Maximizing the Runs Scored by a Team in Cricket using Genetic Algorithm

Theviyanthan Krishnamohan
Faculty of Information Technology, University of Moratuwa, Sri Lanka
Email: theviyanthan@outlook.com

Abstract: **Batting teams get two resources in cricket to score runs with, viz. wickets and balls. Trying to score as many runs as possible off every ball a team faces risks losing wickets. Trying to preserve wickets comes at the cost of not enough runs being scored. Thus, to maximize the runs scored, teams need to make efficient use of the balls and wickets available to them. To that end, teams primarily employ two types of batsmen-aggressive batsmen who try to score as many runs as possible off the balls they face and defensive batsmen who try to protect their wickets. Having too many aggressive batsmen helps a team score more runs off the balls they face but it stands the risk of a team losing all its wickets before they face all of their allotted balls. Having too many defensive batsmen helps a team face all the balls allotted to them but they may not end up scoring enough runs. Hence, selecting the right combination of defensive and aggressive batsmen is essential to maximize the runs a batting team scores. However, this is a computationally complex problem to solve. This study proposes the use of genetic algorithm to optimize the batting lineup of a team to help it maximize the runs scored in an innings. The test results indicate, by using the genetic algorithm, the number of runs scored batting first by the full member teams in the last five years can be improved by 5.46% on average.**

Keywords**: Cricket, Genetic Algorithm, Evolutionary Computing, Batting order, Optimization**

I. INTRODUCTION

A. Cricket

Cricket is a bat-and-ball game that originated in England and is played between two teams of eleven players each [1]. It is the second most popular sport in the world, only behind soccer, with more than a billion fans all over the world [2]. The sport is especially popular in the Indian subcontinent, Australia, England, New Zealand, South Africa, and the West Indies.

The two teams take turns to bat and bowl, the order of which is decided by virtue of a coin toss. A limited-overs game in cricket consists of two innings. An innings is when one team bats and the other team bowls. At the end of an innings, the teams swap their roles and the next innings begins. The length of an innings is defined in terms of balls. A ball entails a bowler bowling—i.e., 'throwing' the ball—at the batsman, who is also known as the striker.

Batsmen score points, known as runs, by hitting the ball and running from one end to the other. If both the batsmen successfully cross to the other end of the pitch, the batting team is rewarded with one run. Batsmen can run between the ends as many times as possible and every successful crossing is given a run.

In addition to running, batsmen can score runs by hitting the ball out of the boundary line, which encircles the ground. If the ball touches the ground before going over the boundary line, then the batting team is given four runs. If the ball crosses the boundary line without touching the ground, then six runs are awarded.

The winner of a game is decided based on who scores more runs. Thus, teams try to outscore each other by scoring as many runs as they can.

All eleven players in a team are allowed to bat. A batsman is no more allowed to bat once they are dismissed, which is also known as losing the wicket. Some of the common forms of dismissals are bowled, caught, and LBW.

At any point in the game, two batsmen are supposed to be on the field, and thus, once when ten batsmen are dismissed, the innings of the batting team comes to an end. The innings is also be brought to an end once the bowling team bowls their allotted overs. Consequently, the aim of the bowlers is to either dismiss all 10 batsmen in a team or bowl their overs without giving away too many runs.

The batsmen, on the other hand, try to score as many runs as possible off the balls they face while making sure that they don't lose all their wickets before they face all of their allotted overs. However, trying to score as many runs as possible comes at the risk of losing wickets and this causes a dilemma for the batting team.

In order to save their wickets, a team can score slowly, i.e., not look to score as many runs as possible, but this will result in the team ending with a lower total of runs. A team that tries to score fast may end up losing all their ten wickets before they face all their overs.

For instance, when considering a T20 game, in which 20 overs are bowled, a team trying to score a run a ball, which doesn't involve too much risk, may end up facing all their 20 overs for 120 runs with the loss of three wickets. In contrast, a team trying to score fast may end up losing all their wickets in 10 overs while scoring 120 runs.

