Applied Soft Computing 18 (2014) 1–11

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Solving the balance problem of massively multiplayer online role-playing games using coevolutionary programming

Haoyang Chen*, Yasukuni Mori, Ikuo Matsuba

Advanced Integration Science, Graduate School of Chiba University, Japan

ARTICLE INFO

Article history:
Received 15 December 2011
Received in revised form 20 October 2013
Accepted 6 January 2014
Available online 24 January 2014

Keywords:
Game balance
Game design
Coevolutionary programming
Cooperative coevolutionary algorithm
Probabilistic incremental program evolution
Multi-objective optimization

ABSTRACT

In massively multiplayer online role-playing games (MMORPGs), each race holds some attributes and skills. Each skill contains several abilities such as physical damage and hit rate. All those attributes and abilities are functions of the character's level, which are called Ability-Increasing Functions (AIFs). A well-balanced MMORPG is characterized by having a set of well-balanced AIFs. In this paper, we propose a coevolutionary design method, including integration with the modified probabilistic incremental program evolution (PIPE) and the cooperative coevolutionary algorithm (CCEA), to solve the balance problem of MMORPGs. Moreover, we construct a simplest turn-based game model and perform a series of experiments based on it. The results indicate that the proposed method is able to obtain a set of well-balanced AIFs more efficiently, compared with the simple genetic algorithm (SGA), the simulated annealing algorithm (SAA) and the hybrid discrete particle swarm optimization (HDPSO) algorithm. The results also show that the performance of PIPE has been significantly improved through the modification works.

1. Introduction

In recent years, massively multiplayer online role-playing games (MMORPGs) become more and more popular all over the world. The global market has risen above $12 billion in 2012 and is forecasted to reach $17.5 billion in 2015. A MMORPG is a multiplayer persistent virtual world which consists of several distinct races. Unlike the board games (such as Chess, Shogi, and Go), which are symmetrical systems, MMORPGs are regarded as asymmetrical since each race is designed to hold some unique features. For example: the wizard can perform spells, while the knight specializes in close fighting. Before starting the game, players should first create one character from a certain race, then play the game by either exploring the virtual world or interacting with the other players.

In MMORPGs, the concept of balance can be divided into two parts, the balance of Player-versus-Environment (PvE) and the balance of Player-versus-Player (PvP) [1]. In this paper, specifically, the term PvP refers to "1 versus 1" combat because it is the core of the MMORPG's combat systems. Generally, balance of PvE mainly means the difficulty control of the game, while balance of PvP, existing only in asymmetrical games, refers to the power balancing among the races. When designing a MMORPG, the power of a race will be represented by some attributes and a set of skills each of which consists of several abilities. In most cases, the attributes and the abilities must be non-negative integers, and they are functions of the character's level, called the Ability-Increasing Functions (AIFs). So the main task of balancing PvP is to construct a set of AIFs by which there are no dominating strategies in the game world and each race should have the same probability to win unless designers have some special design purposes.

Traditionally, game companies solve the balance problem by designer's talent and human testing. For the balance of PvE, such approach can handle very well, even though it is expensive in both time and resources. The challenges of balancing PvE come from the environment rather than other players, so there is only one aim of the tuning process, namely to satisfy the human side. That is why the traditional way succeeded. However, in the case of balancing PvP, the traditional method causes two problems. Firstly, since designers cannot ensure the inexistence of dominating strategies, testers have to take a lot of time to play the games and even human testers cannot explore all strategies. Secondly, once a dominating strategy is identified, all the AIFs need to be modified to ensure the game is in balance, which is too complicated for human testing to handle. For instance (refer to Fig. 1), supposing there are three races (A, B, C) in the game world, designers may modify B's AIFs to get a balanced relationship between A and B, then similarly modify C's

Abbreviations: AIF, Ability-Increasing Function; ACTR, Average Computation Time Ratio; NUR, Number of Uncontrollable Relationships.

* Corresponding author. Tel.: +81 080 4686 1413.
E-mail addresses: hust.chenhaoyang@gmail.com (H. Chen), yasukuni@faculty.chiba-u.jp (Y. Mori), matsuba@faculty.chiba-u.jp (I. Matsuba).

1568-4946/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2014.01.011

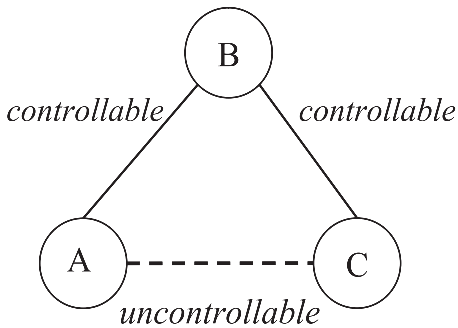

Fig. 1. There is one uncontrollable relationship in the game having three races.

AIFs to get the balanced relationship between B and C. After that, the relationship between C and A becomes determinate because all the AIFs have already been constructed, which means that the last relationship is uncontrollable.

