Leveling-Up in Heroes of Might and Magic III

Dimitrios I. Diochnos[†]

Department of Mathematics, Statistics, and Computer Science,
University of Illinois at Chicago, Chicago IL 60607, USA
diochnos(at)math.uic.edu

Abstract

We propose a model for level-ups in Heroes of Might and Magic III, and give an $\mathcal{O}\left(\frac{1}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)\right)$ learning algorithm to estimate the probabilities of secondary skills induced by any policy in the end of the leveling-up process. We develop software and test our model in an experiment. The correlation coefficient between theory and practice is greater than 0.99. The experiment also indicates that the process responsible for the randomization that takes place on level-ups generates only a few different pseudo-random sequences. This might allow exploitation techniques in the near future; hence that process might require reengineering.

Key words: learning, reverse engineering, inverse coupon collector's problem, software reengineering, Heroes of Might and Magic

1 Introduction

Heroes of Might and Magic III (`HoMM3`) is a turn-based strategy and role-playing video game. It was developed by New World Computing for Microsoft Windows and was released by the 3DO Company in 1999. The game has been popular since its release and there is a big community worldwide. One of the major complaints of the players since the release of the game is that the *manual* was incomplete; in some cases facts were omitted, in other cases the phrasing was vague, and sometimes the descriptions were simply wrong[1]. In 2003 3DO went bankrupt, the rights of the game were sold to Ubisoft, and unfortunately, there has never been an update on the manual or answers to questions about mechanisms of the game. Typically players in the online community devise techniques which aim to uncover certain mechanisms, usually through excessive testing.

This paper has similar flavor. However, we want to minimize time-consuming human testing with the aid of algorithmic techniques. Section 2 has a brief description of the game, some fundamental definitions, and the two major problems related to this paper. Section 3 gives a model regarding a (fundamental from the players' perspective) mechanism of the game. Section 4 presents a Monte Carlo approach on learning efficiently the probabilities of certain attributes under that model. Section 5 has an experiment with a dual impact. First, we examine how close theory and practice are. Second, we derive quantitative estimates on the number of certain pseudo-random sequences generated by the actual game using the inverse version of the coupon

[†]Research partially supported by NSF Grant CCF 0916708.

[1]For example, http://heroescommunity.com/viewthread.php3?TID=17267 has a collection of more than 250 such examples.

primary skill	value
ATTACK	4
DEFENSE	0
POWER	1
KNOWLEDGE	1

slot	secondary skill	expertise
1	OFFENSE	BASIC
2	ARTILLERY	BASIC
3		
4		
5		
6		
7		
8		

Figure 1: Skills at level 1 for Gurnisson; hero class: Barbarian (mighty hero).

collector's problem. Section 6 gives ideas about future work and extensions to our current open-source software[2].

Finally, note that a team of enthusiasts, since 2007, is reengineering the game under the title *Tournament Edition*. Hence, the content of the paper has independent interest since a process of the game that generates randomness might require reengineering for a more balanced game.

2 A Brief Description of the Game and Related Problems

The game allows from 2 up to 8 players to take part in the game, possibly forming allied teams. Each player rules a kingdom that belongs to one of nine different factions; not necessarily different. Each kingdom is composed primarily by different cities and armies. The goal for each player (or team of players) is to eliminate all the opponents. The army can be split into different parts and each part is guided by some *hero*. There are two classes of heroes per faction, which we call *mighty* and *magic* for reasons that will soon be apparent. Hence, we have 18 different hero classes.

Heroes have abilities, called *primary* and *secondary skills*, that mainly reinforce the battles or help in the exploration of uncharted territory. Through victories in battles heroes acquire *experience*. As more experience is accumulated and certain values are surpassed, *heroes gain levels*. This *leveling-up* process typically enhances both the primary and the secondary skills, which, in principle, results in a stronger overall army.

