한국게임학회 논문지
제5권 제2호 2005년 11월

게임 기획

True Count 기법에 의거한 블랙잭 게임 전략

김병수
건양대학교 전산게임학과
bskim@konyang.ac.kr

The Blackjack Game Strategy Using the True Count Method

Byung-soo Kim
Dept. of Computer Game, Konyang University

요약

카지노에서 성행하고 있는 블랙잭 게임을 이기기 위한 노력이 전세계적으로 경주되어 왔다. 본 논문에서는 하이 로우(high-low) true count 기법에 의거하여 게임을 유리하게 전개하기 위한 게임 전략을 선보인다. 켈리 기준이 왜 필요하며 그 기준에 의거한 베팅 전략을 강원랜드 블랙잭 규칙을 엄격히 적용하여 공격적 및 보수적인 베팅으로 구분하여 210만슈 이상 시뮬레이션을 수행하였다. 그 결과 공격적인 베팅이 이익면에서는 가장 유리하였으나 반면에 단기간의 위험도는 가장 크다는 사실을 알 수 있었다.

Abstract

The effort to win the blackjack game in the casino has been taken globally. This article presents a game strategy to win the game based on the high-low true count technic. This article will also identify the necessity of Kelly criterion and the betting strategies based on the standard is applied to the blackjack rules of Kangwon Land. By the way, bettings are distinguished as aggressive and conservative one and the simulations more than 2 million shoes has been performed. As a result, it is revealed that the aggressive betting is most advantageous in winning money while it implies the biggest risk.

Key Words: Kelly, true count, blackjack

1. 서론

블랙잭은 카지노에서 바카라(baccarat)게임과 함께 주요 수입원의 하나로 간주하고 있는 전 세계적으로 널리 알려진 게임이다.

과거에 블랙잭은 플레이어가 딜러에 앞서 히트(hit) 또는 스탠드(stand)를 결정하기 때문에 일반적으로 딜러에게 조금 더 승산이 있다고 간주되어 왔다. 그동안 딜러보다 플레이어에게 더 유리하다고 알려진 전략들이 발표되어 왔으며, 1956년에 현재의 기본 전략(basic strategy)과 유사한 전략이 등장하여 딜러와의 간격을 좁혔다.

마침내 1962년도에 MIT 대학의 Thorp 교수에 의한 카드 세기(card counting) 기법이 담긴 책 Beat the dealer 가 발간되어 선풍적인 대중의 인기를 얻게 되었다.[1] 당시의 컴퓨터의 도움으로 몬테 카를로 시뮬레이션(Monte Carlo simulation)이라 불리는 통계학적인 분석을 통하여 Thorp

교수는 10(J, Q, K도 10으로 간주) 또는 A와 같은 높은 숫자가, 5나 6과 같은 낮은 숫자보다 덱(deck)에 상대적으로 많이 남아있으면 딜러보다 플레이어에게 유리하다는 사실을 발견하였다. 그 결과로 카드 세기 기법이 탄생하였다. 즉, 플레이어는 게임을 하는 동안 10이나 A가 5나 6보다 덱에 많이 남아있는 동안에는 베팅(betting)을 작게하고 그 반대의 경우에는 큰 베팅을 하면 결과적으로 플레이어에게 승산이 있다는 것이다.

Thorp 교수 이후 수십가지의 카드 카운팅 전략이 발표되었으며[2], 그 결과로 카지노측은 블랙잭 규칙을 나름대로 딜러에게 유리하게끔 조금씩 강화시켜 왔다. 현재 전 세계의 각 카지노마다 조금씩 블랙잭 규칙이 다르며 그 결과로 인한 플레이어의 이득은 각 카지노마다 차이가 나는 실정이다.

