Soft computing for content generation: Trading market in a basketball management video game

Jose M. Peña, Ernestina Menasalvas, Santiago Muelas*, Antonio LaTorre *†
*Universidad Politécnica de Madrid, Spain
†Consejo Superior de Investigaciones, Spain
{jmpena,emenasalvas,smuelas,atorre}@fi.upm.es

Luis Peña, Sascha Ossowski ‡
‡ Universidad Rey Juan Carlos
Mostoles, Spain
{luis.pena,sascha.ossowski}@urjc.es

***Abstract*—Although procedural and assisted content generation have attracted a lot of attention in both academic and industrial research in video games, there are few cases in the literature in which they have been applied to sport management games. The on-line variants of these games produce a lot of information concerning how the users interact with each other in the game. This contribution presents the application of soft computing techniques in the context of content generation for an on-line massive basketball management simulation game (in particular in the virtual trading market of the game). This application is developed in two different directions: (1) a machine learning model to analyze the appeal of the trading market contents (the virtual basketball players in the game), and (2) an evolutionary algorithm to assist users in the design of new contents (training of virtual basketball players).**

I. Introduction

Modern video games, in particular multiplayer on-line titles, are founded on the existence of rich virtual environments for the players to be involved in. In this sense, the automatic, programmatic or assisted mechanisms to create new game contents are of great interest for this industry.

The concept of game contents differs from one game genre to the other. For instance, in the case of role-playing or adventure games, the virtual world and its inhabitant characters, as well as quests and plots, are the main game contents. On the other hand, the contents in car racing games are elements such as race tracks or adversary pilots. In the context of construction simulation games, the types of units, research paths and technological advances and the game maps are considered the most relevant contents.

This paper addresses the content generation issues in the context of sport simulation and management games. In such type of games, the human player takes the role of a team manager in charge of a basketball, football or hockey franchise and decides about the hiring of new players or the investment in the stadium facilities, for example. It is quite usual that the human player takes also coach decisions, selecting match tactics, as well as team lineups. In these games, the most relevant elements in terms of content enrichment is the whole set of available virtual players to be hired. The existence of virtual players with different characteristics opens the possibility to put together different team configurations, including alternative match tactics that would make the game experience thrilling and challenging.

In these games, the human player (referred as *manager*, from now on) hires the necessary virtual players (that will be referred in this document just as *players*) in a trading market, which is also part of the sport management game. The dynamics of this trading market, in which players are sold, contracted, transferred or exchanged, are an important part of the game, according to several basic conditions (quite usual in most of these types of games):

1) Hiring players requires an amount of resources (virtual money in the game world).
2) Managers have access to a limited amount of this virtual money. However, their management skills allow them to obtain more of this monetary resources, depending on team performance.
3) Better players (in terms of their match behavior, or characteristics) are more difficult to contract, more expensive to hire or demand higher salaries.
4) Players get old and eventually retire when they reach a given age, being replaced by younger players promoted by different mechanisms.

Sport management video games, such as Football Manager 2013 (Jacobson, 2012), Championship Manager (Beautiful Game Studios, 2009) or FIFA Manager 2013 (Bright Future 2012), belong to a quite popular game genre. Nowadays, several on-line games have appeared in the scenario of sport management simulators, for instance Red Zone Action[1], Broken Bat[2], MMA Tycoon[3] or Hattrick[4]. As in the case of other on-line genres, they are motivated by the more challenging gaming happening when different human users compete against each other. In such on-line games, together with human-human match competition, the managers interact also when they are involved in the trading market of players. Thus, the trading market is populated by the managers as well as by several automatic mechanisms.

The objective of this paper is to apply soft computing techniques (1) to analyze the trading market of a given basketball management game in which thousand of managers interact, in order to identify the game contents (players in this case) attractive for the market, providing the patterns to create these players, and (2) to assist the managers in the design of the training strategies to promote interesting and successful players.

The remainder of the paper is organized as follows:

[1]http://redzoneaction.org/football/
[2]http://www.brokenbat.org/
[3]http://www.mmatycoon.com/
[4]http://www.hattrick.org/

Section II presents the related work on content generation using different computational intelligence techniques. In Section III, we introduce the basket management simulation game and its trading market concepts. Then, Section IV presents the analysis of market trends to obtain appealing contents related to user's demand. Section V shows how evolutionary techniques can be applied to assist the users to generate their own contents. Finally, Section VI concludes this contribution with the principal remarks and open issues.

II. Related work

Game content generation has been one of the most active research topics in the application of computational intelligence techniques in video games [1]. This topic has been of great interest not only for the computational intelligence community but also as a clear example of computational creativity (as the computer takes the partial role of a game designer or artist) [2].