There are four different primary skills; ATTACK, DEFENSE, POWER, and KNOWLEDGE. Mighty heroes develop their ATTACK and DEFENSE with higher probability, while magic heroes develop their POWER and KNOWLEDGE (which are associated with magic spells) with higher probability. Moreover, there are 28 secondary skills, and each hero can acquire and store in different *slots* at most 8 during each game. Secondary skills have 3 different levels of *expertise*: BASIC $<$ ADVANCED $<$ EXPERT which are obtained in that order. Typically heroes start with two BASIC secondary skills or one ADVANCED secondary skill, and some low non-negative integer values on the primary skills. We focus on mighty heroes of these two kinds only. We examine the different heroes between the starting level 1 and level 23; at level 23 the heroes have 8 EXPERT secondary skills for the first time. Figure 1 gives an example of the starting configuration for one popular hero of the game that starts with two secondary skills at BASIC level.

[2]See http://www.math.uic.edu/~diochnos/software/games/homm3/index.php

ATTACK $+1$	
ADVANCED OFFENSE	BASIC EARTH MAGIC

Figure 2: Sample level-up dialogue when Gurnisson (see Fig. 1) reaches level 2. The LEFT option is ADVANCED OFFENSE; the RIGHT is a new secondary skill (BASIC EARTH MAGIC).

Every time a hero gains a level, some primary skill is incremented by one; moreover, the user is presented with two secondary skills among which he has to choose one. We refer to the presented options as LEFT and RIGHT *option* since they appear respectively on the left and right part of the user's screen. Figure 2 gives an example of the dialogue that is shown on the user's screen during a sample level-up. The details that determine the LEFT and RIGHT option are given in Sect. 3.

Definition 2.1 (Level-Up)**.** Level-Up is the process that determines the pair (primary skill, (LEFT secondary skill, RIGHT secondary skill)) which is presented to the user when some hero gains a new level.

Figure 2 implies the pair (ATTACK, (OFFENSE, EARTH MAGIC)). The expertise is omitted for simplicity; it is straightforward to be computed. The pair which is presented on every level-up is called *level-up offer*. An *action* $\alpha \in \mathcal{A} = \{\texttt{LEFT}, \texttt{RIGHT}\}$ determines which secondary skill is selected on a level-up. A *state* $\kappa \in \mathcal{K}$ on a particular level for a particular hero contains the history of all the level-up offers up to this level, as well as the actions that were performed on every level.

Definition 2.2 (Policy [6])**.** A policy π is a mapping $\pi(\kappa, \alpha)$ from states $\kappa \in \mathcal{K}$ to probabilities of selecting each possible action $\alpha \in \mathcal{A}$.

A policy is called *deterministic* if there is a unique $\alpha \in \mathcal{A}$ with $\pi(\kappa, \alpha) = 1$ for every $\kappa \in \mathcal{K}$. Otherwise, the policy is called *stochastic*.

Clearly, not all secondary skills have the same importance; different secondary skills enhance different abilities of the heroes. Since the release of the game there are two main problems that have tantalized the players.

Prediction Problem: The first problem has to do with the *prediction* of the offered skills during level-ups. We present a model in Sect. 3 and we focus on the secondary skills.

Evaluation Problem: The second problem has to do with the *computation* of the probabilities of acquiring secondary skills by level 23 given the policy the players are bound to follow; see Sect. 4.

3 A Model for the Prediction Problem

We are now ready to examine a model for the level-ups as this has been formed by observations and testing throughout the years. A crucial ingredient is the existence of integer weights associated with the secondary skills. These weights can be found in the file HCTRAITS.TXT. On every level-up the model first determines the LEFT option and then the RIGHT.

3.1 The Basic Mechanism on Secondary Skills

Case A: The hero has at least one free slot. We have two subcases.

1. At least one of the secondary skills the hero currently has is not `EXPERT`. On the next level-up the hero will be offered an upgrade of one of the existing secondary skills as the `LEFT` option, while the `RIGHT` option will be a secondary skill the hero does not already have.

2. All the secondary skills the hero currently has are `EXPERT`. On the next level-up the hero will be offered two new secondary skills.

Case B: The hero does not have a free slot. We have three subcases.

1. The hero has at least two secondary skills not `EXPERT`. On the next level-up the hero will be offered two different choices in order to upgrade one of the secondary skills that are not `EXPERT`.

2. The hero has only one secondary skill not `EXPERT`. On the next level-up the hero will be offered only this secondary skill upgraded.

3. All 8 slots of the hero are occupied by secondary skills at `EXPERT` level. No secondary skills will be offered on level-ups from now on.