본 논문에서는 한국에서 유일하게 내국인 입장이 허용되는 강원랜드 카지노의 블랙잭 규칙에 의거한 하이 로우 방식의 카드 카운팅 기법을 적용할 경우 플레이어에게 어느 정도의 이익이 발생하는지 컴퓨터 시뮬레이션을 통하여 규명할 것이다. 또한 켈리 기준(Kelly Criterion)의 범위를 크게 벗어나지 않는 범위에서 베팅 전략을 상이하게 적용하여 어떤 방식이 강원랜드의 룰에 가장 유리한지도 시뮬레이션을 통하여 밝히고자 한다.

2. 켈리 기준

켈리 공식은 전화선상에서 발생하는 간섭현상을 포함하는 문제를 풀기위하여 1956년 켈리에 의하여 도입되었다.[3] 켈리 효용함수(Kelly utility function)는 위험의 총량을 최소화함과 동시에 최대한의 승리를 도출할 수 있는 최적의 베팅(betting)을 수학적으로 정확하게 계산하는 방법을 제시한다.

효용함수는 $U(x) = x^a,\ 0 \le a \le \infty$ 또는 $U(x) = \log x$ 와 같은 것들이 있으며, 일단 효용함수가 정의되면 자본효용의 기대치의 극대화를 목표로 한다. 켈리 기준은 $\log X$ 대화하는 베팅으로서, E는 기대치, X는 자본금을 의미한다. 이러한 켈리 기준이 유용한 척도로서 쓰이게 되는 이유에 대하여 살펴보기로 한다.

2.1 동전 확률 및 파산

동전을 던져서 앞면이 나오면 이긴다고 하고, 동전의 앞면과 뒷면이 나올 확률을 각각 p 와 q(=1-p)라 하고 p 〉 1/2 라고 가정하자. 초기 자본금을 X_0, n번 시행 후의 자본금을 X_n 이라고 할 때, $E(X_n) = X_0 + \sum_{k=1}^{n} E(B_k T_k) = X_0 + \sum_{k=1}^{n} (p-q)E(B_k)$가 된다. (여기서 B_k는 k번째의 베팅 금액, k번째의 시행이 앞면이면 $T_k = 1$, 그렇지 않으면 $T_k = -1$ 이다.)

각각의 k 번째의 시도에서 $E(X_n)$을 최대화하기 위해서는 p-q〉0 이므로 $E(B_k)$ 를 최대화하면 된다. 이 경우 각각의 시행마다 자본금 전부를 베팅해야 하며 $B_1 = X_0$ 이고 이길 경우 $B_2 = 2X_0$, 계속 이길 경우 $B_k = kX_0$... 등등이 된다. 그러나 파산 확률이 $1-p^n$이 되며 $\lim_{n \to \infty}[1-p^n] = 1$ 이 되므로 파산은 필연적이다. 또한 파산 위험을 최소화하기 위해서는 각 B_k를 최소화하면 되지만 수익성 역시 보잘 것 없게 되는 단점이 있다.

따라서 양 극단의 사이값에서 베팅을 해야 하며, i번째의 베팅을 $B_i = f X_{i-1},\ 0 \le f \le 1$ 이라 하면, $X_n = X_0 (1+f)^S (1-f)^F$, $S+F = n$ 이 된다. 이 경우 $\Pr(X_n = 0) = 0$ 이 되므로 파산은 피할 수 있다.

2.2 켈리 전략

$e^{n\log[X_n/X_0]^{1/n}} = X_n/X_0$ 을 이용하여 각 시행시의 지수증가율의 측정치 $G_n(f)$ 을

$G_n(f) = \log[X_n/X_0]^{1/n} = (S/n)\log(1+f) + (F/n)\log(1-f)$ 와 같이 정의하자.

켈리는 성장률 계수의 기대치 g(f)를 최대화하는 것을 목표로 삼았으며,

$$
\begin{aligned}
g(f) &= E\{\log[X_n/X_0]^{1/n}\} \\
&= E\{(S/n)\log(1+f) + (F/n)\log(1-f)\} \\
&= p\log(1+f) + q\log(1-f)
\end{aligned}
$$

가 된다.