In addition, industry has shown its interest on these techniques as a way to reduce production costs and to speed-up game development and updating. These techniques applied to the automatic creation of game content is referred as *Procedural Content Generation* (PCG) [3]. The nature of these contents can be of different kinds, from *non-player characters* (NPC) to scenarios. This set includes aspects such as terrain, maps, levels, stories, dialogues, quests, characters, rule-sets, dynamics and game items.

Besides the construction of NPC AI controllers [4], [5], which is typically beyond of what is referred as PCG, the content generation applied to the production of game elements is related to aspects such as decorative components [6] or functional products that appear in a game. The second of these subsets, the functional elements of the game [7], the levels [8], [9], maps [10], terrains [11] or providing adaptability [12], is the one that we undertake in this paper, adapted to the nature of a specific game genre.

Moreover, PCG has also been an active application field of many different artificial and computational intelligence techniques, such as search methods [3] (with multiobjective optimization [10] or cellular automata [9]), machine learning [13], reinforcement learning [14], software agents [11] or hybrid approaches [5].

There are different reasons for the application of PCG to the game elements in the development and maintenance of the games:

1) First, to reduce the total space of the game, which is a hard constraint in some platforms. For instance, in the current developments for mobile platforms, the feature to produce contents that were not previously stored in the deployed installation of the game, applying compression or programmatic approaches, can save an important amount of space.
2) The second reason is the complexity to manually produce game contents in some developments. Many studios use automatic support and design tools to produce certain parts of the games, like levels or maps, for instance, automatically creating vegetation in a certain area. These tools reduce the production time in the development of massive open worlds which will need to setup large sets of this kind of content in their scenarios.
3) Moreover, the third reason to apply PCG to game development is the possibility of creating a completely new game content, even in real time, possibly producing endless games. The creation can also be parametrized or characterized by certain values. This way, we can identify particular users or play styles (intended as a way to play a game, the choices of certain tasks or quests against others, or any preferences of strategies) to create certain types of contents that are more appealing to these users' profiles, making the game more attractive to different types of players.

Figure 1. Character profile and skills

III. Trading market in BasketStars

In this article, we focus our study on the analysis of the trading market data obtained from a massive multiplayer on-line sport management simulator named BasketStars[5]. This game system has more than 300,000 registered on-line users, each one playing as a team manager of a basketball franchise. The game is organized into four different hierarchies of leagues, named Constellations. Each constellation has several divisions (named SuperStars, Stars, Planet, Comet and Asteroid) each with 16 different players competing in a two-leg regular season with play-offs.

Teams conform their rosters with different players, each of them having their own characteristics and skills (Figure 1). These characteristics, which have values between 0 and 100, are Speed, Jump, Stamina, Pass, Free-Throws, 2Points-Shots, 3Points-Shots, Dunk, Steal, Rebound, and Block; together with some other information, such as age, height, and weight. There are also four other major parameters: Experience (obtained from the number of minutes and the level of the league), Position (Point Guard, Shooting Guard, Small Forward, Power Forward and Center), Salary (per season) and their aggregated Value (VAL), which is derived from a weighted average of the skills and height (where weights depend on the player's position).

[5]http://www.basketstars.com

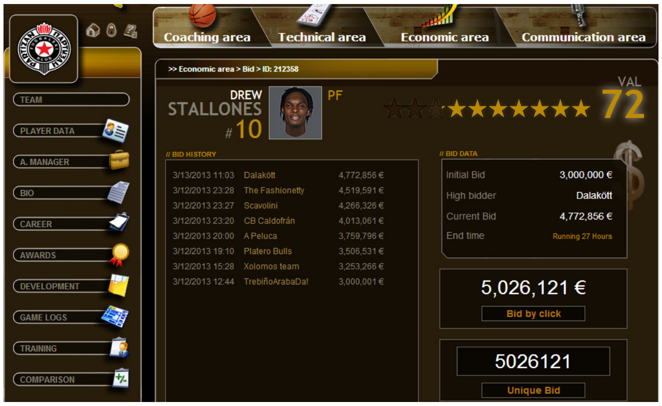

Figure 2. Auction bidding for a player (selling a player)

Figure 3. Auction bidding for a player (transferring a player)

In BasketStars, the teams provide themselves with players by three different means: (1) exchanging players with other teams, (2) training their own rookies, and (3) hiring players in a global trading market. The trading market is based on an auction bidding mechanism (Figure 2). The players are offered in the market for selling or for a temporal transfer. While the players that are sold are hired by a new team, those transferred are still linked with their original team but will play in a new team for one or two seasons. The auction biding system, used for selling players, starts when a team offers one of its players for a given amount of money (between 25% and 150% of the player's season salary). Then other teams bid for the player (multiple teams participate in the auction) rising the biding amount each time. In addition, players can be included in the trading market by the system itself instead of being offered by the different managers.

