,00	3,00	1,32	6,00	6,00	2,64	2,20	Z, often MS, fast A, late exp
ZZ1	1007	1,72	2,93	1,35	5,99	4,88	1,03	6,00	6,00	6,00	6,00	Z rush, A, fast exp
ZZ2	1500	1,82	2,82	1,36	6,00	4,00	1,63	6,00	6,00	1,54	2,83	ZM, often S, A, fast exp
ZZ3	1579	3,97	2,00	1,00	6,00	3,16	2,66	5,99	6,00	1,82	1,54	MS, often Z, fast A, exp
ZZ4	2296	1,76	2,88	1,37	6,00	3,99	1,72	6,00	6,00	3,27	1,44	ZS, often M, A, fast exp
ZZ5	1092	5,23	2,00	1,00	6,00	3,00	1,47	6,00	5,99	1,55	6,00	ZM, fast A, late exp
ZZ6	808	2,14	6,00	1,54	5,96	6,00	1,15	6,00	6,00	6,00	6,00	Z rush, no A, fast exp
ZZ7	959	1,82	2,79	1,39	5,99	4,00	1,48	5,99	5,98	1,55	6,00	ZM, A, fast exp
ZZ8	141	1,67	4,45	1,53	3,39	5,67	2,12	1,21	5,84	5,60	5,72	H, often Z, few MS, late A, fast exp
ZZ9	498	5,95	2,00	1,00	5,99	3,80	1,02	6,00	6,00	6,00	6,00	Z rush, fast A, no exp
ZP0	788	1,71	6,00	1,30	2,99	6,00	1,89	1,14	6,00	6,00	6,00	ZH, no A, fast exp
ZP1	527	1,72	3,02	1,32	5,58	4,87	1,08	6,00	6,00	6,00	5,05	Z rush, few S, A, fast exp
ZP2	1090	1,72	3,21	1,30	4,41	4,35	2,78	1,36	5,75	2,11	4,64	HM, often Z, few S, A, fast exp
ZP3	864	1,72	3,06	1,29	4,68	4,25	2,27	2,35	2,25	5,30	3,59	ZHL, often S, few M, A, fast exp
ZP4	749	1,76	3,00	1,27	5,42	4,09	1,99	5,76	5,84	1,69	2,36	ZM, often S, A, fast exp
ZP5	920	1,74	3,02	1,27	4,85	4,13	2,50	1,14	5,83	5,32	2,47	H, often ZS, few M, A, fast exp
ZP6	826	1,73	3,41	1,33	3,56	5,68	2,12	1,92	2,16	5,82	5,93	ZHL, A, fast exp
ZP7	1416	2,34	6,00	1,26	5,89	6,00	1,13	6,00	6,00	6,00	6,00	Z rush, no A, fast exp
ZP8	571	1,84	2,96	1,30	5,59	4,04	1,70	5,92	5,81	1,32	5,99	ZM, A, fast exp
ZP9	1260	1,63	3,70	1,38	3,31	5,79	1,89	1,16	6,00	6,00	6,00	ZH, A, fast exp
ZT0	278	3,17	2,50	1,26	3,51	5,76	1,33	4,88	1,73	5,96	5,97	ZL, few H, fast A, exp
ZT1	1370	1,43	2,97	1,61	5,72	4,01	1,77	6,00	6,00	1,24	6,00	ZM, A, fast exp
ZT2	1027	1,31	3,00	1,71	4,82	4,15	2,02	4,07	1,61	2,72	5,75	LZ, often M, few H, A, fast exp
ZT3	443	1,45	3,98	1,61	3,44	5,73	1,90	1,24	5,99	6,00	5,82	ZH, A, fast exp
ZT4	790	1,27	3,09	1,74	4,53	4,36	2,78	1,53	5,88	1,93	5,44	HM, often Z, A, fast exp
ZT5	1086	1,49	4,69	1,54	5,77	5,56	1,18	6,00	6,00	6,00	6,00	Z rush, A, fast exp
ZT6	816	1,40	3,11	1,63	3,93	5,58	1,85	2,35	1,89	5,86	5,72	ZL, often H, A, fast exp
ZT7	451	1,42	2,98	1,64	5,47	4,13	2,16	5,61	5,86	1,97	2,11	M, often ZS, A, fast exp
ZT8	209	6,00	5,62	1,00	5,88	5,97	1,05	6,00	6,00	6,00	6,00	Z rush, no A, no exp
ZT9	940	1,24	3,01	1,78	4,68	4,30	1,43	4,93	2,30	4,32	2,53	Z, often LS, few HM, A, fast exp

Fig. 5. Zerg strategies. Structure order: 1=built as first, 6=never built. Unit frequency: 1=always created, 6=never created

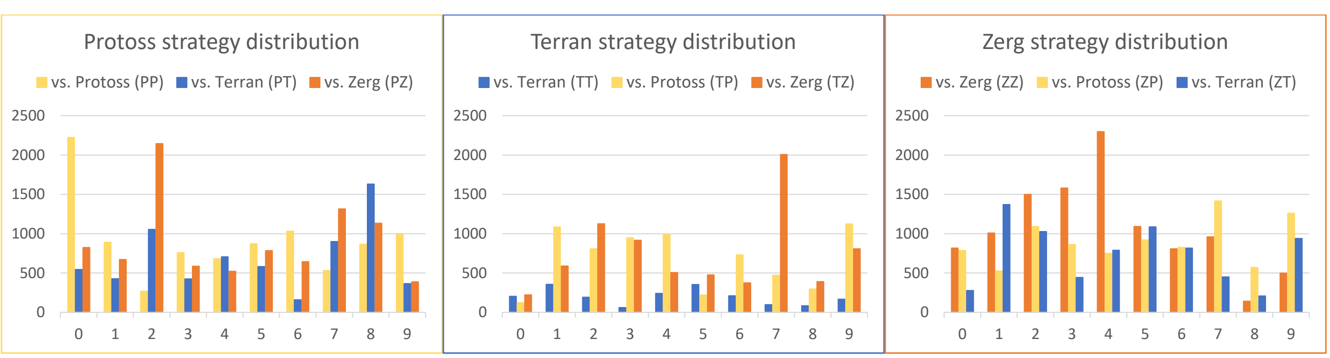

Fig. 6. Strategy distribution