이때 $g(f) = (1/n)E(\log X_n) - (1/n)\log X_0$ 가 되므로 $g(f)$를 최대화하는 것은 $E\log X_n$ 을 최대화하는 것과 동일하다.

$g(f)$를 미분하면 $g'(f) = (p-q-f)/\{(1+f)(1-f)\}$ 가 되며, $f = f^* = p-q$ 일때 $g'(f) = 0$이 된다.

$g''(f) < 0$이고 $g'(f)$의 연속성 성질로 $g(f)$ 는 $f = f^*$, $g(f^*) = p\log p + q\log q + \log 2 > 0$ 이 됨을 알 수 있다. 또한 $0 < f^* < f_c < 1$ 의 범위내에서 $g(f_c) = 0$인 $f_c > 0$가

존재한다. 위의 사실로 우리는 다음과 같은 사실을 알 수 있다.

(정리 1)

(1) n이 커짐에 따라 $(0, f_c)$구간에 속하는 f 에 대하여 일정목표액 M 보다 큰 X_n 을 항상 얻을 수 있다.

(2) i번째 시행시의 단위베팅의 결과를 U_i 라 하고, 성공확률을 $p_i > 1/2$라 하자. $E\log X_n$ 은 $E\log(1+f_iU_i)$ 를 최대화시키는 $f_i^* = p_i - q_i$ 인 f_i^* 를 선택함으로써 최대화된다.

(3) 다른 어떤 대체 전략보다 $E\log X_n$ 을 최대화하는 켈리 전략이 우수하다.

(증명) Thorp[4]과 Breiman[5]의 정리로부터 명백하다.

2.3 블랙잭에의 적용

Kelly 기준은 블랙잭에도 적용할 수 있으며, 그 분석은 동전 확률보다 더 복잡하다. 왜냐하면 true count가 변함에 따라 승률은 동전 확률처럼 일정하지가 않으며, 동일 숫자의 분할(pair splitting), 더블 다운(double down), 블랙잭 출현 등과 같은 플레이어에게 유리한 국면이 얼마나 자주 전개되느냐에 따라 그 시점에서의 이익 발생률이 달라지기 때문이다.

일반적으로 true count 가 불리한 상황에서 플레이어는 최소한의 베팅을 하면서 상황이 호전되기를 기다린다고 가정하자. 이때 우리는 불리한 상황에서 최소한의 일정 베팅 f_0 를 하는 경우와, 유리한 경우의 베팅 f 에 비례하는 af $(0<a<1)$를 불리한 경우에 베팅하는 두가지 경우를 가정할 수 있다.

이기거나 질 확률 X 에 대하여 P(X=1) = 0.51, P(X=-1) = 0.49 라 하자. 이때 첫 번째 경우는 두번째 경우의 특수한 상황이므로 Kelly 성장률 g(f)는

$$g(f) = 0.5(0.51\log(1+f)+0.49\log(1-f)) + 0.5(0.49\log(1+af)+0.51\log(1-af))$$

라고 할 수 있다. 기대 이익을 기대 베팅으로 나누면 0.02(1-a)/(1+a) 가 되며 만약 a = 0 이라면 $f^* = 0.02$ 가 된다. a가 1로 증가할 때 f^* 는 0.02에서 0으로 수렴하게 된다. 이제까지의 설명에서 우리는 다음과 같은 결론을 얻을 수 있다.

(정리 2)

true count 가 불리한 상황에서는 가능한한 최소한의 베팅을 하는 것이 Kelly 성장률을 극대화한다.

(증명) Thorp[6] 으로부터 명확하다.

3. 하이-로우 시스템

자본금을 늘리기 위한 수많은 베팅 전략들이 존재해 왔으며 이들은 크게 두가지로 나누어진다.