The auction mechanism for transferring players is slightly different. In this case the managers are not offering money for the player, but a commitment to make him play an average number of minutes per match and paying a percentage of his salary (see Figure 3). According to this, the player's owner decides to whom the player will be transferred to (considering not only their offers but also the division where the team is playing).

Moreover, the design of a successful team is restricted by different rules. Teams with enough money can purchase great players (with higher Value indicators and huge salaries); but they are always constrained by a Salary Cap depending on the division level. If the team exceeds in salaries this limit, it will be penalized, making it very expensive maintaining an all-top-star team. This makes that, in the trading market, the players with the highest Value (Salary is derived from this Value, the Experience and the Age) are not necessarily the most highly demanded. These constraints make the game very interesting and not monopolized by just a limited number of teams.

In such a massive game, the operations in the trading market generate a lot of information. This information can be used to analyze the dynamics of the game and to identify patterns for interesting players. The evaluation of the attractiveness for a player comes from the competition of many managers to hire a given player. Although the recruitment costs for such a player can be several times his salary, it is worthwhile comparing with a player cheaper to hire but much more expensive to maintain (due to the Salary Cap penalties).

A. *Data description and objective*

We have recorded 9,101 trading market operations (in two weeks between the end of one season and the starting of the new one). These records include information about the type of operation (Selling or Transferring), the player's characteristics, the number of bids, the starting one and the last one, and whether the player was transferred/sold.

The objective is to identify the patterns that describe interesting players (according to the market). Thus, the game engine could use the models defining these patterns to sample players with interesting profiles. This should be done carefully, as part of the appealing of the players is because some of the profiles are rare enough to be attractive and to justify the investment of a lot of money.

In addition, being able to model interesting players allows the game designers not only to introduce these players in the market, but also to identify the potential bias in the match simulations (for example if a higher rank in a given characteristic deeply influences the overall player's performance). Moreover, having also a model for inexpensive players with good performance allows also to assign one of these players to the starting managers in the game. This action improves potentially the involvement of the new managers in the game, increasing the fidelity of new users.

IV. Analysis of the Trading Market

The process to discover insights in the trading market data is composed of the following steps: (1) preprocessing data, (2) data set segmentation, and (3) data mining phase.

A. *Preprocessing step*

At this step new attributes are added to the trading market data in order to enrich the information. In particular we have derived the following attributes:

1) Salary: The managers from other teams cannot access to the player's salary, unless they use one of the features of the game (the *assistant manager*) that allows the manager to scout other team's players. But this feature can only be used a limited number of times. The only clue about player's salary is that the initial bet (the starting bid in the auction process) is a number between 25% and 150% of the salary. Thus, an automatic process to estimate this number is needed.

The description of the game mentions that the salary depends on the aggregated value (VAL), the age and the experience. As the game documentation mentions that a 0 value in experience is equivalent of having a 90% of the performance and 100 experience is equivalent to a 110% performance, a derived feature named adjusted value has being included. It is computed as the original VAL multiplied by a linear interpolation between 0.9 and 1.1 depending on the experience. In addition, the documentation indicates that the best moment in the player's career is when he has 30 years old, so we have derived two new attributes (years to be 30-yo[6], and years after being 30-yo[7], so they represent with positive values when the player is too young or too old from this "optimal" performance age).

Then, we have applied a prediction model to calculate an estimate of the salary of each player. Different methods have been applied to build an estimator of the salary (linear regression, neural nets, support vector machine (SVM), C&R and CHAID). The one showing better performance has been the Classification and Regression (C&R) Tree from SPSS (Clementine). The (C&R) Tree is a tree-based classification and prediction method [15]. Similar to C5.0, this method uses recursive partitioning to split the training records into segments with similar output field values. All splits are binary.

The application of the C&R algorithm focuses also the attention that the most important variable to calculate the salary of a player is the adjusted value and the new two age variables. The correlation value of the proposed method is 0.998 with a relative error of 0.004 (See Table I).

