HYPER DICE BACKGAMMON OF FINITE SIZE

AMIR M. RAHIMI

Abstract. In this note, we extend the notion of standard backgammon to a more general setting and call it hyper dice backgammon (or HD-gammon for short) of size $n \geq 6$ (a positive even integer) by extending the regular die to a hyper die (i.e., hyper cube) with n faces and the board from 24 pips to $4n$ pips, where $n = 2k \geq 6$ and there are $4k + 3$ checkers for each player. The rules of the game are similar to the rules of standard backgammon when $n = 2k = 6$ and the number of the n-sided dice depends on n. Finally, we include a list of references related to some theoretical studies on standard backgammon.

1. A Generalization of Standard Backgammon

This note is intended to introduce mathematicians, computer scientists, artificial intelligence (AI) professionals, and game theorists to hyper dice backgammon. Also, we will introduce a new concept in this field. In this section we start with the definition (notion) of the *hyper dice backgammon* and conclude the section with a question and a motivational comment together with an open problem related to the (infinite) hyper dice backgammon. In the second section, we recall the rules of *standard backgammon* with a brief motivational history on *complex board games* and conclude the paper with a selected list of references related to some theoretical studies on the standard backgammon.

Definition 1.1. A *hyper die* of size $n = 2k \geq 6$ is a *hyper cube* with n faces, where each face (or side) is marked (denoted) by exactly one distinct number from 1 to n. For the sake of convenience, by an n-die (or n-dice for the plural case), we mean a hyper die of size n or an *n-sided cube*.

We now extend the notion of standard (classic) backgammon to a more general setting and call it hyper dice backgammon.

This research was in part supported by grant no.96130029 from IPM.

Definition 1.2. The hyper dice backgammon (or *HD-gammon* for short) of size $n = 2k \geq 6$ (a positive integer) is defined to be as follows:

(a) Extending the regular die to a hyper die (i.e., hyper cube) with n faces (see the above definition).
(b) Extending the board from 24 pips to $4n$ pips, where $n = 2k \geq 6$.
(c) Extending the number of checkers from 15 to $4k+3$ for each player.
(d) HD-gammon is played on a board with $2n + 3$ checkers per player, and $4n$ pips (organized into four tables of n pips each) that checkers may rest on.
(e) The rules of the game are similar to the rules of the classic backgammon when $n = 2k = 6$ as defined in the next section for the sake of completeness and comparison.
(f) For playing the game, we divide the set of white [resp. black] checkers into four parts of sizes $k - 1$, k, $k + 2$, and $k + 2$ and put them in the corresponding pips as in the classic backgammon game when $k = 3$. That is, $k + 2$, k, $k + 2$, and $k - 1$ checkers in positions n, $n + 2$, $2n + 1$, and $4n$, respectively.
(g) The number of n-dice is $m = n/3$, where n is a multiple of 3 and if the remainder of $n/3$ is not zero, we use $(m + 1)$ dice, where m of them are n-sided dice and one is an $n/2$-sided die. For example, if $n = 14$, we use 4 dice of size 14 and one die of size 7. Of course, we can study the game when the number of n-dice are between 2 and m or $m + 1$ for mathematical, probabilistic, and complexity classification purposes.

We now end this section with a question and a motivational comment regarding the HD-gammon together with an open problem related to the infinite HD-gammon.

Question. What is the minimum number of n-dice in an HD-gammon of size n to have the same *complexity class* with the standard backgammon?

Comment. The author believes that using a *Quantum Programming Language* or some *quantum computational* aspects such as *super position* or *entanglement* could be a powerful machinery to study the HD-gammon (especially, when its size is very large or even *infinite*, i.e., a board of (countable) infinite, infinitely many checkers, and *infinitely many finite-sided dice*).

Open Problem. What is a reasonable arrangement or an initial position for infinitely many checkers in the infinite HD-gammon as noted in the above comment?

2. A Brief Background on the Standard Backgammon

In this section, we recall the rules of standard backgammon (e.g., *doubling cube*, see ($*$) below) with a brief motivational history on complex board games and conclude the paper with a selected list of references related to some (theoretical) studies on the standard backgammon.

• Note that the material in this section is taken from different sources and edited by the author for the sake of completeness and comparison. For the rules of the game see, for example, `http://www.bkgm.com` and other backgammon-related articles and information are available at Backgammon Galore.