첫째는 i번째 시행에서 졌을때 그 다음 i+1 번째 시행에서는 i번째 시행까지 만회하기 위하여 더 많은 베팅을 하는 방식으로, 마팅게일(Martingale) 및 수퍼(super) 마팅게일, 피보나치(Fibonacci) 방법, 아렘버트(D'Alembert) 시스템, 2-5 시스템 등이 있다.

두 번째는 i번째 시행에서 이겼을때 그 다음 i+1 번째 시행에서는 더 많은 베팅을 하는 방식으로 달의 급진행(Dahl's Progression), 밀고 당기기(Up and Pull), 호일 방식, 역 마팅게일 시스템들이 그것이다.

이러한 모든 방식들은 일견 그럴싸하게 보이지만 일정 베팅(flat betting) 보다 더 큰 손실로 나타나게 되며 결국에는 파산에 이르게 된다.[7]

3.1 카드 계산 기법

Thorp 교수의 명저 'Beat the dealer' 에서 제시한 카드 계산 기법 이후 수 십 가지의 카드 계산 방법이 등장하게 되었다. 표 1은 그 중에서 비교적 많이 알려진 기법들을 게임 효율, 베팅 효율 및 보험(Insurance) 효율을 중심으로 비교 나열한 것이다[8].

표 1에서 BC는 베팅 연관성(Betting Correlation), PE는 게임 효율(Playing Efficiency), IC는 보험 연관성(Insurance Correlation)를 의미하며 난이도는 위에서 아래로 내려갈수록 실전에서 운용하기가 어렵다.

3.2 하이 로우 시스템

표 1에서 보는바와 같이 대부분의 카드 세기 시스템들은 덱에서 카드가 빠져 나갈 때 로우 카드이면 +로, 10이나 A 같은 하이카드는 -로 계산한다. 그 결과로 중간 단계에서 전

체 계산이 +인 경우 상대적으로 더 많은 베팅을 하며, 반대의 경우 최소한의 베팅을 하면서 상황이 +로 좋아지기를 기다리는 전략을 구사한다.

전략	A	2	3	4	5	6	7	8	9	T	BC	PE	IC
Canfield Expert	0	0	1	1	1	1	1	0	-1	-1	.87	.63	.76
Canfield Master	0	1	1	2	2	2	1	0	-1	-2	.92	.67	.85
Hi-Lo	-1	1	1	1	1	1	0	0	0	-1	.97	.51	.76
Hi-Opt I	0	0	1	1	1	1	0	0	0	-1	.88	.61	.85
Hi-Opt II	0	1	1	2	2	1	1	0	0	-2	.91	.67	.91
K-O	-1	1	1	1	1	1	1	0	0	-1	.98	.55	.78
Omega II	0	1	1	2	2	2	1	0	-1	-2	.92	.67	.85
Red Seven	-1	1	1	1	1	1	.5	0	0	-1	.98	.54	.78
Revere Adv. Plus-Minus	0	1	1	1	1	1	0	0	-1	-1	.89	.59	.76
Revere Point Count	-2	1	2	2	2	2	1	0	0	-2	.99	.55	.78
Silver Fox	-1	1	1	1	1	1	1	0	-1	-1	.96	.53	.69
Unb. Zen 11	-1	1	2	2	2	2	1	0	0	-2	.97	.62	.84
Uston Adv. Plus-Minus	-1	0	1	1	1	1	1	0	0	-1	.95	.55	.76
Uston APC	0	1	2	2	3	2	2	1	-1	-3	.91	.69	.90
Uston SS	-2	2	2	2	3	2	1	0	-1	-2	.99	.54	.73
Wong Halves	-1	.5	1	1	1.5	1	.5	0	-.5	-1	.99	.56	.72
Zen Count	-1	1	1	2	2	2	1	0	0	-2	.96	.63	.85

〈표1〉 카드 카운팅 시스템

하이 로우 시스템은 Thorp에서 시작하여 Braun을 거쳐 Wong에서 완성도를 보인 카드 세기 전략으로 단순하면서도 강력한 전략이라고 할 수 있다. Hi-Opt II, Uston APC, Zen 과 같은 시스템들은 하이 로우 기법보다 약간(0.1%) 더 우수할지도 모르지만, 난이도에 따른 실전에서의 계산 오류 가능성을 고려하면 차라리 하이 로우 시스템이 훨씬 안정적이라고 할 수 있다.