Even though in both scenarios, the same number of runs has been scored, two different resources have been left not utilized completely. In the first case, the team has 7 wickets remaining whereas in the second case, the team has 10 overs remaining. It could be argued that had, in the first case, the team been a little more aggressive, and in the second case, had they been more circumspect, they could have scored more runs.

Thus, striking the right balance between conserving the wickets and scoring runs becomes important to maximize the number of runs scored.

A cursory observation will reveal that, in cricket, based on the number of runs scored per ball (strike rate) and the number of balls faced per match on average, batsmen can be categorized into four quadrants as shown in Fig. 1.

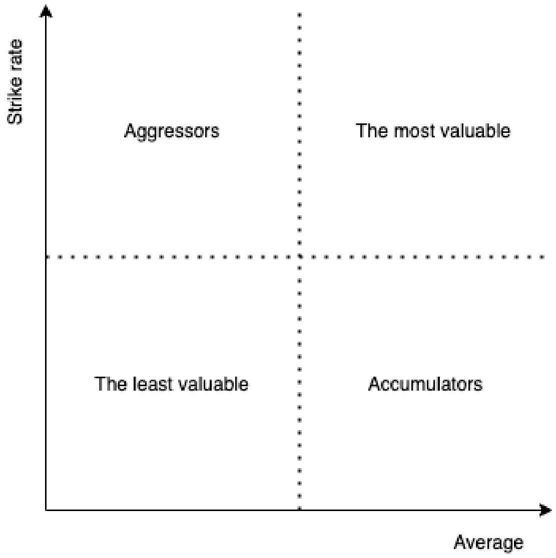

Fig. 1. The quadrants based on the strike rate and average of batsmen.

The batsmen with a higher strike rate and average, though rare, are the most valuable players. The batsmen with a lower strike rate and average are the least valuable. The batsmen with a higher strike rate and lower average—the aggressors—will help a team score faster but they will not last longer. The batsmen with a lower strike rate and a higher average—the accumulators—will not score faster but they will last longer.

Most batsmen belong to one of the latter two categories and teams have to choose the right combination of batsmen from the quadrants to maximize the runs they score. Too many aggressors in the team mean that the team may end up wasting balls and too many accumulators mean the team may end up wasting their wickets.

Mostly, teams play seven batsmen, those who specialize in batting, and five bowlers, those who specialize in bowling. Choosing seven batsmen in a certain order from a pool of 15 batsmen, thus, becomes a permutation problem. In such a case,

$$\frac{15!}{(15-7)!} = 3.24324 \times 10^{7}$$

One can come up with 3.24324E+7 permutations. Finding the optimal order among these permutations is a computationally expensive task. Thus, a heuristic way should be found to find a better batting order.

This study proposes the use of genetic algorithm to find a batting order in a heuristic manner from the large state space.

B. Genetic Algorithm

Genetic Algorithm is a heuristic search algorithm that was inspired by the biological process of evolution [4]. This algorithm is best used for optimization problems and is not guaranteed to produce the optimal solution. Thus, this algorithm is more useful to solve computationally complex problems that cannot be solved by traditional search algorithms. The solution produced, while not being theoretically the best solution, is usually the most practical solution given constraints such as time, space, and money.

Like evolution in nature, the genetic algorithm consists of genes as the fundamental units. These genes are the units that are arranged in different ways to produce a solution. Multiple genes are combined to produce a chromosome. A chromosome carries a probable solution, the quality of which is computed to make the process of searching more optimized.

For instance, if we are to formulate a diet plan taking into consideration the total calorie value of the diet and its price, we will have to come up with different combinations of food items to produce an optimized diet plan. In such a case, a gene would be a food item. A combination of food items would be a chromosome.

Multiple chromosomes are called a population. This population undergoes crossover, which is akin to mating in nature, mutation, and parent and survivor selection, which are analogous to natural selection, to produce a new generation of chromosomes of better quality.