3.2 Presenting Secondary Skills on Level-Ups at Random

It is unclear who discovered first the data of the file `HCTRAITS.TXT` and how. However, these *weights* appear in forums and various pages about the game for many years now. The interpretation is that the weights are directly related to the probability of acquiring a specific (`LEFT`, `RIGHT`) secondary skill offer during a level-up. In particular, consider a set $\mathcal{S}$ of secondary skills, and to each $s \in \mathcal{S}$ we have a weight w_s associated with it. We say that a secondary skill s *is presented at random from the set* $\mathcal{S}$ and we imply that s is selected with probability

$$\mathbf{Pr}\left(\text{selecting } s\right) = \frac{w_s}{\sum_{s' \in \mathcal{S}} w_{s'}}\,. \tag{1}$$

We implement (1) the usual way; i.e. a pseudo-random number generated on run-time is reduced mod $\sum_{s' \in \mathcal{S}} w_{s'}$, and then an ordering on secondary skills determines which $s \in \mathcal{S}$ is selected. In principle we have two sets of secondary skills that we are interested in; the set $\mathcal{A}$ of secondary skills the hero already has but are not `EXPERT`, and the the set $\mathcal{U}$ of the secondary skills that the hero does not have in any of his slots.

3.3 Two Groups of Secondary Skills Appear Periodically

Two groups of secondary skills appear periodically and hence the randomized scheme presented in Sect. 3.2 is not always applied on level-ups; the limitations of Sect. 3.1 are always applied. These two groups are:

- The *Wisdom group* which is composed by one secondary skill; `WISDOM`.

- The *Magic Schools group* which is composed by the secondary skills AIR MAGIC, EARTH MAGIC, FIRE MAGIC, and WATER MAGIC.

Let $\mathsf{T}_{\text{Wisdom}}$ be the period for the Wisdom group, and $\mathsf{T}_{\text{Magic}}$ be the period for the Magic Schools group. Hence, every at most $\mathsf{T}_{\text{Wisdom}}$ level-ups, if the hero does not have EXPERT WISDOM, WISDOM will be offered; as BASIC if WISDOM does not appear in one of the slots (and clearly there is at least one empty slot), otherwise as an upgrade of the current expertise. Similarly, every at most $\mathsf{T}_{\text{Magic}}$ level-ups a secondary skill from the Magic Schools group has to appear. We refer to these events respectively as *Wisdom Exception* and *Magic School Exception*; i.e. exceptions to the randomized scheme of Sect. 3.2. When these two exceptions coincide on a particular level-up, then Wisdom is treated first; if necessary, Magic School Exception propagates to the next level-up. Hence it might take $\mathsf{T}_{\text{Magic}} + 1$ level-ups until Magic School Exception is applied; read below.

The model first determines the LEFT option and then the RIGHT option. Hence, the model first attempts to apply the Wisdom Exception on the LEFT option, and if this is impossible (e.g. Case A1 of Sect. 3.1 but the hero does not have WISDOM in any of the slots) then the model attempts to apply the Magic School Exception on the LEFT option (which might be impossible again). If the above two steps do not yield a solution, then the randomized scheme of Sect. 3.2 determines the LEFT option. The model then works in the same fashion in order to determine the RIGHT option. Note that each exception can be applied in at most one of the options on every level-up. For mighty heroes it holds $\mathsf{T}_{\text{Wisdom}} = 6$ and $\mathsf{T}_{\text{Magic}} = 4$.

3.4 The Leveling-Up Algorithm

Algorithm 1 gives the overall prediction scheme by incorporating the descriptions of Sects. 3.1, 3.2, and 3.3. There are four functions of primary interest during the level-ups since they handle randomness. RndNew returns a secondary skill at random from the set U. RndNewMagic returns a secondary skill at random from the set of Magic Schools that the hero does not already possess. RndUpgrade returns an upgrade of a secondary skill at random among the skills the hero has but not at EXPERT level. RndUpgradeMagic returns an upgrade of a Magic School secondary skill at random. Clearly, if there are two calls on the same function on a level-up, then, the skill that appears due to the first call is excluded from the appropriate set in the computations of the second call.