하이 로우 시스템에서는 2,3,4,5,6 과 같은 로우 카드를 -1로 10,J,Q,K 및 A는 +1로 계산한다. 그런 다음 현재의 계산된 숫자(running count)를 덱에 남아 있는 벌(각 벌은 52장) 수로 나눈 수(true count)에 의거하여 베팅의 크기가 조절되며, 또한 현 시점에서의 게임 전략도 트루 카운트에 의존적이 된다. 예를 들어 한 슈(shoe)가 6벌의 카드로 구성되며, 현재의 러닝 카운트(running count)가 7이고, 두 덱 반이 빠져나갔다면 트루 카운트는 7/(6-2.5) = 2 가 된다.

3.3 하이 로우 카드 카운팅 알고리즘

Wong[9]은 true count에 의거하여 기본적인 힛(hit) 또는 스탠드(stand) 외에도 동일 숫자의 분할(pair splitting), 더블다운(double down), 블랙잭 판별하기 등 모든 카드의 경우에 대한 대응 방안을 하이 로우 시스템에 의거하여 그의 저서 "Professional Blackjack"[9]에서 설명해 놓았다. 기본적인 히트 및 스탠드의 경우 이외의 모든 알고리즘은 다음과 같다.

(하이 로우 카드 세기 시스템 알고리즘)

```
if(dealer's card is A && true count >= 3)) bet insurance;
if(split card) {
    if(card number is A or 8)                split;
    if(card number is 2 or 3 or 7)
        if(dealer' s card >=2 && dealer' s card <= 7)      split;
    if(card number is 4)
        if(dealer' s card is 4 or 5)         split;
    if(card number is 6)
        if(dealer' s card >=2 && dealer' s card <= 6)      split;
    if(card number is 9)
        if(dealer' s card !=7 && dealer' s card != 10)     split;
        else if(true count >= 3)             split;
}
if(double down) {
    if(hard double) {
        if(the sum of the two cards          is 11) {
            if(dealer' s card != A)          doubledown;
            else if(true count >= 1)         doubledown;
        }
        if(the sum of the two cards          is 10) {
            if(dealer' s card != A && dealer' s card != 10)
                             doubledown;
            else if(true count >= 4)         doubledown;
        }

        if(the sum of the two cards          is 9) {
            if(dealer' s card == 2 && true count >= 1)
                doubledown;
            else if(dealer' s card == 7 && true count >= 3)
                doubledown;
            else if(dealer' s card > 2 && dealer's card < 7)
                doubledown;
        }
        if(the sum of the two cards      is 8 && the cards are
          not same number 4) {
            if(dealer' s card == 4 && true count >= 5)
                doubledown;
            else if(dealer' s card == 5 && true count >= 3)
                doubledown;
            else if(dealer' s card == 6 && dealer's card >= 1)
                doubledown;
        }
    }
    else if(soft double) {
        if(card number is A and 2) {{
            if(dealer' s card is 5 or 6 and true count > -1)
                doubledown;
```

```
        else if(dealer' s card is 4 and true count 〉 3)
                doubledown;
    }
    if(card number is A and 3) {
        if(dealer' s card is 5 or 6) doubledown;
        else if(dealer' s card is 4 and true count 〉 1)
                doubledown;
    }
    if(card number is A and 4) {
        if(dealer' s card is 5 or 6) doubledown;
        else if(dealer' s card is 4 and true count 〉 0)
                doubledown;
    }
    if(card number is A and 5) {
        if(dealer' s card is 4 or 5 or 6)  doubledown;
        else if(dealer' s card is 3 and true count 〉 4)
                doubledown;
    }
    if(card number is A and 6) {
        if(dealer' s card is 3 or 4 or 5 or 6) doubledown;
        else if(dealer' s card is 2 and true count 〉 1)
                doubledown;
    }
    if(card number is A and 7) {
        if(dealer' s card is 3 or 4 or 5 or 6) doubledown;
        else if(dealer' s card is 2 and true count 〉 0)
                doubledown;
    }
    if(card number is A and 8) {
        if(dealer' s card is 5 or 6 and true count 〉 1)
                doubledown;
        else if(dealer' s card is 4 and true count 〉 3)
                doubledown;
    }
  }
}
```