We now provide a brief motivational history on complex board games.

Ever since the days of Shannon's proposal [19] for a chess-playing algorithm and Samuel's checkers-learning program [18], the domain of complex board games such as Go, chess, checkers, Othello, and backgammon has been widely regarded as an ideal testing ground for exploring a variety of concepts and approaches in artificial intelligence and machine learning. Such board games offer the challenge of tremendous complexity and sophistication required to play at expert level. At the same time, the problem inputs and performance measures are clear-cut and well-defined, and the game environment is readily automated in that it is easy to simulate the board, the rules of legal play, and the rules regarding when the game is over and determining the outcome.

Academic research on backgammon has been largely confined to two areas: *strategies* for using the doubling cube (offering or accepting), see ($*$) below, and *computerized backgammon players*. Examples of the former include Thorp [29] and [30, this volume, pp. 237–265], Keeler and Spencer [10], Orth [13], Zadeh and Kobliska [33], Zadeh [32, 34], and Buro [6]. Computerized players, often based on *neural networks*, are now better than the best human players [22].

Complex board games such as Go, chess, checkers, Othello and backgammon have long been regarded as great test domains for studying and developing various types of *machine learning procedures*. One of the most interesting learning procedures that can be studied in such games is *reinforcement learning* from self-play. In this approach, which originated long ago with Samuel's (1959) checkers program, the program plays many games against itself, and uses the "reward" signal at the end of each game to gradually improve the quality of its move decisions.

• The rest of the section is devoted to the rules of the backgammon.

3. The Game of Backgammon

Backgammon is said to be one of the oldest board games in the world. Its roots may well reach back 5,000 years, into the former *Mesopotamia*. From there, it spread out in variants to Greece and Rome as well as to India and China.

Backgammon is a game of *chance and skill*. It was played in England in 1743 when Edmond Hoyle fixed the rules for backgammon in Europe. After a revision in 1931 in the US, these rules are still in use today. What turned it into the subtle and skillful game we know today was a brilliant innovation in the rules early in the 20th century: the doubling cube (see $(*)$ below). Who invented it is unknown, but it emerged in American gambling clubs some time in the 1920s.

Backgammon is a *two-player perfect information game* in which a player's move is governed by a throw of the dice. Backgammon is played on a board with 15 checkers per player, and 24 points (organized into four tables of six points each) that checkers may rest on. The objective of the game is to be first to *bear off*, i.e., move all fifteen checkers off the board. Backgammon is a member of the *tables family*, one of the oldest classes of board games. Backgammon involves a combination of *strategy and luck* (from rolling dice). While the dice may determine the outcome of a single game, the better player will accumulate the better record over a series of many games, somewhat like poker. With each roll of the dice, players must choose from numerous options for moving their checkers and anticipate possible counter-moves by the opponent. The optional use of a doubling cube allows players to raise the *stakes* during the game. Like chess, backgammon has been studied with great interest by *computer scientists*. Owing to this research, *backgammon software* has been developed that is capable of beating world-class human players (see TD-Gammon for an example).

Players take turns rolling the two dice to move checkers toward their respective *home boards*, and ultimately off the board. The first player to move all of his checkers off the board wins. When the dice are rolled, they are applied sequentially to the board by moving one checker by the amount on one die, and then another checker by the amount on the other die. The checker moved with the second die may be the same one that was moved with the first die. Thus, if a player has only one checker on the 3 point and another on the 6, and rolls a 2 and a 1, then the checker on the 6 point may be moved to the 4 point, and the checker on the 3 point may be moved to the 2 point (written 6-4 3-2). Alternatively, the player may play 6-5 3-1 or 6-5 5-3. If doubles are rolled, the player may move four checkers.

Thus, if double-ones were rolled in the above situation, the player could move 6-5 5-4 3-2 2-1. No moving checker may land on a point that has two or more opponent checkers. If a point has exactly one opponent checker (a *blot*), it may be landed on and "hit." This complicates the game. After the two players can no longer hit each other, they enter a "race" to get their checkers off the board. First, they "bear in" checkers onto their *6 home board points.* Once all 15 of a player's checkers are in, that player begins the "bearoff," moving checkers off the board. This is done by moving checkers off the points that correspond to the dice rolls. If no checker appears on that point, one from a higher point must be moved down. If there are no checkers on higher points, the highest checker may be borne off.

