Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation*

José Antonio Parejo Maestre[1,2,3][0000−0002−4708−4606] and Pablo Calle Pérez[3]

[1] Smart Computer systems Research and Engineering Lab (SCORE)
[2] the Research Institute of Informatics Engineering (I3US)
[3] Universidad de Sevilla, Spain
japarejo@us.es

Abstract. Many software design problems have been addressed using search-based approaches. However, software development usually involves domain-specific design decisions and problems, which are not so common in the SBSE literature. For instance, in the case of video-game development, game-play balancing is a crucial task that involves choosing parameter values for many of the mechanics of the game. This paper addresses the problem of automatically generating balanced decks for an online software implementation of the popular card-based game named Top Trumps using a multi-objective evolutionary algorithm. Specifically, three objective functions are defined, some of which involve the simulation of matches by confronting bots with different gaming strategies. The results of the experiments performed, show that the approach can generate decks without silver-bullet cards, for which better strategies usually lead to win, and for which games between players which similar strategies lead to close matches. Finally, an analysis of the impact of game parameters on the objective functions allowed the identification of interesting relationships and bounds on those parameters for a proper use of the algorithm and the objective functions.

1 Introduction

The digital entertainment industry, and specifically video-games, are the most popular choice for digital entertainment both in Spain and the world, exceeding by far the combined turnover of the audiovisual and musical sectors [5]. The creation of video-games involves the development of software through similar methodologies to those used for other types of software systems, and consequently many of its activities can be automated or aided by search based algorithms or machine learning techniques [6]. However, the development of such

* This work has been partially supported by the European Commission (FEDER) and Junta de Andalucia under projects APOLO (US-1264651) and EKIPMENT-PLUS (P18-1278 FR-2895), by the Spanish Government (FEDER/Ministerio de Ciencia e Innovación – Agencia Estatal de Investigación) under project HORATIO (RTI2018-101204-B-C21) and the Excellence Network SEBASENet 2.0 (RED2018-102472-T)

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

software also involves domain-specific activities that are an essential part of its design, such as adjusting the difficulty of the levels, balancing the characteristics and capabilities of the characters and the game-play mechanics, etc.

These choices are an essential part of user experience design, an important activity on the process of building such kind of software, and a crucial factor for its commercial success. If we were to modify the user interface of an application by means of a search algorithm to speed up its use and improve the user experience, no one would doubt that we are solving a software engineering problem. In this other case, the parameters that determine satisfaction with user experience are those of the game-play itself, and their choice is therefore also a software engineering problem. Software engineering is much more than the mere implementation and testing of code, it is also the elicitation of requirements, the adaptation of the software to its changes, and the search for user satisfaction through changes in the software configuration even when they don't know how to express their needs, and they have to evaluate design alternatives with a *like* or *dislike* when using the application, as A/B testing has shown.

This paper proposes the use of multi-objective evolutionary algorithms and simulation to generate balanced decks an online implementation of Top Trumps, a popular card-based game. However, the approach used in this paper can be applied to any card-based video-game were deck balance plays an important role on the game-play. The results of the experiments performed show that the proposed approach -based on NSGA-II and bots that simulate different gaming strategies- can generate decks that are balanced, and lead to fair and close matches (according to the definition of the corresponding objective functions defined and simulation results).

The remainder of the paper is structured as follows: Section 2 provides additional details about Top Trumps, the game which is subject of study in this paper. In Section 3 the related work is discussed, highlighting the differences with the previous approaches proposed in the literature. In Section 4 a formal definition of the problem at hand is provided, including the objective functions to be optimised, and the different strategies used by the bots in the simulations of games. Section 5 describes the adaption of the algorithm performed to solve the problem at hand, the parameter values used in its execution, and the implementation details. Section 6 describes the experiments performed, and provides answers to our research questions through the analysis of their results. Finally, Section 7 draws conclusions and describes future research directions of the proposals of this work.