4. 강원랜드 블랙잭 규칙에 의거한 시뮬레이션

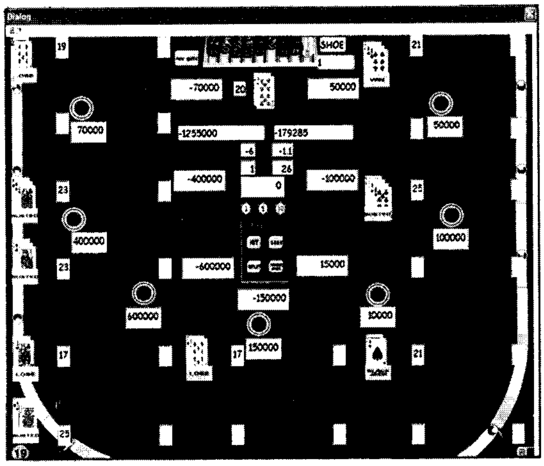


그림1. 강원랜드 블랙잭 모의 테이블

그림 1에서는 강원랜드 블랙잭 규칙에 의거한 테이블 운용을 visual c++로 프로그래밍한 결과로서 나타난 한 예이다. 강원랜드 블랙잭은 베가스 스트립(Vegas Strip) 방식에 비교적 가까우며 적용되고 있는 대표적인 규칙은 다음과 같다.

- 일반 테이블은 최대 베팅 10만원 및 30만원의 두 종류가 있으며 본 논문에서는 30만원 한도의 테이블만 고려하기로 한다.
- 한 슈의 카드는 6벌을 사용하며 한 덱 반(75%)에서 카드 자르기(penetration)를 한다.
- 카드 분할은 4번까지지만 가능하지만 A 카드는 두 번만 분할 가능하며 무조건 한 장씩의 카드만 받는다.
- 더블 다운은 카드가 두장인 경우 가능하며 카드 종류에 상관없다.
- 모든 카드는 분할 후 더블 다운이 가능하다.
- 블랙잭은 1.5 배를 지불하며, 보험은 베팅 금액의 절반까지 걸 수 있다.
- 딜러는 소프트(soft) 17에서 멈춘다.

본 논문에서는 베팅 전략을 공격적, 보수적으로 나누어 시뮬레이션 하였다. 임의의 플레이어가 일년에 10000슈씩 30년 플레이한다고 가정하여 각각 300000슈를 실행하였으며, 한번에 10000슈씩 30번을 독립적으로 수행하였다.

Kelly 및 트루카운트의 규칙을 강원랜드의 베팅 시스템에 적용한 결과, 보수적 및 공격적 베팅에 대한 도표를 표 2에

트루 카운트	<0	1	2	3	4	5	>6
보수적	10	14	68	116	188	236	300
공격적	10	18	87	147	238	300	300

단위: 천원

<표2> 트루카운트를 강원랜드 베팅시스템에 적용한 결과

서 보여준다.

표 2의 베팅을 적용한 경우를 표 3과 4에서 보여준다.