Tuning game balance is a multi-objective optimization problem which will become hard to solve with the number of races increasing. The number of uncontrollable relationships, denoted by NUR, can be calculated by the following formula:

$$\mathrm{NUR} = \binom{n}{2} - n + 1 \tag{1}$$

where n is the number of races.

Moreover, since the nonlinearity of the objective functions, it can also be regarded as a nonlinear integer programming problem, which belongs to the class of NP-hard problems [2]. Therefore, there does not exist an exact algorithm, which can solve the problem in time, depending polynomially on the problem input data length or on the problem size. In this circumstance, heuristic algorithms, such as genetic algorithm (GA), become the only viable option because they find approximate solutions but have acceptable time and space complexity.

In the game industry, there are two kinds of revenue models, profiting from the game software and profiting from the game services. For most traditional games (such as console-based games, etc.), game companies employ the former one, that is, they earn money by selling the game software. However, in the case of MMO games, the latter one is preferred. The game software of MMO games are free to install, but players should subscribe to the game before they can play. In this revenue model, game companies make money by selling either the game time or the virtual items. Fig. 2 shows the sequence diagrams of the revenue models. In the left part, the yellow activation box refers to the life cycle of the game which means the total game time that the player spent in playing the game. But in the right part, the life cycle equals to the summation of all the yellow activation boxes. Since the free game software, it's obvious that the life cycle is much more crucial to the MMO games. If the players stop the subscription, the incomes of the game company will definitely decrease. Therefore, having a long life cycle has become the most important feature of the high-quality MMO games.

In the case of MMORPG, beside the user interface and the story line, the game balance also directly affects the life cycle of the game. For example, if we let a crab and a duck to play the rock-paper-scissors game, undoubtedly, the duck will quit the game as quickly as possible because it always loses. Generally, to be enjoyable, the game must be balanced well [3]. In other words, for the individual player, it must be neither too easy nor too hard, and for players competing against each other, fairness is crucial. A game without balance is untidy and ugly, flaws are reflected in the experience of playing it. An unbalanced game is less satisfying. More seriously, from the designer's viewpoint, not only time but also effort has been wasted. If parts of the game are not well-balanced, there will be a risk that some of them will be used rarely, if at all, in which case the time spent developing them has gone to waste.

Leigh et al. [4] have presented a solution for balancing a two-player, real-time action game called CaST. They use the competitive coevolutionary algorithm to search the dominating strategies and once found, tune the game rules and parameters. This solution highlights game imbalance as well as provides intuition towards balancing the game. It can be applied to on-line RPGs with some improvements. Alternatively, Olsen [5] has described the outline of an automated testing framework for MMORPGs. In such framework, game designers firstly create all AIFs empirically. Then the in-game combat data, recorded in game log, is used to construct a set of decision-making models by which a set of complex AI systems can be built. Based on those AIs, combats are simulated to verify whether one race is superior to another. If so, the AIFs will be continually tuned, until the game is in balance. Moreover, the decision-making models are refined periodically and after each refinement, the combat simulations as well as the tuning tasks should be redone (refer to Fig. 3). This approach can be regarded

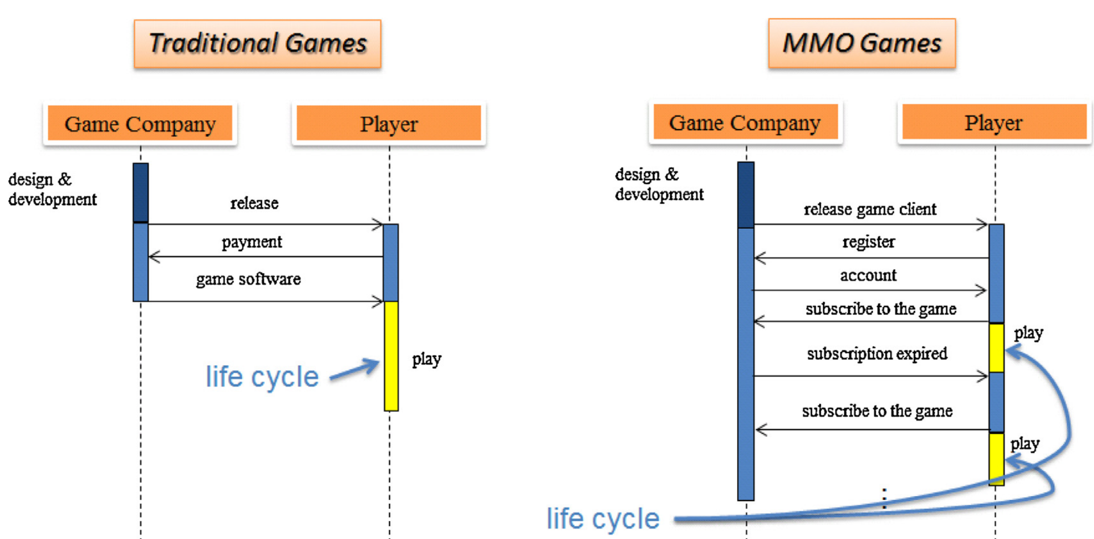

Fig. 2. Sequence diagrams of the two revenue models of the game industry.