2 Top trumps

Top Trumps is a card-based game, where each card contains a list of numerical data. The aim of the game is to compare these values to try to trump and win opponent's cards. A wide variety of different packs of Top Trumps has been published since 1978 based on different themes, such as cars, aircraft, books, boats, dinosaurs, or characters from a popular films or television series. For example,

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

cards may represent car models (example attributescategories: top speed, engine capacity, price) or football players (example attributescategories: height, years of international experience, caps). Figure 1 shows a card from the *Fantasy* pack: the Dragon.

Fig. 1: A Top Trumps card from the Fantasy pack: the Dragon[4]

In the original version of Top Trumps, the game is played with at least two players, and with a typical deck of 30 cards, up to 6 players can play. At the start of a game, the deck is shuffled and all cards are dealt among the players, each receiving an equal number of cards. The starting player draws a number of his topmost cards from his deck, and chooses one card and a category, reading its value. The other players draw their topmost cards from their decks (obviously the same number of cards than the staring player) and choose one. The player with the highest value on the chosen category wins all the cards played, that are placed at the bottom of the winner's deck. The cards drawn but not chosen by each player are placed at the bottom of their respective decks. The category of the next round is chosen by the winner of the previous round based on the values of his topmost cards and so on. The winner of the game is the player who collects all the cards, or the player with the maximum number of cards after a number of rounds (a parameter of the game that can be fixed at the beginning of the game to limit its duration). The number of topmost cards from which the players choose in each round is a parameter of the game.

3 Related work

Top Trumps has been used along with search-based algorithms with different purposes. In [1], Cardona et al. use an evolutionary algorithm and Open Data to generate cards for Top Trumps games. However, the focus of their work is

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

on the potential of the game to teach players, not in the balancing of the deck, and consequently the decks generated can have silver-bullet cards (cards with the bestworst values on all the categories), many dominated cards, etc. In [8], Top Trumps is used to demonstrate the feasibility of search-based techniques for game balancing. However, in both previous works, expert human intervention is required to evaluate the results of the algorithm and the fitness of the decks. In this work we propose the use of simulation and bots with different strategies to evaluate the fairness and closeness of the matches through two different objective functions, providing a fully automated approach.

Out of Top Trumps, search-based algorithms has been proposed as balancing mechanisms and strategy generators for the game-play of different kinds of games such as shooters[7] or real-time strategy [2]. Recently, other AI approaches such as deep learning, and reinforcement learning has also been proposed for the balancing of game parameters and the design of levels [9].

4 Problem definition

For the remainder of this paper, the number of cards in a deck is denoted as K and the number of characteristics (categories) displayed on a card L. A deck can be represented as a $K \times L$ matrix M, where the value of the k-th card in the l-th category is the element of the matrix $v_{k,l} \in R$, having $m \in \{1, ..., K\}$ and $l \in \{i, ..., L\}$. The values of the m-th card in the deck are therefore a tuple $v_m =< v_{m,1}, ..., v_{m,L} >$.

A partial order for the cards in a deck can be expressed as a dominance relation were, card c_1 dominates card c_2 if it has higher values for all the categories $\forall l \in \{1, ..., L\} \cdot v_{c_1,l} > v_{c_2,l}$. We assume here that the higher the value of the category the better.

4.1 Bot strategy definitions for game simulation

Random strategy This strategy randomly chooses the category for the round and the card played. This strategy is use as a bottom-line approach or straw man, to validate the fairness of the deck when we apply smarter strategies.

Locally Greedy strategy This strategy chooses the category $j \in \{1, ..., L\}$ with the maximum value in the set of cards available, i.e. given n cards $c_1, ..., c_n$ with values $< v_{c_1,1}, ..., v_{c_1,L}, ..., v_{c_n,1}, ..., v_{c_n,L} >$, the cathegory j is chosen such that $v_{c_x,j} \geq v_{c_y,m}, \forall m \in \{1, .., L\}, x, y \in \{1, ..., n\}$. The card to be played is chosen in a similar way to maximise the value on the chosen category.

4.2 Objective functions

In order to solve the problem of generating decks for Top Trumps, we adopt a multi-objective perspective and define three objective functions as follows:

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

Minimise the number of dominated cards The number of pairs of cards in the deck where one of them strictly dominates the other should be minimum. We denote the number of pairs of cards with strictly dominance as D_c.