플레이어	1	2	3	4	5	6	7
합계	60697	-8507	61496	40432	24880	-42749	17954
평균	2023	-283	2050	1348	829	-1425	599

단위: 만원

<표3> 보수적 베팅

전체평균(10000슈) : 734(만원)

플레이어	1	2	3	4	5	6	7
합계	73167	-67234	101082	55833	12501	-94721	17740
평균	2439	-2241	3369	1861	417	-3157	658

단위: 만원

<표4> 공격적 베팅

전체평균(10000슈) : 478(만원)

그림 2는 보수적 베팅의 경우를 각 10000슈씩 30번 수행한 경우의 그래프이다. +수익을 거둔 경우가 -수익의 경우보다 월등히 많으며 수익 자체의 폭도 +인 경우가 훨씬 큼을 알 수 있다.

그림 3은 보수적 베팅과 공격적 베팅을 선 그래프로 비교한 것이다. 대부분의 플레이어의 경우 공격적 베팅의 플레이어가 보수적 플레이어보다 얻는 이익이 많음을 알 수 있지만 손실이 난 플레이어의 경우는 그 반대임을 알 수 있다. 그 손실의 폭이 비교적 큰 관계로 전체 수익률은 오히려 보

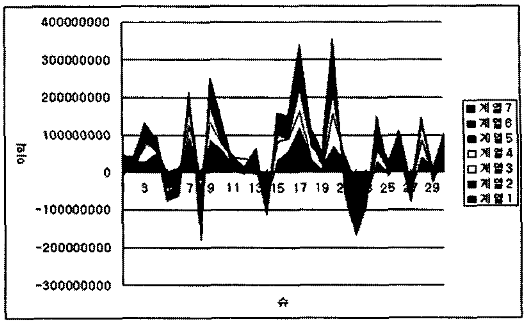


그림2. 보수적 베팅의 경우

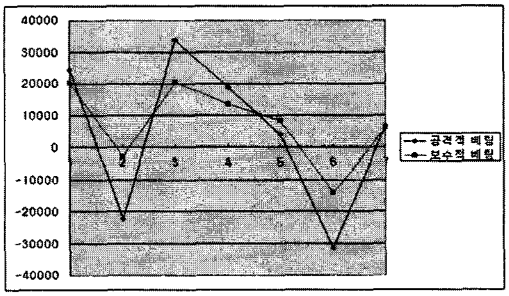


그림3. 보수적 베팅과 공격적 베팅의 수익 비교

수적 베팅의 경우가 더 좋다는 것을 보여주고 있다.

그림 4와 5는 보수적 베팅 및 공격적 베팅을 각각 100000슈식 시뮬레이션 한 결과이다. 두 그래프에서 보듯이 단기적으로는 편차(fluctuation)가 있지만 비교적 양호하게 수익

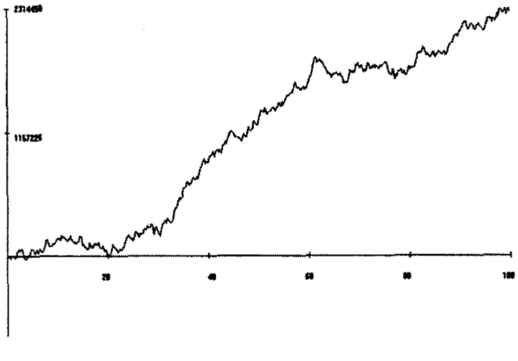


그림4. 보수적 베팅의 100000슈 시뮬레이션 결과

을 얻음을 알 수 있다.