This process can be repeated many times until the iterations no more result in an increase in the quality of the chromosomes. Consequently, an optimized solution to a problem is found.

Genetic algorithm, usually, consists of the following steps [5]:

- Initialization of the population
- Fitness value calculation
- Parent selection
- Crossover
- Mutation
- Survivor selection

First, chromosomes are randomly or heuristically produced to form the initial population. The chromosomes in the population are, then, assigned a fitness score using a fitness function. The fitness score determines the quality of the chromosome.

Then, from the population, a selected few chromosomes are chosen to produce the next generation of chromosomes. Various strategies such as tournament selection, fitness proportionate selection, roulette wheel selection, and rank selection are employed to carry out this process called parent selection.

Once the parents are selected, the parent chromosomes are crossed with one another to produce child chromosomes. This process, called the crossover, involves combining different parts of a pair of chromosomes together to produce a new chromosome called the child chromosome. Different technics such as uniform crossover, single point cross over, multi-point crossover, and ordered crossover are used to perform crossover.

Then, the chromosomes not subjected to crossover are subjected to mutation. A mutation is a random change made to a chromosome to reduce the chances of overfitting. Bit flip, swap, and random resetting are some of the techniques used to perform mutation.

Finally, the chromosomes that would survive to the next generation are chosen using strategies such as elitism, age-based selection, and fitness-based selection. This process is called survivor selection.

These steps are repeated over and over again until the fitness value of the population stops increasing, after which convergence is said to have taken place.

II. BACKGROUND

Studies on maximizing the runs scored by a team in limited-overs cricket have been scarce even though several studies have explored the possibility of selecting an entire cricket team using different technics including the use of genetic algorithm.

A study by S.M. Aqil Burney et al. proposed the use of genetic algorithm to select an entire team [6]. The number of games won and lost by each player was considered by the study to formulate the fitness function of the genetic algorithm.

The batting average, bowling average, batting strike rate, number of wickets per match, win-loss ratio, and experience were considered in the fitness function of the study carried out by Isuru Kusumsiri et al. to select the optimal One-Day-International cricket team using genetic algorithm [7].

Date Envelopment Analysis (DEA) was used to pick a cricket team factoring in the batting average, batting strike rate, bowling average, and bowling strike rate of players by a study done by Gholam R. Amin and Sujeet Kumar Sharma [8]. Shelvin Chand et al. used integer linear programming to construct a franchise team considering both the performance of the team and the overall cost of the players [9].

The research carried out by Harsha Perera et al. used an annealing algorithm to optimize a cricket team by considering the expected runs scored by and against the team using match simulation [10]. Simulated annealing was once again used by Tim B. Swartz et al. to generate an optimal batting lineup for an ODI using probabilities of every possible outcome of a ball [11].

The elitist non-dominated sorting genetic algorithm (NSGAII) was used by Faez Ahmed et al. to select a T20 cricket team taking into account the budget constraint [12]. The research work done by Matthews Oves et al. employed the Markov chain to model an optimal batting order for ODI matches by considering the probability of different outcomes for a ball [13].

Statistical simulation was combined with a genetic algorithm to optimize a batting lineup in baseball, a sport that closely resembles cricket, in a study by Sen Han [14]. A study by Abhijit R. Joshi et al. used genetic algorithm to construct an entire cricket team [15].

III. METHODOLOGY

For the purpose of this study, an optimized batting order consisting of seven batsmen from a pool of fifteen batsmen was generated using the Python-based evolutionary computation framework DEAP [16].

To encode the batting order into chromosomes, each batsman was assigned a unique integer ranging from 0 to 14 as shown in Table I. A gene was represented by an integer and a chromosome was formed by a list of fifteen such integers as depicted in Fig. 2.

TABLE I. ENCODING BATSMEN INTO GENES.