Maximise match fairness For bots with a clearly better strategy than others, such as greedy and random, the results of the simulation should maximise the number of cards obtained by the best strategy during a number of games. This objective function is computed by confronting two sets of bots with different strategies in the simulation, and counting the number of cards owned at the end of each simulated game depending on its strategy. Let $B^d = \{b_1^d, .., b_N^d\}$ be set of N bots with a dummy strategy and $B^s = \{b_1^s, ..., b_M^s\}$ the set of M bots with a smarter strategy. Being $P(b, g)$ a function that provides the number of cards owned by bot b at the end of match g. Fairness for a set $G = \{g_1, ..., g_L\}$ of L games confronting a set of bots $B = B^d \cup B^s$ is computed as:

$$F(B, G) = \frac{\sum_{l=1}^{L}(\sum_{i=1}^{M} P(b_i^s, g_l) - \sum_{j=1}^{N} P(b_j^d, g_l))}{K \cdot L}$$

where K is the number of cards in the deck used for playing the games and L the number of games played among the bots. The image of $F(B, G)$ is $[-1, 1] \in R$ for any set of games and values of the problem parameters (number of players of each type, number of rounds per game, and number of cards). A value of 1 for $F(B, G)$ means that for all the games played, the smart bots owned all the cards at the end of the game. A value of -1 for $F(B, G)$ means that for all the games played, the dummy bots owned all the cards at the end of the game. Consequently, for a large number of games played if smart bots use a better strategy than dummy ones, and the deck is fair, $F(B, G)$ should provide positive values.

It is worth noting that the balance between deck size, number of players, and number of rounds per game impacts strongly in the maximum an minimum values that this function can take. For instance, for any number of games, if the number of rounds per game is 1 and the number of players/bots is 2 bot can gain or loss only one single card, as a consequence, the minimum value of this function is $-(2/K)$ (if the card is gained in all the games by a dummy bot) and the maximum value that this function can take is $(2/K)$ (smart bots win on all games). Hence, in order to use this function in a meaningful way, at least for games with P players/bots the number of rounds to be used in our experiments will be at least K/P. Hence, the minimum and maximum value of the function will be -1 and $+1$ respectively, and its value will accurately reflect the number of rounds won by each type of bot.

Maximise balanced matches closeness For bots with the same strategy, the results of the simulation should minimise the differences in the number of cards obtained for a number of games L. The aim of using this objective function is

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

ensure the generation of decks that lead to close matches, promoting vibrant games were players feel they can win even in the last rounds of the game. Let $B = \{b_1, ..., b_N\}$ a set of N bots with the same strategy and a set $G = \{g_1, ..., g_L\}$ of L games confronting those bots. This function is computed as follows:

$$C(B,G) = \frac{\sum_{i=1}^{N}(\sum_{b_j,b_k \in B \cdot j>k} |P(b_j,g_i) - P(b_k,g_i)|)}{K \cdot (N-1) \cdot L}$$

If we evaluate a set of identical games where the number of rounds is equal to the number of bots, and each bot wins one round, the value of this function would be 0, since all bots would have the same number of cards and all the values summed in the numerator of the fraction are 0. Those are clearly close and vibrant games. If we evaluate a set of identical games where the number of rounds R is equal to the number of cards divided by the number of players $R \geq K/P$, and there is a single bot in each that wins all rounds (that is a pretty boring game except for the winner), for each game we would sum K (the number or cards owned by the winner, whose maximum value is K, all the cards in the deck) $N-1$ times (the number of comparison among bots with cards). Thus, in such situation where one bot wins all the rounds and owns the entire deck at the end of the game, the value of the function would be 1. Those are games that are pretty boring except for the winner and are not close at all.

Any other set of games with different results would generate values of $C(B,G)$ between 0 and 1. As a consequence, to maximise the balance and closeness of the matches among players/bots with similar strategies, the objective is to minimise the value of $C(B,G)$.