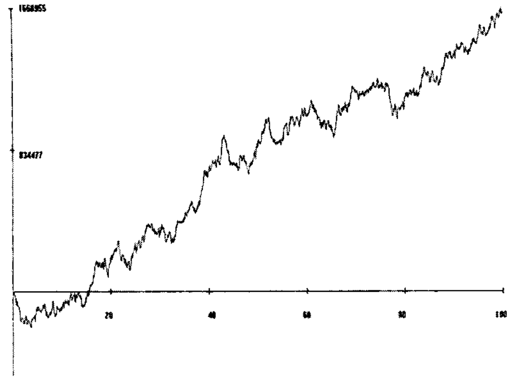


그림5. 공격적 베팅의 100000슈 시뮬레이션 결과

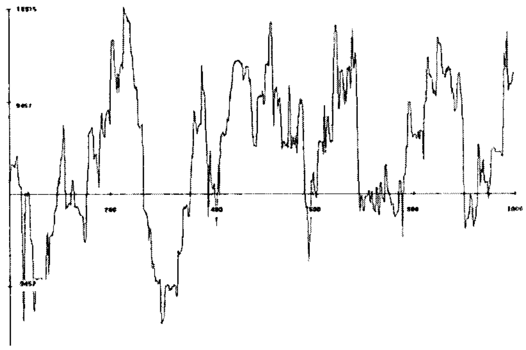


그림6. 공격적 베팅의 극단적인 예

그림 6에서는 공격적 베팅에 의거하여 단기적으로 1000슈 시뮬레이션의 극단적인 예이다. 이 예에서 볼 수 있듯이 단기적으로는 이익과 손실의 편차가 극심하게 나타날 수 있다. 따라서 플레이어는 연속 이익(winning streak)뿐만 아니라 연속손실(losing streak)의 경우도 겪게 될 가능성이 많음을 항상 유의해야 한다.

5. 결론

본 논문에서는 켈리 기준 및 트루 카운트에 의거한 베팅을 강원랜드 블랙잭 규칙에 적용하였으며, 켈리 기준에 크게 벗어나지 않는 한도내에서 보수적인 베팅 및 공격적 베팅으로 각 210만슈씩 시뮬레이션 하였다.

시뮬레이션 결과 각 플레이어마다 보수적인 베팅보다 공격적 베팅이 더 큰 편차를 보였으며 평균적으로는 보수적 베팅이 더 좋은 수익률을 가짐을 도표 및 그래프에서 보여준다.

개별 플레이어의 수익은 슈가 증가할수록 수익이 증대하는 함수 형태로는 결코 보여주지 않으며 오히려 극심한 연속 이익(winning streak)과 그 반대의 연속 손실(losing streak)을 번갈아 보여주는 경우가 많음을 그래프에서는 보여주고 있다. 이것은 실제 상황에서도 빈번히 발생하는 현상으로서 플레이어는 연속 손실에 빠지지 않게끔 또는 불가항력적으로 연속 손실의 덫에 걸리더라도 손실을 최소화하게끔 세심한 주의가 필요하다고 하겠다.

참고문헌

[1] E.O.Thorp, "Beat the Dealer", New York: Random House, 1962

[2] www.blackjackforumonline.com/content/hundred.htm

[3] J.L.Jr Kelly,, "A New Interpretation of Information Rate," Bell Systems Technical Journal,35, pp 917-926, 1956

[4] E.O.Thorp, "Optimal Gambling Systems for Favorable Games," review of the International Statistical Institute, pp 273-293, 1969

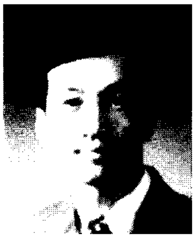
[5] L. Breiman, "Optimal Gambling Systems for Favorable Games," Proceedings of the Berkeley symposium on Mathematical Statistics and Probability, 1961

[6] E.O.Thorp, "The Kelly Criterion in Blackjack, Sports Betting, and the Stock Market," the 10th International Conference on Gambling and Risk Taking, June 1997

[7] www.bjmath.com/bjmath/progress/prog1.htm

[8] www.bjmath.com/bjmath/counting/OSRating.htm

[9] S.Wong, "Professional Blackjack," Pi Yee Press, 1994



김 병 수

1981년 서울대학교 수학과 졸업
1987년 오클라호마 주립대 석사(전산학전공)
1993년 고려대학교 이학박사(전산학전공)
1991년 ~ 현재 건양대학교 전산게임학과 교수
관심분야 : 게임이론, 가상현실

논문투고일 2005년 3월 28일
심사완료일 2005년 4월 22일