Index	**Batsmen**
0	Batsmen A
1	Batsmen B
2	Batsmen C
3	Batsmen D
4	Batsmen E
5	Batsmen F
6	Batsmen G
7	Batsmen H
8	Batsmen I
9	Batsmen J
10	Batsmen K
11	Batsmen L
12	Batsmen M
13	Batsmen N
14	Batsmen O

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Fig. 2. A chromosome

The fitness function calculated the number of runs the first seven batsmen of a chromosome could score. Only seven batsmen were considered since teams mostly play seven batsmen and four bowlers.

To calculate the number of runs a batting order is expected to score, the sum of the product of the average number of balls faced by a batsman and the average number of runs scored per ball by the batsman was found with the constraint that the total number of balls faced by batsmen didn’t exceed the allotted number of balls in a format, which is 120 for T20s and 300 for ODIs. To improve the performance of the algorithm, batsmen whose strike rate was below 1 and those who had played less than 6 innings were given no runs.

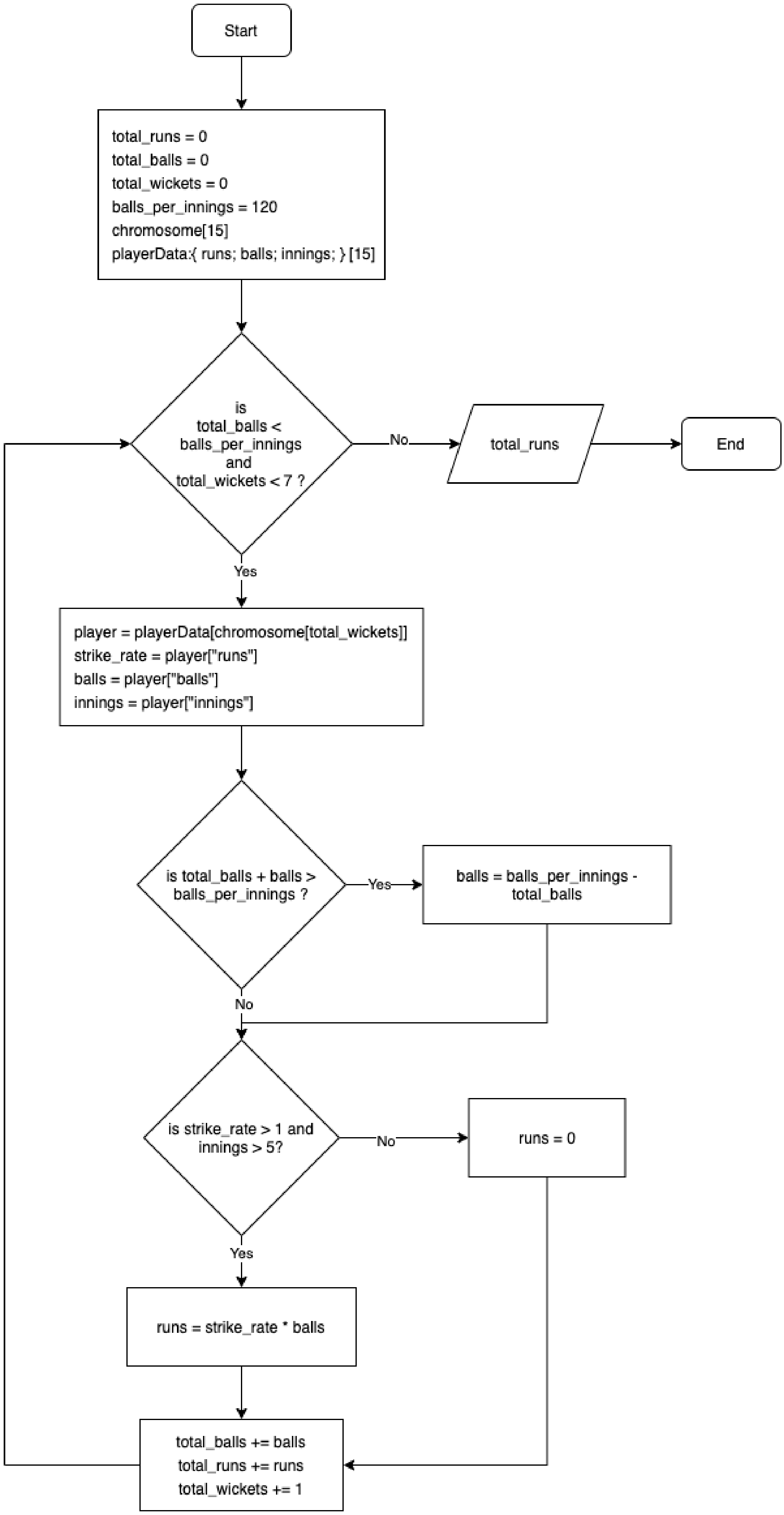

Fig. 3. A flowchart describing the fitness function.

The score calculated thus served as the fitness value of the chromosome with a higher value signifying more fitness. A flowchart describing the fitness function is provided in Fig. 3.

Tournament selection was used to select the parents for the crossover process with a tournament size set to 3. This was done to reduce selection pressure and thereby prevent premature convergence.

To perform crossover, ordered crossover was used since there can be no repetition when generating a batting order. The genes were shuffled to perform mutation as this helped avoid the repetition of genes as well.

The crossover probability was set to 0.5 and the mutation probability was set to 0.2. The chromosomes were randomly generated and the population size was set to 300. Since the population was found to converge by 100 generations, the algorithm was iterated over 100 generations.