5 Empirical study

5.1 Research Questions

(**RQ1**): Can the evolutionary algorithm generate suitable decks?, i.e., Can the algorithm generate decks that have no dominated cards ($D_c = 0$), for which most of games are fair ($F(B,G) \geq 0.4$ i.e. $> 70\%$ of games were fair) and where a majority of matches are close ($C(B,G) \leq 0.25$, i.e. the differences in number of cards is below a 25% of deck size in general, and most players can win)?

(**RQ2**): What is the impact of game parameters on the quality of the solutions provided by the algorithm?, specifically:

(a) What is the impact of the number of cards in the deck on D_c, $F(B,G)$, and $C(B,G)$?

(b) What is the impact of the number of categories on D_c, $F(B,G)$, and $C(B,G)$?

(c) What is the impact of the number of topmost cards used by the players on each round on D_c, $F(B,G)$, and $C(B,G)$?

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

(d) What is the impact of the number of rounds per game on D_c, $F(B,G)$, and $C(B,G)$?
(e) What is the impact of number of players/bots per simulation on D_c, $F(B,G)$, and $C(B,G)$?

5.2 Algorithm parameters

Given the rather exploratory nature of our research questions, we do not compare different algorithms for solving this problem, but a well known and performing algorithm for problems with three objectives such as NSGA-II [4] is applied, using the recommended parameter values shown in table 1.

Table 1: Parameters of the algorithms used in the experiment.

Parameter	**Value**
Population Size	100
Crossover Operator	SBX [3]
Crossover probability	100%
Mutation Operator	Polynomial mutation
Mutation probability	1.0 / Deck size
Selection operator	Binary Tournament Selection based on ranking and crowding distance [4]
Termination criterion	25000 objective function evaluations

5.3 Solution encoding and Operators

The algorithms applied in the experiments of this paper use a vector-based solution encoding. In this encoding, the deck represented as the $L \times M$ matrix M. Such matrix is encoded as a linear vector $< v_{1,1}, ..., v_{1,L}, ..., v_{K,1}, ..., v_{K,L} >$ of real numbers of size $L * M$. For instance, in a deck with fifty cards and five categories the length of the vector would be $50 * 5 = 250$. The first five values in the vector encode the values for the corresponding categories of the first card of the deck, values from the sixth to the tenth position encode the values of second card of the deck, and so on. Regarding evolutionary operators, we used Binary Tournament Selection (SBX) based on ranking and crowding distance as crossover operator [3], and Polynomial mutation as recommended by [4] for problems with the aforementioned encoding.

5.4 Implementations and conduction details

The problem definition, solving algorithm and experiments were implemented using Java 11 and the JMetal 5.12 framework run up to 4 parallel experiments using the multi-core processing. All the experiments were executed on Windows 11 using a laptop with a 11th Gen Intel i7-11800H processor at 2.30GHz, and 32 GB of RAM. Simulations evaluating fairness $F(B,G)$ use an even number of players with each strategy.

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

6 Experimental design and Results

6.1 Capability of the algorithm for generating high-quality decks (*RQ*1)

In order to provide a response to *RQ*1, we conducted 30 executions of the algorithm for generating decks with 30 cards and 6 categories (it is the typical number of cards and categories provided in top trumps packs). Regarding simulation parameters, 30 two-player games with 15 rounds each were played, and consequently all the cards in the deck were used in each game. Players can choose the three top-most cards of their respective decks. The bounds used for the categories' values were 0 as minimum and 100 as maximum. It is difficult to define specific bounds per category since the values usually depend on the semantics of the property expressed in the category, and in this case our categories are synthetic, but those are parameters that could be configured in the algorithm and provided by the user.

The descriptive statistics for each objective function are shown in Table 2.