Table I. SALARY ESTIMATION USING ADJUSTED VAL, UNDER30 AND OVER30 VARIABLES

Model	Build time (mins)	Correlation	Relative error
C&R Tree	< 1	0.998	0.004
Neural net	< 1	0.953	0.096
CHAID	< 1	0.902	0.186
Linear Regression	< 1	0.771	0.406
SVM	< 1	0.552	1.073

2) Label attributes: The objective of the study is to identify interesting players in the market, which requires to define this label according to how the market (the other managers) has behaved when these players have appeared. Thus, we should define these labels according to the number of bets, or the difference between the estimated salary and the amount payed by the highest bet or something equivalent. Furthermore, this label attribute is the one we would like to predict according to the skills, age, height, salary and position.

[6] $under30 = IF(age < 30; 30 - age; 0)$
[7] $over30 = IF(age > 30; age - 30; 0)$

When a player auction is successfully finished the bidding process may have between 1 bet to more than 400 bets. These high numbers of bets are quite unusual: although they identify highly demanded players they only represent less than 1% of the auctions (with more than 100 bets). Moreover, there is always the possibility for an auction to finish with no bets, this means that the player stays in his current team.

In order to perform the corresponding analysis, we proposed three different labeling criteria:

- **sold**: If the player has received one or more bets (57% of the players (5,214)). This is the baseline characterization of the appeal of a given player (for the market perspective). A player who is not sold represents the lack of interest from other managers.
- **4bets**: Whether the player is transferred or sold after 4 or more bets (16% of the players (1,457)). As a second label, players, with multiple offers are considered more interesting than those with just one bet. The threshold value of 4 bets has been selected as the higher value for having a minimum of 15% of positive cases (for avoiding the case of an unbalanced classification problem).
- **good_deal**: As an indicator of the interest, there is also the possibility to consider the ratio between the final bet and the estimated salary. Although it is only valid for selling operations (not transferring), we have derived a new attribute that will establish a player to be a good deal for his original owner whenever the final bet price is more than twice his estimated salary (there are 6,104 selling operation 67% of the overall operations. 43% of them are labeled as *good_deal* (2,614 out of the 6,104)).

B. Data segmentation step

The players in BasketStars have five (5) possible positions, as mentioned before: Point Guard (PG), Shooting Guard (SG) Small Forward (SF), Power Forward (PF) and Center (C). There is a clear dependency between the interest of the player and the ideal skill values you expect for a given game position. Thus, we have segmented the dataset according to the position skill, in order to obtain an independent model for each player position.

C. Data mining step

For the 15 models calculated (5 positions times 3 goals/labels to predict) the C4.5 algorithm has been applied to calculate a predictive model [16]. The C4.5 algorithm has been chosen after a process in which this algorithm outperformed the other considered algorithms (naïve bayes, logistic regression, support vector machines, and random forests). Furthermore, the possibility to interpret the results makes it a perfect candidate for this kind of problems.

The models use all the skill variables (Speed (SPE), Jump (JUM), Stamina (STA), Pass (PAS), Free-Throws (FT), 2-Points-Shots (2P), 3-Points-Shots (3P), Dunk (DUN), Steal (STE), Rebound (REB), and Block (BLO)), experience (EXP), age (AGE), value (VAL), height (HEIGHT), weight

(WEIGHT) and estimated salary (SALARY). For the cases of *sold* and *4bets* labeled data we have also used the attribute specifying whether it is a selling or a transfer operation (TYPE). Table II shows the accuracy of the application of the C4.5 algorithm to the data sets as previously stated.

The league level is not included in the model, because the game implements an open market, which means that teams from different divisions may concur in the auction for the same player. A particular player could be interesting as a rotation member in a top-level team or as a key franchise player in a low division category. Thus, the team category is not unique for a given player when he is offered in the trading market.

Table II. C4.5 ACCURACY RESULTS FOR EACH OF THE LABELS AND POSITIONS

Position	PG	SG	SF	PF	C
sold	77,03%	76,24%	74,79%	71,87%	71,86%
4bets	86,18%	87,51%	88,06%	89,41%	88,85%
good_deal	79,42%	79,51%	74,93%	69,00%	65,44%

D. Discussion

First of all, we have seen that each model tries to identify two separate profiles. From one side, we have got inexpensive players, whose discriminant characteristics are difficult to identify. These profiles are only relevant in the *sold* label, being a minority in the *good_deal* label. These players are hired by the different teams just to meet one of the requirements of the game (to have a minimum number of players in their rosters) or as a complementary player to balance an exchange agreement between two teams.

A second part of the models, and the one mainly present in the *4bets* and *good_deal* labels, describes regular players (those that are enrolled to play a significant part of the matches). For those types of players, the selected attributes are highly dependent on the position, Table III shows which attributes are statistical different (higher/lower) in the example of the *4bets* model.