The function WisdomException returns TRUE if at least $\mathsf{T}_{\text{Wisdom}}$ level-ups have passed since the last WISDOM offer and the hero does not have EXPERT WISDOM. Similarly, MagicException returns TRUE if at least $\mathsf{T}_{\text{Magic}}$ level-ups have passed since the last offer of a Magic School and the hero does not have all the Magic Schools (with non-zero weight) at EXPERT level. In any other case these two functions return FALSE. HasWisdomToUpgrade returns TRUE if the hero has WISDOM in one of the slots but not EXPERT, otherwise FALSE. Similarly, HasMagicToUpgrade returns TRUE if the hero has at least one Magic School not EXPERT in one of the slots, otherwise FALSE. HasWisdom returns TRUE if the hero has WISDOM in one of the slots, otherwise FALSE. CanAcquireMoreSkills returns TRUE if the hero has at least one empty slot, otherwise FALSE. MagicSkillsAreAvailable returns TRUE if there are Magic Skills with nonzero weight that the hero does not already possess, otherwise FALSE. AllMagicAreExpert returns TRUE if all the Magic Schools the hero has are EXPERT, otherwise FALSE. HasFreeSlots returns TRUE if the hero has at least 1 slot empty, otherwise FALSE. Finally, NumberOfSkillsToUpgrade returns the number of secondary skills that occupy one of the hero's slots but are not EXPERT.

Algorithm 1: Determine Skills on a Level-Up.

```
Input: Appropriate amount of experience points to gain a level.
Output: A level-up offer.
level ← level + 1;
primary ← GetPrimarySkill ();
if AllSecondarySkillsExpert () then
    if not HasFreeSlots () then return (primary, (NULL, NULL));
    if HasWisdom () then
        if MagicException () and MagicSkillsAreAvailable () then
            LEFT ← RndNewMagic ();
        else LEFT ← RndNew ();
        RIGHT ← RndNew ();
    else
        if WisdomException () then
            LEFT ← WISDOM;
            if MagicException () and MagicSkillsAreAvailable () then
                RIGHT ← RndNewMagic ();
            else RIGHT ← RndNew ();
        else
            if MagicException () and MagicSkillsAreAvailable () then
                LEFT ← RndNewMagic ();
            else LEFT ← RndNew ();
            RIGHT ← RndNew ();
else
    if WisdomException () and HasWisdomToUpgrade () then
        LEFT ← WISDOM;
    else if MagicException () and HasMagicToUpgrade () then
        LEFT ← RndUpgradeMagic ();
    else LEFT ← RndUpgrade ();
    if CanAcquireMoreSkills () then
        if WisdomException () and not HasWisdom () then
            RIGHT ← WISDOM;
        else if MagicException () and MagicSkillsAreAvailable () and
        (AllMagicAreExpert () or WisdomException () ) then
            RIGHT ← RndNewMagic ();
        else RIGHT ← RndNew ();
    else if NumberOfSkillsToUpgrade () > 1 then
        if WisdomException () and MagicException () and
        HasMagicToUpgrade () then
            RIGHT ← RndUpgradeMagic ();
        else RIGHT ← RndUpgrade ();
    else RIGHT ← NULL;
return (primary, (LEFT, RIGHT));
```

4 Evaluating Policies

We resort to a Monte Carlo approach (Theorem 4.3) so that we can compute efficiently the probabilities of acquiring secondary skills by level 23 given any policy with bounded error and high confidence.

Proposition 4.1 (Union Bound). *Let* $Y_1, Y_2, \dots, Y_S$ *be* S *events in a probability space. Then* $\mathbf{Pr}\left(\bigcup_{j=1}^{S} Y_j\right) \leqslant \sum_{j=1}^{S} \mathbf{Pr}\left(Y_j\right)$.

Proposition 4.2 (Hoeffding Bound [3]). *Let* $X_1, \dots, X_R$ *be* R *independent random variables, each taking values in the range* $\mathfrak{I} = [\alpha, \beta]$*. Let* μ *denote the mean of their expectations. Then* $\mathbf{Pr}\left(\left|\frac{1}{R}\sum_{i=1}^{R} X_i - \mu\right| \geqslant \epsilon\right) \leqslant e^{-2R\epsilon^2/(\beta-\alpha)^2}$.