Table 2: Summary of the values for each objective function of all the pareto fronts generated

	RQ1 (N = 459)
Closeness	
minimum	0.10
median (IQR)	0.15 (0.13, 0.17)
mean (sd)	0.15 ± 0.03
maximum	0.25
Fairness	
minimum	0.28
median (IQR)	0.48 (0.44, 0.51)
mean (sd)	0.47 ± 0.05
maximum	0.56
Dominated Cards	
minimum	0.00
median (IQR)	1.00 (0.00, 2.00)
mean (sd)	1.66 ± 2.41
maximum	21.00

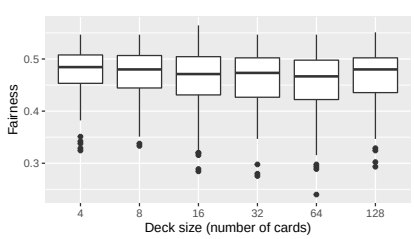
In all the pareto fronts generated for each execution we could find a solution that meets our goals (the bounds defined by *RQ*1 at Section 5.1). Consequently, *the algorithm can generate Top Trumps decks that are balanced and fair, and that promote close matches among players with similar strategies.*

The total execution time of the overall experiment was 25 minutes and 43 seconds, with an average execution time per run of individual experiment of 205.73 seconds (30 runs and 4 parallel execution threads).

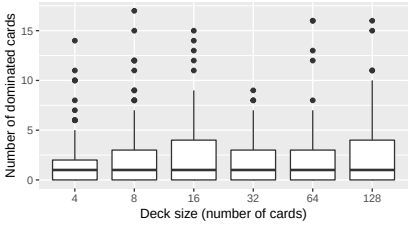
Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

6.2 Impact of game parameters on the quality of the decks generated (*RQ*2)

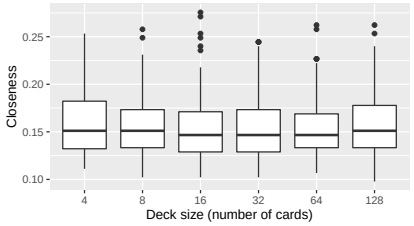
***RQ*2.*a* Impact of the number of cards per deck** In order to evaluate the impact of the number of cards per deck in the value of the objective functions, we executed the algorithm with the same parameters than in the first experiment but varying the number of cards in the deck K and the number of rounds R in each game, in order to ensure that all cards have the opportunity of being are played at least one time, by maintaining the ratio $R = K/P$, were P is the number of players in the game, that in the experiment is fixed to 2. We varied the number of cards using as values $\{4, 8, 16, 32, 64, 128\}$. For each value we run the algorithm 10 times and use all the solutions provided in the pareto fronts generated (without extracting the non-dominated solutions of the whole set) with a total execution time of 19 minutes and 32 seconds.



(a) Impact of deck size on games fairness



(b) Impact of deck size on the number of dominated cards

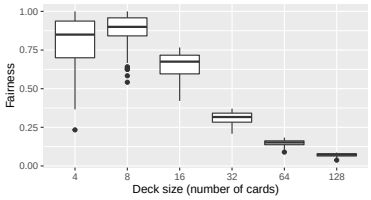


(c) Impact of deck size on games closeness

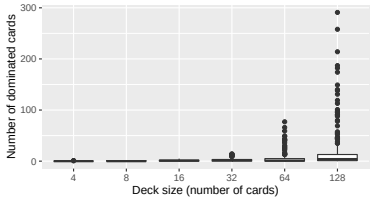
Fig. 2: Impact of deck size (number of cards) on the values of the objective functions in the pareto fronts generated by the algorithm

Figure 2 depicts the evolution of the distribution of values of each objective function in the pareto fronts as the number of cards varies. It shows that deck size seems to have no impact on any of the objective functions used as long as the ratio of rounds per player and deck size is maintained. However, if we use a constant number of rounds of 15 and repeat the experiment, the impact on fairness and closeness is dramatic, and Figure 3 reflects it.