Table III. SELECTED ATTRIBUTES APPEARING IN THE 4BETS MODEL

Variable	PG	SG	SF	PF	C
Pctg	16%	17%	19%	15%	12%
SPE	-	-	-	-	-
JUM	-	-	-	-	-
STA	-	-	-	-	-
PAS	↑	-	-	-	-
FT	-	-	-	-	-
2P	-	-	-	-	-
3P	-	↑	-	-	-
DUN	-	-	↑	-	-
STE	↑	-	-	-	-
REB	-	-	-	-	-
BLO	-	-	-	-	-
EXP	-	-	-	↑	-
AGE	-	-	-	↑	-
HEIGHT	-	-	-	-	-
WEIGHT	-	-	-	-	-
VAL	↓	-	-	-	-

Pctg indicates the percentage of players with this position that received 4 bets or more (*4bets*=TRUE)
The symbol ↑ indicates that the positive labels statistical **higher** scores in the corresponding variable (Mann-Whitney test $p - value < 0.001$).
The symbol ↓ indicates that the positive labels statistical **lower** scores in the corresponding variable (Mann-Whitney test $p - value < 0.001$).

Although the table above presents the relationship between a single variable and the label, it is far from being complete, as the profile characteristics depend on multiple variables. For instance, in the case of PF models (one of the represented by Table II), neither 3-Points-Shot (3P) nor Rebound (REB) skills are discriminant by themselves, but in a deeper analysis of the models produced by the machine learning algorithm, it is possible to distinguish between two separate profiles: a long-range 3-pointer shooter and a more physique and defensive player. Indeed, the combined attribute max(3P,REB) is able to discriminate the *4bets* label in the case of PFs with a $p - value < 0.001$. This same circumstance (together with the low percentage of them receiving 4 or more bets) happens for the C models, there is no single attribute that discriminates from interesting/non-interesting players. In particular, in this position it is quite common that the managers hire either offensive or defensive players, with a complete different set of primary attributes.

Nevertheless, these variables demonstrate that, beyond the use of the these parameters by the simulation engine, the managers perceive that these skills are the most important for a good performing player.

Moreover, it is worthwhile to comment that the aggregated Value (VAL) of the player is not recurrently appearing and sometimes it appears as a negative condition (selecting players with VAL values lower than a given threshold). This can be interpreted under the constraints imposed by the Salary Cap limit, which deeply penalizes high-VAL players. This consideration complements another finding: it is very common to find out high threshold values of very precise skills. Both issues emphasize that specialized players with moderated VAL values are preferred over generalist players (with average good skills but no remarkable in any of them) and much more than top players which are difficult to be sold due to their high salaries. Thus, the strategy for having a good balanced team is to hire good performing players but not too expensive. The appropriate performance of players depends on their characteristics, or more precisely of a specific combination of these characteristics. Indeed, well balanced players with moderated salaries are much more attractive to complete a good team configuration in which there are salary limits for just two-three top (expensive) players.

The PCG process derived from these results could be implemented by the straightforward sampling of the obtained models. Although, this should be done under some limitations, in order to maintain the appropriate diversity in player profiles, the overall improvement in the quality (defined according to the insights obtained from the market analysis) would be beneficial for the game.

This final conclusion from the analysis would be of particular interest for the game designers, as it happens to be an undesired bias of the game dynamics. With the insights derived from this analysis it is clear that the salary negotiation should be reconsidered in order to make it not directly depending from the VAL parameter and the experience and approach it to something derived from the offer/demand balance.

Figure 4. Player training

V. Assisting user's content generation

As mentioned above, the teams conform their rosters by different mechanisms. The trading market is the most representative, but this market together with the exchange of players between two teams does not generate new game contents (new players, in our context). Trading or exchanging players only moves players from one team to another (either by money or by other players in return). But the players get older in the game (seasons in Basketstars take one month and a half in real time and after every season the players are one year older), and they will eventually retire, which makes players to disappear from the game. To compensate this, there is always the mechanism of creating new players (and the patterns previously identified would be an interesting hint to articulate this process), but it is not the only way to produce new players.

The managers always have the possibility to scout for new promising players for the future. These players are discovered when they are very young and prior to be promoted to the main team they stay for few seasons in a training camp. In this training camp the manager decides which characteristics will be trained, considering the present values of these characteristics plus the potential of improvement each player has (Figure 4). This training process consists in the allocation of an amount of training activities on each characteristic to make it improve. The total amount of these training activities is limited and the managers have to decide how to distribute them to design the desired player (see Figure 4). This process represents a collaborative mechanism to generate new game contents for the game, as these new players can be transferred, sold or exchanged with other teams in the future.