Testing

To test the solution, the fifteen batsmen who had played the most number of balls batting first for their country from the top cricket-playing nations during the last five years (17 Nov 2015 - 17 Nov 2020) were considered. The data was obtained from ESPN Cricinfo's statsguru [17].

The optimized batting order for each of these teams was found using the above methodology and the maximum runs they could score was found from the fitness score. This score was compared against the average number of runs scored by each team while batting first. Only the runs scored while batting first were considered because while batting second, teams often adopt a batting approach that suits the target and in case of a successful chase, teams might not bat out their 120 balls.

IV. RESULTS

Table II shows the comparison between the average score of teams batting first, the optimized score obtained using the algorithm, and the percentage of improvement.

TABLE II. THE AVERAGE SCORES OF TEAMS BATTING FIRST AND THE OPTIMIZED SCORES

Team	Average score batting first	Optimized score	Improvement
England	170	176	3.53%
India	172	177	2.9%
Pakistan	161	169	4.97%
Ireland	157	169	7.64%
New Zealand	174	185	6.32%
Afghanistan	171	183	7.02%
South Africa	171	188	9.94%
Sri Lanka	145	148	2.07%
Bangladesh	156	160	2.56%
West Indies	150	165	10%
Australia	181	183	1.10%
Zimbabwe	147	158	7.48%

Table III shows the optimized batting order generated for each team by the algorithm. It is worth noting that, each iteration of the algorithm produced different batting orders even though the optimized score remained the same, which shows that different batting orders could be generated to produce the same optimized score.

TABLE III. THE BATTING ORDERS GENERATED BY THE ALGORITHM.