Theorem 4.3 (Monte Carlo Evaluation). *We can compute the probabilities of secondary skills induced by any policy* π *using* $\mathcal{O}\left(\frac{1}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)\right)$ *simulation runs such that the aggregate error on the computed probabilities is at most* ε *with probability* $1-\delta$.

Proof. Let $X_i^{(j)}$ be the indicator random variable that is 1 if the secondary skill j appears in the i-th run while following a policy π, and 0 otherwise. After R simulation runs, any skill j has been observed with empirical probability $\widetilde{p_j} = \frac{1}{R}\sum_{i=1}^{R} X_i^{(j)}$. We apply Proposition 4.2 to each $\widetilde{p_j}$ with $\alpha = 0, \beta = 1, \epsilon = \varepsilon/28$, we require to bound the quantity from above by $\delta/28$, and solve for R. We get $R \geqslant \left\lceil \frac{28^2}{2\cdot\varepsilon^2}\ln\left(\frac{28}{\delta}\right)\right\rceil$. Let Y_j be the event that $\widetilde{p_j}$ is not within $\varepsilon/28$ of its true value μ_j. The above analysis implies $\mathbf{Pr}\left(Y_j\right) \leqslant \delta/28$ for every skill j. None of these bad events Y_j will take place with probability $1 - \mathbf{Pr}\left(\bigcup_{j=1}^{28} Y_j\right)$. By Proposition 4.1 this quantity is at least $1 - \sum_{j=1}^{28}(\delta/28) = 1 - \delta$. We now sum the errors of all the 28 computed probabilities. □

5 An Experimental Study

In 2006 a player named `Xarfax111` suggested that the number of different sequences of secondary skill offers is actually fairly limited. An experimental study was conducted in order to verify the validity of the claim as well as test the effectiveness of the model.

The experiment consisted of 200 tests with Crag-Hack (mighty hero, class: Barbarian). Crag-Hack starts with ADVANCED OFFENSE but has zero weight on two secondary skills (one of which is a skill from the Magic Schools group) and hence these two can not be obtained. During the first 7 level-ups the new secondary skill that appeared was picked in every case, thereby filling all 8 slots of the hero by level 8; hence, all 8 secondary skills that appear in level 23 are determined by level 8. We call this policy AR (*Always Right*) since the user selects the (new) secondary skill that appears on the RIGHT of every level-up offer. Later in this section we examine estimates on the amount of different RIGHT sequences generated by the game; all imply expected values of at most 256. Note that the model allows more. For example, if we consider the cases where a Magic School appears every 4 levels and WISDOM every 6 levels due to exceptions (see Sect. 3.3), and allow only very heavy skills (with weights equal to 7 or 8) in between, there are $7\cdot6\cdot3\cdot5\cdot1\cdot4\cdot3 = 7,560$ different ordered combinations of new secondary skills until level 8. A similar calculation allowing all possible skills in between gives $7,325,640$ different combinations.

Out of the 200 tests, only 128 yielded different sequences of new secondary skills; 76 sequences occurred once, 33 sequences occurred twice, 18 sequences occurred three times, and 1 sequence