Therefore, we propose an assisted mechanism to suggest the managers how to design successful players, either interesting for their teams or for the market (being able to obtain good profit for selling or exchanging the player).

A. Training new players

The problem of training interesting players is an optimization problem by itself. The objective is to distribute the training activities in such a way that the final player produced by this mechanism would maximize the economical benefit of selling him.

In detail, the players improve their skills according to the number of training activities (managers have the pre-season and regular season periods to allocate one activity per day to each of their training rookies, up to 36 activities per season). Rookies can be trained for a maximum of 4 seasons, giving 144 training activities to be distributed along their training phase. Rookies improve their skills depending on the number of training activities per skill and the potential improvement (common for a group of skills). The improvement is not lineal, as the higher the improvement (from the original skill value) is, the more training activities are needed to get an extra point in that skill.

In order to assist in the training of new players, an evolutionary algorithm has been applied. The optimization process proposes different training plans (candidate distributions of 144 training activities among the different skills) and the resulting plans are evaluated according to the following fitness function:

$$f(p) = \text{salary} \times \sum_{m \in \text{Models}} \text{Apply}(m, p) \qquad (1)$$

$$\text{Apply}(m, p) = \begin{cases} 1 & \text{if model } m \text{ is true for player } p \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

$$\text{Models} = \{\text{sold}, \text{4bets}, \text{good\_deal}\} \qquad (3)$$

Where $salary$ is the estimated player salary (according to the C&R model shown in Section IV-A). $Models$ are the models obtained by the market data analysis (Section IV).

B. Results

The proposed evolutionary algorithm is a Genetic Algorithm with the parameters reported on Table IV. Several parameter values have been considered, but their influence in the final results were rather limited.

Table IV. Tested and selected evolutionary algorithm parameters

Algorithm	Genetic Algorithm
Crossover	**RoundedBlendCrossover** ($\alpha = 1.0$) TwoPointCrossover
Crossover Probability	75%, 90%, **100%**
Mutation	UniformMutator **RoundedNormalMutator** ($\mu = 0.0, \sigma = 10$)
Mutation Probability	1%, 5%, 10%, **15%**
Elitism (top N per Generation)	N=0, **N=1**, N=5
Population Size	25, **50**, 100
Selection Scheme	ProportionalSelection UniformSelection **TournamentSelector** ($n = 2$)
End Condition	1000 generations

In bold type fonts, we have indicated the parameter values finally used for reporting the results.

The coding scheme uses an integer vector representing each of the eleven (11) skills to be trained. The gene indicates the number of training activities for this skill (from 0 to 144). All those candidate solutions which sum more than 144 training activities (the limit for 4 seasons) are discarded (no repair scheme).

Table V. EVOLUTIONARY ALGORITHM PARAMETERS

	Ini	Pot	Tra	Fin	Ini	Pot	Tra	Fin
	Rookie 1				**Rookie 2**			
POS	PG				PF			
SPE	54	5	1	60	54	5	17	73
JUM	56	5	10	72	54	5	2	66
STA	51	5	5	63	60	5	1	66
PAS	54	4	18	72	28	5	19	48
FT	47	4-	12	59	37	5-	12	50
2P	44	4+	22	66	37	5	15	55
3P	39	4	12	55	3	5+	15	24
DUN	36	4	8	49	54	5	11	70
STE	63	5	34	87	39	5	7	53
REB	28	5+	6	43	54	5	20	74
BLO	29	5	16	47	55	5	25	77
EXP	20				18			
AGE	19				18			
HEIGHT	192				213			
WEIGHT	103				125			
Models	*sold, 4bets, good_deal*				*sold, 4bets, good_deal*			
	Rookie 3				**Rookie 4**			
POS	SG				SF			
SPE	61	5	13	78	49	5	23	70
JUM	62	5+	17	84	48	5-	6	58
STA	60	5	1	66	47	5	12	64
PAS	38	1	5	44	41	5+	4	54
FT	44	1-	16	52	53	5	16	71
2P	40	1	17	51	38	5	20	58
3P	26	1	9	34	53	5+	23	77
DUN	55	1	17	66	42	5	18	61
STE	54	3-	9	64	42	3	5	52
REB	34	3	16	49	29	3	6	40
BLO	31	3	24	49	32	3	11	46
EXP	17				24			
AGE	18				19			
HEIGHT	200				203			
WEIGHT	105				113			
Models	*sold, 4bets, good_deal*				*sold, 4bets*			
	Rookie 5				**Rookie 6**			
POS	C				C			
SPE	41	1	6	48	39	5	13	56
JUM	38	1	15	42	38	5	5	50
STA	51	1	2	51	51	5	0	51
PAS	28	1	7	35	29	2	11	41
FT	48	1	18	59	51	1	19	65
2P	46	1	17	57	47	2	8	57
3P	19	1	5	25	23	2	6	32
DUN	47	1+	19	55	47	2	11	59
STE	32	1	13	42	32	5	21	52
REB	58	1	25	70	59	5+	25	84
BLO	59	1	17	70	56	5	25	78
EXP	14				18			
AGE	18				18			
HEIGHT	208				213			
WEIGHT	116				125			
Models	*sold*				*sold, 4bets, good_deal*			