Team	Batting order
England	SW Billings, JM Bairstow, DJ Malan, JC Buttler, JJ Roy, EJG Morgan, JL Denly
India	V Kohli, RG Sharma, KL Rahul, HH Pandya, MK Pandey, SS Iyer, MS Dhoni
Pakistan	Shoaib Malik, Imad Wasim, Mohammad Hafeez, Umar Akmal, Babar Azam, Fakhar Zaman, Hussain Talat
Ireland	HT Tector, MR Adair, KJ OBrien, GJ Delany, A Balbirnie, SR Thompson, PR Stirling
New Zealand	MJ Guptill, C de Grandhomme, TL Seifert, CJ Anderson, C Munro, TC Bruce, MS Chapman
Afghanistan	Shafiqullah, Mohammad Nabi, Mohammad Shahzad, Najibullah Zadran, Hazratullah Zazai, Rahmanullah Gurbaz, Noor Ali Zadran
South Africa	F du Plessis, JP Duminy, AB de Villiers, HM Amla, DA Miller, H Klaasen, Q de Kock
Sri Lanka	AD Mathews, MD Gunathilaka, MDKJ Perera, N Dickwella, NLTC Perera, BKG Mendis, MD Shanaka
Bangladesh	Soumya Sarkar, Mahmudullah, Tamim Iqbal, Liton Das, Sabbir Rahman, Imrul Kayes, Afif Hossain
West Indies	SO Hetmyer, KA Pollard, N Pooran, E Lewis, D Ramdin, CAK Walton, LMP Simmons
Australia	DA Warner, MP Stoinis, TM Head, GJ Maxwell, AJ Finch, SPD Smith, AC Agar
Zimbabwe	E Chigumbura, H Masakadza, MN Waller, SC Williams, PJ Moor, RP Burl, CT Mutombodzi

V. DISCUSSION

The algorithm optimized the runs scored by the 12 major cricket-playing nations by 5.46% on average. The West Indies showed the biggest improvement of 10% whereas Australia showed the smallest of 1.1%. Further research needs to be done to study this discrepancy since, as shown in Fig. 4, no positive correlation can be found between the average percentage of matches played by players from the generated batting order, and the improvement shown. This could imply that the proposed order of batsmen had a bigger impact than the selection of the proposed batsmen on the improvement but further research is needed before such conclusions can be drawn.

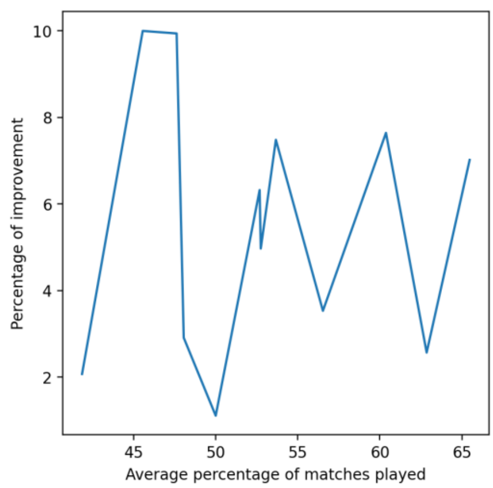

Fig. 4. Average percentage of matches played by the generated batting order vs. the percentage of improvement.

VI. CONCLUSION

This research demonstrated the efficacy of genetic algorithm in finding an optimized batting lineup to maximize the runs scored in an innings in cricket, factoring in runs scored per ball and the average number of balls faced by batsmen. Tests performed based on the data obtained of the performance of the top cricketing nations in T20Is between 17 Nov 2015 and 17 Nov 2020 showed that genetic algorithm could improve the runs they score by 5.46% on average. The algorithm produced different batting lineups each time it was run even though the optimized score remained the same, which means that different batting lineups could end up scoring the maximum possible runs. This research could further be improved by factoring in the performance of batsmen during different phases of an innings and their performance against individual bowlers.

VII. FUTURE WORK

Even though this study demonstrates theoretical improvement to the runs scored, real match performance is contingent on the skillsets of the batsmen. For instance, a batsman who habitually opens for his team and has a higher strike rate may be manifesting such a statistic due to the fielding restrictions during the first six overs of an innings. One cannot expect such batsmen to score at the same rate lower down the order.

Since the fitness function of this algorithm doesn't take into account the performance of batsmen during different phases of an innings such as the powerplay (overs 1-6), middle overs (overs 7-15), and death overs (overs 16-20), the optimized batting order could have batsmen excelling during the powerplay bat during the death overs. This is well observable in the batting lineups obtained by this study. For example, Q de Kock, who opens for South Africa, has been slotted in at number 7 by the algorithm.