In addition, the player wins *double the normal stake* if the opponent has not taken any checkers off; this is called *winning a gammon.* It is also possible to win a *triple-stake* "backgammon" if the opponent has not taken any checkers off and has checkers in the far most quadrant; however, this rarely occurs in practice.

The one-dimensional racing nature of the game is made considerably more complex by two additional factors. First, it is possible to land on, or "hit", a single opponent checker (called a "blot") and send it all the way back to the far end of the board. The blot must then re-enter the board before other checkers can be moved. Second, it is possible to form blocking structures that impede the forward progress of the opponent checkers. These two additional ingredients lead to a number of subtle and complex expert strategies [11, 16].

4. (∗) The Doubling Cube

Like poker, Backgammon is a gambling game. It has an element of chance (introduced by the dice), and there is a notion of the stake for which the players are playing. (Of course, this does not have to be actual money; in a tournament it could be points, for example.) The actual amount that changes hands can be more than the stake, however. For instance, in certain winning positions called gammon and backgammon, the stake is doubled or tripled, respectively. The other way the stake can change is by means of the doubling cube.

Backgammon differs from other common games (checkers, chess, etc.) because there is a formal way to increase the stakes as the game progresses. It is important to know one's current chances of winning in order to decide whether or not to increase the stakes. The doubling cube, which was invented in the 1920s, is how the stakes may be increased during the game. The cube starts in the middle of the board with a value of 1. When one player (call him A) decides that his chances of winning are just right, he gives the cube to his opponent (B) before A's turn to roll. The opponent can either accept or reject the cube. If B rejects, the game is over and he must pay up. If he accepts, the two players play for twice as much money as before. Then, B possesses the cube, and A cannot use it until A is doubled and accepts.

If one of the players thinks that she is in the position to win the game, she can turn the doubling cube and announce a double, which means that the total stake will be doubled. If her opponent refuses the double, he immediately loses his (undoubled) stake and the game is finished. If he accepts the double, the stakes are doubled and, as a compensation, the doubling cube is handed over to him and he gets the exclusive right to announce the next double. (He is now said to own the cube.) If the luck of the game changes so that he later judges that he is now winning, he will be in a position to announce a so-called redouble, which means that the stake is doubled again. If the first player refuses the double, she now loses the doubled stake; if she accepts, the game goes on with a redoubled stake, four times the original value.

There is no limit to how many times the stake can be doubled, but the right to announce a double switches from one player to the other every time it is exercised. (Initially either player can double - no-one owns the cube.) This nice subtlety leads to a variety of tactical possibilities and problems.

References

[1] R. Andrews and J. D. Tickle, *A.: Survey and critique of techniques for extracting rules from trained artificial neural networks*, Knowledge-Based Systems, **8.6** (1995), 373–389.

[2] J. Baxter, A. Tridgell, and L. Weaver, *KnightCap: A chess program that learns by combining TD(A) with minimax search*, Proc. ICML-98, Madison, WI, 1998, pp. 28–36.

[3] A. Benjamin and M. A. Ross, *Enumerating backgammon positions: the perfect hash*, Interface: Undergraduate Research at Harvey Mudd College, **16.1** (1997), 3–10.

[4] H. J. Berliner, *Computer backgammon*, Sci. Amer., **243.1** (1980), 64–72.

[5] H. J. Berliner, *BKG - A program that plays backgammon*, D. N. L. Levy, editor, Computer Games I, Springer-Verlag, 1988, 3–28.

[6] M. Buro, *Efficient approximation of backgammon race equities*, International Computer Chess Association Journal, **22.3** (1999), 133–142.

[7] R. H. Crites and A. G. Barto, *Improving elevator performance using reinforcement learning*, D. Touretzky et al. (Eds.), Advances in Neural Information Processing Systems, Vol. 8, MIT Press, Cambridge, MA, 1996, pp. 1017–1023.

[8] C. Goren, *Goren's Modern Backgammon Complete*, Doubleday, New York, 1974.

[9] O. Jacoby and J. R. Crawford, *The Backgammon Book*, Bantam Books, New York, 1970.