Ini — Initial value of the skill when he is picked up to start training.
Pot — Potential improvement rate (from 1: poor to 5: excellent). In addition - sign indicate a -50% in his actual value and + sign a +50% in the value.
Tra — Number of training activities proposed by the assisted mechanism.
Fin — Final skill value after the training period.

To test the scenario we have used 6 real drafted rookies provided by the game. Table V shows the rookies, their potential values and the proposed training scheme.

C. Discussion

The assisted mechanism produces a proposal of training plan for the new players. This plan is usually a trade-off between specialized players (produced by concentrating training activities in just one or two skills) and players with an overall good set of skills. The former used to be more appealing to the market, but the improvement obtained from concentrating the training in just one skill becomes very slow as the number of improvement points and getting larger (improving from the base skill value to the first 10 points is cheaper in training activities than improving from point 11 to 20).

Indeed, the training mechanism tries to balance between generalist high-value players and specialist market-attractive players. For example, in the case of Rookie 1 in Table V, the algorithm emphasized the Steal (STE) skill as it is an attractive profile for a Point Guard (PG) position, making the player a specialist in this matter; a similar case happens in the case of Rookie 6, in which a defensive profile is proposed, with good Rebound (REB) and Blocking (BLO) skills.

Another important aspect to highlight is that, in those cases in which the player's potential is rather weak (for instance in Rookie 5), the proposed algorithm is trying a more uniform distribution of the training activities. This is due to the impossibility to get the player very appealing (the example of Rookie 5 does not match neither *4bets* nor the *good_deal* profile). Nevertheless, the algorithm is able to make him good enough to be sold and then try to maximize the salary, which is directly related to the Value (VAL) (becoming then a more generalist player). In contrast, Rookie 3 is designed to become a physical Shooting Guard (SG), with good Speed (SPE) and Jump (JUM), and dedicating significant number of training activities to improve Rebound (REB) and Blocking (BLO) skills. The final numbers are not very high for an overall player, but it is above the average for this position, making him an interesting player because you have a guard high enough that can contribute in the team rebound. Additionally, as a result of the heuristic nature of the search mechanism, the proposed training schemes are not always the same. With the same initial value and potential, the algorithm has offered training plans with more Speed (SPE) and Steal (STE) instead of Jump (JUM) and Rebound (REB).

Moreover, we would like to highlight the effect of the trading market interest in the design of the players. If you look at Rookie 4, he has a great potential in two skills (a 5+ in both Pass (PAS) and 3-Points-Shot (3P)). If the algorithm was trying to maximize the salary (correlated with VAL) of the player, it would develop both of them equally and beyond the training of the other skills. This would be so, because both skills contribute the same for the Value of a Small Forward position (according to how this Value (VAL) parameter is computed by the game). Nevertheless, the models obtained from the market analysis show that having higher 3-Points-Shot (3P) skill is much more appealing than having a great Pass (PAS) value. Thus the training plan is biased towards this skill to make the player more attractive for the market.

As a final remark, the scenario presented here is a proof-of-concept. In this article, the objective is to present a first scenario of how soft computing techniques could be beneficial to improve content generation in this particular game genre. The optimization problem itself is not a tough one. However, a greedy approach would be easily stuck in a local optimum due to the multimodality and deceptive effect introduced by the combination of the estimated salary and the model multiplier. This behavior motivated the proposed heuristic optimization algorithm based on a soft computing technique. This optimization algorithm (the genetic algorithm in our case) does not represent a state-of-the-art approach to solve an optimization problem like this, but it provides a good solution in a reasonable time. Future improvements could be repair mechanisms for candidate solutions with more than 144 training activities or specialized crossover/mutation operators to avoid these malformed solutions to appear.

VI. CONCLUSIONS AND OPEN ISSUES

This paper has presented a massive on-line game based on basketball management simulation, named BasketStarts. This game incorporates a trading market system to allow different managers (game users) to exchange, transfer and sell virtual players (which represents the main asset in the game contents).