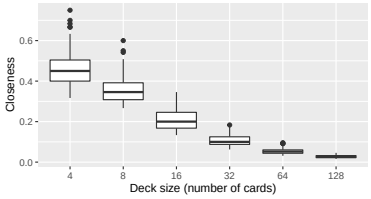
As the number of cards per deck increases for a fixed number of 15 rounds, fairness decrease and games become closer (the values of the objective function



(a) Impact of deck size on games fairness with a fixed number of rounds



(b) Impact of deck size on dominated cards (fixed number of rounds)



(c) Impact of deck size on games closeness with a fixed number of rounds

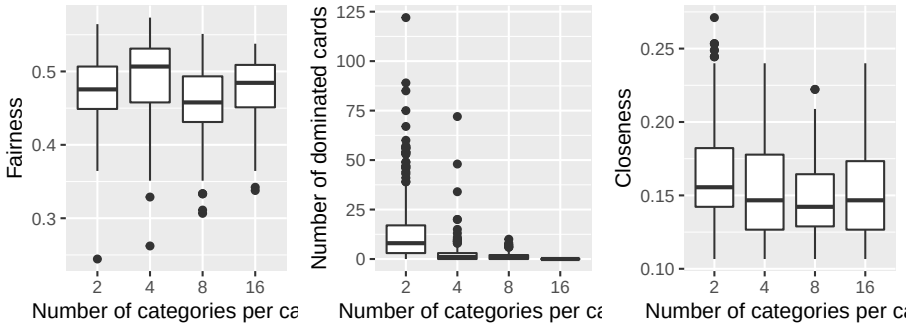
Fig. 3: Impact of deck size (number of cards) on the values of the objective functions in the pareto fronts generated by the algorithm with a fixed number rounds (15)

$C(B, G)$ decrease). This is caused by the remaining cards in the decks of players that have not been used due to the number of rounds.

***RQ2.b* Impact of the number of categories per card** In order to evaluate the impact of the number of categories per card on the value of the objective functions, we executed the algorithm with the same parameters than in the first experiment but varying the number of categories per card, using values $\{2, 4, 8, 16\}$. For each value we run the algorithm 10 times and use all the solutions provided in the pareto fronts generated. The total execution time of the experiment was 357 seconds. Figure 4 shows the evolution of the distribution of values of each objective function in the pareto fronts as the number of categories varies. The number of categories per card seems to have no impact neither on the fairness nor on the closeness of the games (no significant statistical differences were found). However, as the number of categories per card increases, it becomes easier for the algorithm to generate decks without dominated cards.

***RQ2.c* Impact of the number of topmost cards used per round** In order to evaluate the impact of the number of topmost card used per round on the value of the objective functions, we executed the algorithm with the same parameters used in the first experiment but varying the number of topmost cards used per round, among values $\{1, 2, 4, 8, 16, 30\}$. For each value we run the algorithm 10 times and use all the solutions provided in the pareto fronts generated. The

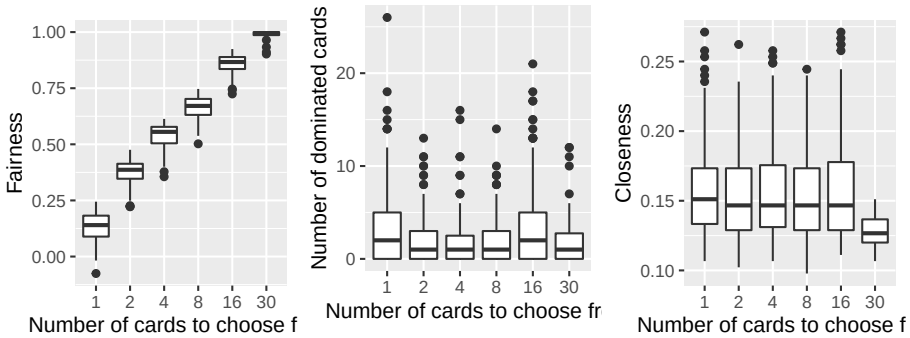
Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765



(a) Impact of #categories on fairness (b) Impact of #categories on dominated cards (c) Impact of categories on closeness

Fig. 4: Impact of the number of categories per card on the values of the objective functions in the pareto fronts generated by the algorithm

total execution time of the experiment was 422 seconds for $10 \cdot 6 = 60$ individual experiment runs. Figure 5 shows the evolution of the distribution of values of each objective function in the pareto fronts as the number of cards varies.