This research could be further improved by factoring in the phase-wise performance of batsmen. However, obtaining phase-wise performance of batsmen is a challenge owing to the lack of availability of openly accessible data.

Moreover, batsmen's performance has also been shown to vary depending on the type of bowlers they face. Hence, optimal batting lineups can be generated for every team a team is likely to play against. Such a study would require head-to-head data between bowlers and batsmen, and obtaining such data is also a major challenge.

REFERENCES

[1] H.S. Altham and E.W. Swanton, *A History of Cricket*. London: Allen & Unwin, 1948, pp. 15-22.

[2] S. Ray, *Management of the Cricketing Ecosystem*. Singapore: Springer, 2022, pp. 119-140.

[3] S. Katoch, S.S. Chauhan and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Application*, vol. 80, no. 5, pp. 8091-8126, 2021. DOI: 10.1007/s11042-020-10139-6.

[4] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, pp. 65-85, 1994. DOI: 10.1007/BF00175354.

[5] S.M. Burney, N. Mahmood, K. Rizwan and U. Amjad, "A Generic Approach for Team Selection in Multiplayer Games using Genetic Algorithm," *International Journal of Computer Applications*, vol. 40, no. 17, pp. 11–17, 2012. DOI: 10.5120/5071-7440.

[6] I. Kusumsiri and S. Perera, "Optimal One Day International Cricket Team Selection by Genetic Algorithm," *International Journal of Sciences: Basic and Applied Research*, vol. 36, no. 4, pp. 213-221, 2017.

[7] G.R. Amin, and S.K. Sharma, "Cricket team selection using data envelopment analysis," *European Journal of Sport Science*, vol. 14, no. 1, pp. 369-376, 2014. DOI: 10.1080/17461391.2012.705333.

[8] S. Chand, H.K. Singh and T. Ray, "Team Selection Using Multi-/Many-Objective Optimization with Integer Linear Programming," *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2018. DOI: 10.1109/CEC.2018.8477945.

[9] H. Perera, J. Davis, and T.B. Swartz, "Optimal lineups in Twenty20 cricket," *Journal of Statistical Computation and Simulation*, vol. 86, no. 14, pp. 2888–2900, 2016. DOI: 10.1080/00949655.2015.1136629.

[10] T.B. Swartz, P.S. Gill, D. Beaudoin and B.M. deSilva, "Optimal batting orders in one-day cricket," *Computers and Operations Research*, vol. 33, no. 7, pp. 1939–1950, 2006. DOI: 10.1016/j.cor.2004.09.031.

[11] F. Ahmed, A. Jindal and K. Deb, "Cricket Team Selection Using Evolutionary Multi-objective Optimization," *Swarm, Evolutionary, and Memetic Computing (SEMCCO)*, pp. 71-78, 2011. DOI: 10.1007/978-3-642-27242-4_9.

[12] M. Ovens and B. Bukiet, "A mathematical modelling approach to one-day cricket batting orders," *Journal of Sports Science and Medicine*, vol. 5, no. 4, pp. 495–502, 2006.

[13] S. Han, "Batting order optimization by genetic algorithm," *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation (GECCO '12)*, pp. 599–602, 2012. DOI: 10.1145/2330784.2330882.

[14] A.R. Joshi, P.P Raut, V. Desai, H. Doshi and K. Shah, "Cricket Team Selection Using Artificial Intelligence," *Proceedings of International Conference on Advances in Communication and Computing Technologies (ICACACT)*, pp. 27, 2013.

[15] F.A. Fortin, F.M. De Rainville, M.A. Gardner, M. Parizeau and C. Gagné, "DEAP: Evolutionary Algorithms Made Easy," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2171–2175, 2012.

[16] *Statsguru*. [Online]. Available: https://stats.espncricinfo.com/ci/engine/stats/index.html