In this contribution, we have shown how different soft computing techniques can be applied to a game of this kind. Firstly, we proposed the use of machine learning and data mining approaches to analyze the most demanded profiles among the virtual players involved in the trading market. To do this, we analyzed more than 9000 on-line trading operations based on a multiplayer auction bidding. This analysis has shown interesting patterns in the demand of virtual players. These patterns can be use to sample new virtual players to populate the trading market in the game.

In addition, the analysis of the trading market also discovered some undesired bias in the game dynamics. A number of high-value players (representing the equivalent of an all-star basketball player) are difficult to sell due to their high salaries and the Salary Cap restriction. However, the game does not provide any mechanism to negotiate down the wages in order to harmonize the quality of the players and their salary. It looks unreasonable for a top player to prefer to be fired and not hired by any other team rather than reducing his salary.

A second part of the article proposed the use of another soft computing technique, evolutionary algorithms, to assist managers in the training of new players. The optimization approach searched to design new players, using the mechanisms available for each manager, trying to maximize the appealing of the trained players according to the aforementioned profiles. This assisted training tool maximized the market value (salary and attractiveness of the player), which is not necessarily the same objective to having a good player to keep for your team. The optimization objectives could be different and should take into account the current team roster to identify weaknesses and good complementary players and trying to minimize salaries.

As a summary, this paper has exemplified the application of different soft computing techniques in the field of content generation. Moreover, it is, as far as the authors are concerned, one of the few contributions in the use of content generation approaches in the genre of sport management video games. In addition, other multiplayer games offering an on-line market place, such as Diablo III (Wilson & Boyarsky, 2012) or several of the Zynga productions (e.g. Mafia Wars or Chef Ville) could potentially benefit from equivalent content analysis/generation techniques.

ACKNOWLEDGMENTS

This work was financed by the Spanish Ministry of Science (TIN2010-21289-C02-02) and supported by the Cajal Blue Brain Project. The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the Centro de Supercomputación y Visualización de Madrid (CeSViMa) and the Spanish Supercomputing Network. A. LaTorre gratefully acknowledges the support of the Spanish Ministry of Science and Innovation (MICINN) for its funding throughout the Juan de la Cierva program. L. Peña is supported in part by the Spanish Ministry of Science and Innovation through the projects AT (Grant CONSOLIDER CSD2007-0022, INGENIO 2010) and OVAMAH (Grant TIN2009-13839-C03-02).

REFERENCES

[1] J. Togelius and J. Schmidhuber, "Computational intelligence and game design," in *Proceedings of the PPSN Workshop on Computational Intelligence and Games.*, 2008.

[2] C. Browne and S. Colton, "On game design constraints and computational creativity," in *Proceedings of the IEEE Conference on Computational Intelligence and Games, 2012*, 2012.

[3] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 172–186, 2011.

[4] S. M. Lucas and G. Kendall, "Evolutionary computation and games," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 10–18, 2006.

[5] L. Peña, S. Ossowski, J. Peña, and S. Lucas, "Learning and evolving combat game controllers," in *IEEE Congress in Computational Intelligence in Games (CIG'12)*, 2012.

[6] J. Whitehead, "Towards procedural decorative ornamentation ingames," in *Proceedings of the FDG Workshop on Procedural Content Generation*, 2010.

[7] E. J. Hastings, R. K. Guha, and K. O. Stanley, "Evolving content in the galactic arms race video game," in *IEEE Congress in Computational Intelligence in Games (CIG'09)*. IEEE, 2009, pp. 241–248.

[8] N. Sorenson and P. Pasquier, "Towards a generic framework for automated video game level creation," in *Applications of Evolutionary Computation*. Springer, 2010, pp. 131–140.

[9] L. Johnson, G. N. Yannakakis, and J. Togelius, "Cellular automata for real-time generation of infinite cave levels," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2010, p. 10.

[10] J. Togelius, M. Preuss, and G. N. Yannakakis, "Towards multiobjective procedural map generation," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2010, p. 3.

[11] J. Doran and I. Parberry, "Controlled procedural terrain generation using software agents," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 111–119, 2010.

[12] R. Lopes and R. Bidarra, "Adaptivity challenges in games and simulations: a survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 2, pp. 85–99, 2011.

[13] L. Galway, D. Charles, and M. Black, "Machine learning in digital games: a survey," *Artificial Intelligence Review*, vol. 29, no. 2, pp. 123–161, 2008.

[14] K. E. Merrick and M. L. Maher, "Motivated reinforcement learning for adaptive characters in open-ended simulation games," in *Proceedings of the international conference on Advances in computer entertainment technology*. ACM, 2007, pp. 127–134.

[15] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge Univ. Press, 1996.

[16] J. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.