(a) Impact on fairness (b) Impact on dominated cards (c) Impact on closeness

Fig. 5: Impact of the number of topmost cards used per round on the values of the objective functions in the pareto fronts generated by the algorithm

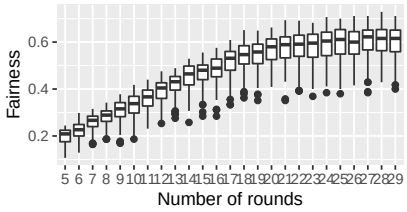
Figure 5a shows clearly that as the number of topmost cards used per round increases, the fairness fitness function becomes higher. This makes sense, since if the number of cards chosen is one, for a given category, the random dealing of cards at the start of the game usually decides the winner. However, as the number of cards used increases, it also increases the chances that bots with smarter strategies would find better values and win. The impact of the number

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

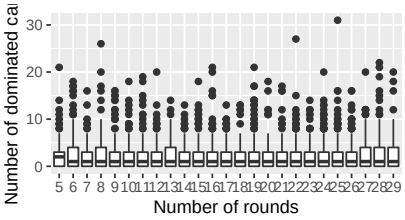
Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

of cards used seems to have no impact on the other objective function except for very extreme values in the case of closeness (as shown in Figures 5b and 5c).

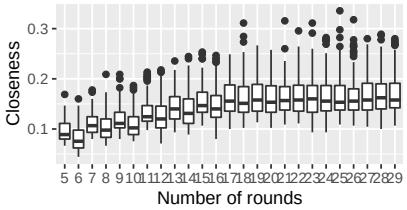
***RQ2.d* Impact of the number of rounds per game.** In order to evaluate the impact of the number of rounds per game, we executed the algorithm with the same parameters and in the initial experiment varying the number of rounds from 5 to 30 for two-player games. The total execution time of the experiment was 1952 seconds for $25 * 10 = 250$ individual algorithm runs. Figure 6 depicts the impact of the number of rounds played per game on the objective functions.



(a) Impact of rounds on games fairness



(b) Impact of rounds on the number of dominated cards



(c) Impact of rounds on games closeness

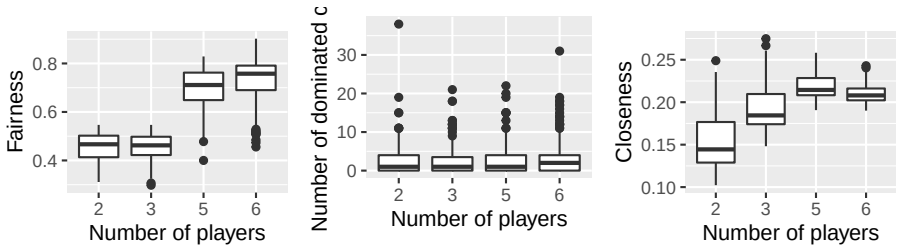
Fig. 6: Impact of number of rounds per game on the values of the objective functions in the pareto fronts generated by the algorithm

Clearly, the number of rounds per game has a strong impact on the fairness fitness function, see Figure 6a. As the number of rounds per game increases (up to a certain value), the value of fairness becomes higher. This makes sense, since if we play less than K/P rounds (where K is the number of cards in the deck, and P is the number of players in the game, players will end the game owning a minimum number of cards even if they lose all the games (not all cards are player), and consequently, fairness value will be low. After that threshold, the increase of fairness with the number of rounds becomes smaller, and for the interval between 20 and 29 rounds it becomes almost negligible. The number of rounds seems to have also a small negative impact on closeness (Figure 6c shows a positive impact on value of the objective function that the algorithm tries to minimise), which also makes sense, since for games with many rounds, it

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

is difficult that all players reach to the last round with possibilities of winning. The number of rounds has no impact on the number of dominated cards in the deck as shown in Figure 6b.

***RQ2.e* Impact of the number of players/bots simulated.** With a standard deck of 30 cards the maximum number of recommended players were 6, thus we executed the algorithm 10 times for 2,3,5 and 6 players (for distributing the cards evenly among players), running 30 games with 15 rounds per game. The execution time of the algorithm was 319 seconds for a total number of 40 algorithm runs. Figure 7 depicts the impact of the number of players on the objective functions.