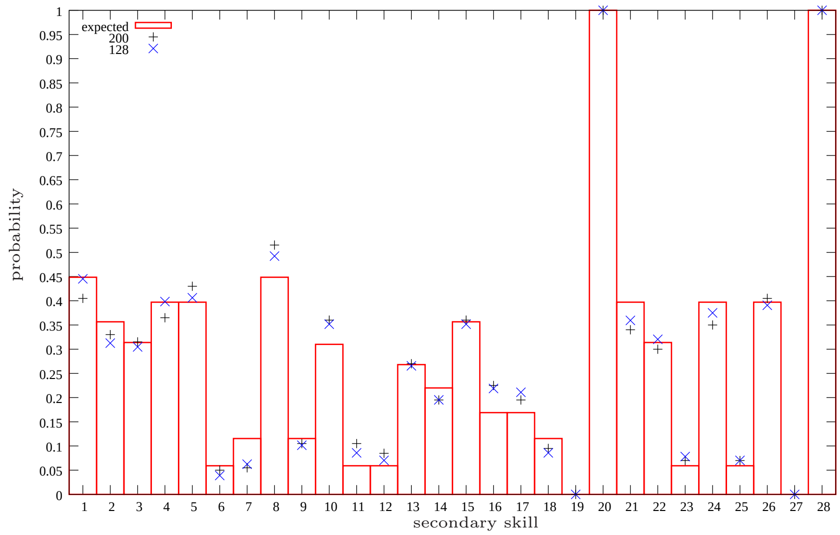

Figure 3: Probabilities of secondary skills; Crag-Hack, AR policy. The boxes indicate the expected values based on Sect. 3, the +'s present the values computed on all 200 tests, while the ×'s present the values computed on the 128 different sequences. The secondary skills are shown in lexicographic ordering; i.e. `1: AIR MAGIC`, `2: ARCHERY`, etc. Note that `28: WISDOM` and in every test `WISDOM` was offered by at most level 6.

occurred four times. Figure 3 presents graphically the probabilities of the secondary skills in three cases; the expectations according to the model, the values as these were recorded on the 200 tests, and the values when we consider only the 128 distinct sequences. The correlation coefficient between the expected probabilities and the ones computed empirically after 200 tests was 0.9914. Moreover, the correlation coefficient between the expected probabilities and the ones formed by the 128 unique sequences was 0.9946.

We now turn our attention to the second part of the experiment. We want to estimate the amount of different sequences of new secondary skills that appear when we follow this policy up to level 8 based on our observations on the collisions of the various sequences. This problem is essentially the inverse version of the Coupon Collector's Problem, and is well studied in Statistics; see e.g. [1, 2]. We will follow the simple route of working with expectations, assume equal selection probability for each coupon, and in the end we will arrive to the same formula for prediction as in [1]; see also [5, Sect. 3.6]. Let H_N be the N-th harmonic number. The expected time T_i to find the first i different coupons is given by

$$T_i = \sum_{j=N-i+1}^{N} \frac{N}{j} = \sum_{j=1}^{N} \frac{N}{j} - \sum_{j=1}^{N-i} \frac{N}{j} = N(H_N - H_{N-i}) \,. \qquad (2)$$

Lemma 5.1. *Let* $N \geqslant 3$ *and* k *be fixed such that* $k \in \{2, 3, \ldots, N-1\}$. *Then, the quantity* $Q(N) = N\,(H_N - H_{N-k})$ *is monotone decreasing.*

Proof. We want $Q(N) > Q(N+1)$ or equivalently $N\left(-\frac{1}{N+1} + \frac{1}{N+1-k}\right) > H_{N+1} - H_{N+1-k}$. However, $H_{N+1} - H_{N+1-k} = \sum_{i=N+2-k}^{N+1} \frac{1}{i} < \frac{k}{N+2-k}$. It suffices to show $\frac{kN}{(N+1)(N+1-k)} > \frac{k}{N+2-k}$, which holds since $k > 1$. □

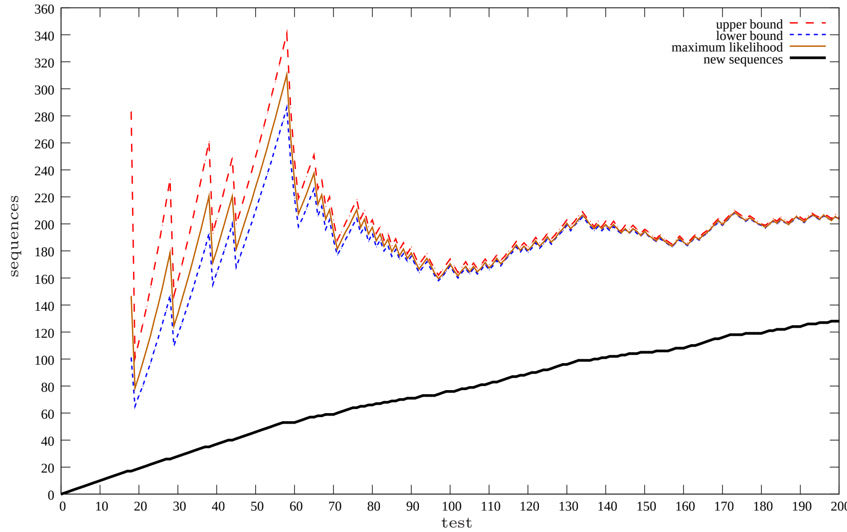

Figure 4: New sequences and estimates on the amount of different sequences.