[10] E. B. Keeler and J. Spencer, *Optimal doubling in backgammon*, Operations Research, **23.6** (1975), 1063–1071.

[11] P. Magriel, *Backgammon*, Times Books, New York, 1976.

[12] R. Nowakowski (editor), *Games of No Chance*, Cambridge U Press, 1996.

[13] P. J. Orth, *A comment on Optimal Doubling in Backgammon*, Operations Research, **24.6** (1976), 1179.

[14] N. Papahristou and I. Refanidis, *Improving Temporal Difference Learning Performance in Backgammon Variants*, Advances in Computer Games (ACG-13), LNCS, Vol. 7168, 2012.

[15] J. B. Pollack and A. D. Blair, *Co-evolution in the successful learning of backgammon strategy*, Machine Learning, **32** (1998), 225–240.

[16] B. Robertie, *Advanced Backgammon (Vols. 1 and 2)*, The Gammon Press, Arlington, MA, 1991.

[17] B. Robertie, *Carbon versus silicon: Matching wits with TD-Gammon*, Inside Backgammon, **2.2** (1992), 14–22.

[18] A. Samuel, *Some studies in machine learning using the game of checkers*, IBM J. of Research and Development, **3** (1959), 210–229.

[19] C. E. Shannon, *Programming a computer for playing chess*, Philosophical Mag., **41** (1950), 265–275.

[20] R. S. Sutton, *Learning to predict by the methods of temporal differences*, Machine Learning, **3** (1988), 9–44.

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

[22] G. Tesauro, *Programming backgammon using self-teaching neural nets*, Artificial Intelligence, **134.1-2** (2002), 181–199.

[23] G. Tesauro, *Neurogammon wins Computer Olympiad*, Neural Computation, **1**, (1989), 321–323.

[24] G. Tesauro, *Optimal doubling in multi-outcome probabilistic games*, IBM Research, Unpublished manuscript, 1990.

[25] G. Tesauro, *Practical issues in temporal difference learning*, Machine Learning, **8** (1992), 257–277.

[26] G. Tesauro, *Temporal difference learning and TD-Gammon*, Comm. ACM, **38.3** (1995), 58–68, HTML version `http://www.research.ibm.com/massive/tdl.html`.

[27] G. Tesauro, *TD-gammon*, `http://www.scholarpedia.org/article/Td-gammon`.

[28] G. Tesauro and G. R. Galperin, *On-line policy improvement using Monte-Carlo search*, M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), Advances in Neural Information Processing Systems, Vol. 9, MIT Press, Cambridge, MA, 1997, pp. 1068–1074.

[29] E. O. Thorp, *Backgammon: part I, the optimal strategy for the pure running game*, Presentation at the Second Annual Conference on Gambling, South Lake Tahoe, NV, 1975.

[30] E. O. Thorp, *Backgammon: the optimal strategy for the pure running game*, In: S. N. Ethier and W. R. Eadington, (eds.) Optimal Play: Mathematical Studies of Games and Gambling. Institute for the Study of Gambling and Commercial Gaming, University of Nevada, Reno, (2007), 237–265.

[31] M. A. Wiering, *Self-Play and Using an Expert to Learn to Play Backgammon with Temporal Difference Learning*, Journal of Intelligent Learning Systems and Applications **2** (2010), 57–68.

[32] N. Zadeh, *On doubling in tournament backgammon*, Management Science, **23.9** (1977), 986–993.

[33] N. Zadeh and G. Kobliska, *On optimal doubling in backgammon*, Management Science, **23.8** (1977), 853–858.

[34] N. Zadeh, *Computation of Optimal Poker Strategies*, Operations Research, (July-August, 1977).

[35] N. Zadeh, *Winning Poker Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
[36] D. Zare, *Backgammon Ends*, GammonVillage, 2000.

MSC2010: 90C02, 68W02, 68T02, 68N02, 60G02, 91A02

Key words and phrases: Backgammon, hyper die of size n (= n-sided die), hyper dice backgammon (= HD-gammon)

Department of Mathematical Sciences, University of South Africa, P.O. Box 392, 0003 Pretoria, South Africa, and, School of Mathematics, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395-5746, Tehran, Iran.

E-mail address: `amrahimi@ipm.ir`