(a) Impact on fairness (b) Impact on dominated cards (c) Impact on closeness

Fig. 7: Impact of the number of players on the values of the objective functions in the pareto fronts generated by the algorithm

The number of players impacts positively on fairness and negatively on closeness (it increases the value of closeness which should be minimised). This impact on fairness make sense, since as the number of players increase, the number of bots with smart strategies have more opportunities to find a card with higher values among their topmost cards. The impact on closeness is also sensible, since the higher the number of players in the game, the harder is to reach to the end of the game with possibilities for all those players (if one player has a sequence of bad rounds due to the values of the cards, the value of the objective function will increase due to the differences in the number of cards owned).

7 Conclusions

In this paper we have demonstrated the suitability of a fully automated approach based on multi-objective evolutionary algorithms to generate balanced Top Trumps decks that promote fair games (where players with clearly better strategies tend to win) with close results (there are not big differences on the results for players with similar gaming strategies). Three objective functions

Parejo Maestre, J. A., Calle Perez, P.: Creation of balanced Top Trumps decks with multi-objective evolutionary algorithms and simulation. In: Goñi, A. (ed.) Actas de las XXVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2022). Sistedes (2022). https://hdl.handle.net/11705/JISBD/2022/6765

have been defined and several empirical experiments has been performed. Furthermore, we have studied the effect of the different game parameters on the objective functions, and consequently, on the suitability of the approach. In so doing, we have identified a key relationship among problem parameters for a proper use of the fairness objective function (the number of players multiplied by the rounds should be greater or equal than the size of the deck), Additionally, we have also identified which parameters have a direct impact on each objective function for typical game settings. Regarding future work, we plan to integrate the algorithm in an online tool for the creation of downloadable decks, where users can provide category names and bounds, and a set of images, and download a deck in pdf format. Additionally, we plan to implement an online version of the game to carry out an empirical study, that will confront the results of this paper with the experience of human users using the decks generated by the algorithm for real-world online matches of Top Trumps.

Verifiability and reproductibility

For the sake of verifiability, our prototype implementation as well as all artifacts of the experiments are available at https://github.com/japarejo/Balanceo-de-cartas-mazo-Top-Trumps.

References

1. Cardona, A., Hansen, A., Togelius, J., Friberger, M.: Open trumps, a data game. In: Proceedings of the 9th International Conference on the Foundations of Digital Games (2014), ISBN: 978-0-9913982-2-5
2. Critch, L., Churchill, D.: Sneak-attacks in starcraft using influence maps with heuristic search. In: IEEE Conference on Games (CoG). pp. 1–8 (2021)
3. Deb, K., Agrawal, R.B., et al.: Simulated binary crossover for continuous search space. Complex systems **9**(2), 115–148 (1995)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
5. by Asociación Española de Empresas Productoras y Desarrolladoras de Videojuegos y Software de Entretenimiento, P.: Libro blanco del desarrollo español de videojuegos (2020), https://dev.org.es/es/publicaciones/libro-blanco-dev-2020
6. Harman, M., Mansouri, S.A., Zhang, Y.: Search-based software engineering: Trends, techniques and applications. ACM Computing Surveys (CSUR) **45**(1), 1–61 (2012)
7. Preuss, M., Pfeiffer, T., Volz, V., Pflanzl, N.: Integrated balancing of an RTS game: Case study and toolbox refinement. In: 2018 IEEE Conference on Computational Intelligence and Games, CIG. pp. 1–8 (2018)
8. Volz, V., Rudolph, G., Naujoks, B.: Demonstrating the feasibility of automatic game balancing. GECCO 2016 - Genetic and Evolutionary Computation Conference pp. 269–276 (7 2016). https://doi.org/10.1145/2908812.2908913
9. Volz, V., Schrum, J., Liu, J., Lucas, S.M., Smith, A.M., Risi, S.: Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In: Genetic and Evolutionary Computation Conference, GECCO 2018. pp. 221–228. ACM (2018). https://doi.org/10.1145/3205455.3205517