The history of the 128 different sequences of new secondary skills is shown with a thick solid line in Fig. 4. Let D_i be the number of different new secondary skill combinations that have occurred on the i-th test. Working only with expectations we want to use (2), set $T_i = D_i$, and solve for N. This is precisely the solution asserted by the maximum likelihood principle. Typically, this is a floating point number; both the floor and the ceiling of that number are candidates for the solution; see [1]. We draw the average of those candidates with a thin solid line in Fig. 4. In order to get a better picture we apply Lemma 5.1 and calculate all the values of N such that $T_i \in [D_i - 0.5, D_i + 0.5)$; note that T_i rounded to the closest integer is equal to D_i. We get a lower and upper bound on the above mentioned values of N and we plot them with dashed lines in Fig. 4.

Another heuristic estimate can be obtained by looking at the ratio

$$\lambda(x,t) = \lambda_x(t) = \frac{\text{new sequences found in the last } x \text{ tests}}{x} . \tag{3}$$

We use the ratios $\lambda_x(200)$ for $x = 10, 20, 40$, and 80 as heuristic approximations of the true probability of discovering a new sequence at test $t = 200$. All four of them lie in the interval $[0.4, 0.5]$ which implies an estimate of $213 \leqslant N \leqslant 256$ different sequences in total.

6 A Glimpse Beyond

All the estimates of Sect. 5 are interesting since they are at most 256; a single byte can encode all of them. Quite recently (2009), a player named `AlexSpl` has developed similar software[3] and according to the descriptions we have[4] there are 255 different cases to be evaluated; there is a description of the random number generator too. Our model and `AlexSpl`'s description differ in the function RndUpgradeMagic of Sect. 3.4 where `AlexSpl` suggests treating all the

[3] http://heroescommunity.com/viewthread.php3?TID=27610

[4] http://heroescommunity.com/viewthread.php3?TID=17812&pagenumber=12

participating skills with weights equal to 1. Compared to our 0.9914 and 0.9946 values for the correlation coefficient in the experiment of Sect. 5, AlexSpl's approach achieves 0.9959 and 0.9974 respectively. AlexSpl's software is not open-source, and there is no description about his method on attacking the problem. In any case, we embrace relevant software; at the very least it promotes more robust software for everyone. Both Sect. 5 and AlexSpl's description imply a small space to be explored in practice. This suggests techniques of exploitation that will allow us to predict most, if not all, sequences online after a few level-ups, at least for a few popular heroes and policies followed in tournaments. Unfortunately, this will greatly reduce the fun and the luck factor of the real game! This is why reengineering is needed.

Coming back to our approach, perhaps the most important thing is to extend the current implementations and compute probabilities for magic heroes too. Another important extension is the computation of the probabilities for all the intermediate levels. Moreover, there are policies closer to tournament play which have not been implemented yet. In addition, we would like to ask questions such as *what is the probability of acquiring* (TACTICS $\wedge$ AIR MAGIC $\wedge$ OFFENSE) $\vee$ (EARTH MAGIC$\wedge$LOGISTICS) *under various policies ?* Some work has been done (ansaExtended); however, it is not part of the Monte Carlo approach. Also, parallelize the computations with the inclusion of a library such as [4]. Finally, is there a simpler alternative for Algorithm 1 of Sect. 3.4?

There are certainly exciting times ahead of both the developers and the players. We are eagerly looking forward into that future!

Acknowledgements.

The author wants to thank the participants in the thread

http://heroescommunity.com/viewthread.php3?TID=17812;

a special thanks to Ecoris for his tests and techniques. His insight will affect future work for sure. Finally, the author thanks György Turán for fruitful discussions, and the referees for their insightful comments.

References

[1] B. Dawkins. Siobhan's Problem: The Coupon Collector Revisited. *The American Statistician*, 45(1):76–82, 1991.

[2] B. Efron and R. Thisted. Estimating the Number of Unseen Species: How Many Words Did Shakespeare Know? *Biometrika*, 63(3):435–447, 1976.

[3] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[4] M. Mascagni. SPRNG: A Scalable Library for Pseudorandom Number Generation. In *Parallel Processing for Scientific Computing*, 1999.

[5] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[6] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.