A Balanced Squad for Indian Premier League Using Modified NSGA-II

SHANU VERMA1, VIVEKANAND PANDEY1, MILLIE PANT1,2, AND VAČLAV SNASEL3, (Senior Member, IEEE)

1Department of Applied Mathematics and Scientific Computing, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India
2Mehta Family School of Data Science and Artificial Intelligence, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India
3Department of Computer Science, VSB—Technical University of Ostrava, 70800 Ostrava, Czech Republic

Corresponding author: Millie Pant (pant.milli@as.iitr.ac.in)

This work was supported in part by the Council of Scientific and Industrial Research (CSIR), Ministry of Science and Technology, Government of India, under Grant 09/143(0966)/2019-EMR-I; and in part by the Metaheuristics Framework for Combinatorial Optimization Problems (META-MO-COPS) under Grant DST/INT/Czech/P-12/2019.

ABSTRACT Selecting team players is a crucial and challenging task demanding a considerable amount of thinking and hard work by the selectors. The present study formulated the selection of an IPL squad as a multi-objective optimization problem with the objectives of maximizing the batting and bowling performance of the squad, in which a player’s performance is estimated using an efficient Batting Performance Factor and Combined Bowling Rate. Also, the proposed model tries to formulate a balanced squad by constraining the number of pure batters, pure bowlers, and all-rounders. Bounds are also considered on star players to enhance the performance of the squad and also from the income prospects of IPL. The problem in itself is treated as a 0/1 knapsack problem for which two combinatorial optimization algorithms, namely, BNSGA-II and INSGA-II, are developed. These algorithms were compared with existing modified NSGA-II for IPL team selection and three other popular multi-objective optimization algorithms, NSGA-II, NSDE, and MOPSO-CD, on the basis of standard performance metrics: hypervolume, inverted generational distance, and number of Pareto optimal solutions. Both algorithms performed well, with BNSGA-II performing better than all the other algorithms considered in this study. The IPL 2020 players’ data validated the applicability of the proposed model and algorithms. The trade-off squads contained players of each expertise in appropriate proportions. Further analysis of the trade-off squads demonstrated that many theoretically selected players performed well in IPL 2020 matches.

INDEX TERMS Cricket, twenty20, knapsack problem, combinatorial, Indian premier league, multi-objective, optimization, squad selection.

I. INTRODUCTION
The Indian premier league (IPL) is a franchise-based competition started in 2008 by the Board of control for cricket in India (BCCI) to promote cricket in India [1]. IPL is the most attended cricket league and has a significant contribution to the social and economic sectors. In 2015, IPL contributed ₹11.5 B to the Indian economy.

In this league, the players are bought by high-profile owners through an auction process. Each franchise owner spends a significant amount of money on purchasing players that have the ability to win and which also fit their business objectives. Considering the huge amount of money invested in the purchase of players, the franchise owners try to explore the best possible combinations of players for their squad.

In view of this, the researchers in [2] have formulated IPL team selection as a bi-objective optimization problem with the aim to maximize batting and bowling performances and satisfy the IPL regulations: budget constraint, bound on the
number of wicket-keepers, captains, and overseas players. Similar objectives and constraints can be used to choose players for an IPL squad. However, since an all-rounder contributes to both batting and bowling performances while a pure batter (pure bowler) primarily contributes to squad batting performance (bowling performance), the trade-off squads for this problem, obtained using an optimization algorithm mainly contain all-rounders, which is a feasible solution theoretically but not a practical one. Keeping this drawback in mind, a new model is proposed in the present study in order to choose a squad for the IPL which is more balanced. The proposed model has additional bounds on the number of players based on their expertise, ensuring a sufficient proportion of pure batters, pure bowlers and all-rounders in a squad.

Further, only a player’s batting average (batting strike rate) was used in the existing multi-objective team selection models in [2] and [3] to determine its batting performance (bowling performance). However, in IPL, maintaining a high strike rate is the primary focus of a batter because, in a short cricket format, scoring runs rapidly is more important for a team’s success than the average run per innings. On the other hand, the efforts of bowlers with a good economy rate (bowlers who prevent the batters from scoring runs) and the pressure to maintain a high strike rate increase the chances of batters being dismissed. Therefore, we use more efficient measures to better estimate a player’s performance: the batting performance factor [4] and the combined bowling rate [5]. These measures also include the batting strike rate (bowling strike rate and bowling economy) along with the batting average (bowling average) for computing the batting (bowling performance) of a batter (bowler).

A trade-off squad may contain many players who have participated only in a few matches because these players have strong statistics and are less expensive than more experienced players. Instead, each squad should have some star players who not only contribute to the success of the franchise through their performance but also through their popularity and fame. Therefore, we put bounds on the minimum number of star batters, star bowlers, and star all-rounders so that the squad contains a sufficient number of star players within the same budget. Additionally, we use data from all the franchise leagues and T20 international matches to more accurately estimate a player’s performance.

The Knapsack problems are among the most studied combinatorial optimization problems (COPs), finding solutions to which remain an exciting challenge [6]. The proposed IPL squad selection problem can be viewed as a knapsack problem with the objective to choose limited players from available such that the total performance of the squad is maximized under the fixed budget. In IPL 2020, there are \(10^{27}\) to \(10^{30}\) possible combinations for selecting a squad of 21 to 25 players out of 185 auctioned players. Such a large number of combinations makes the selection of players a challenging COP.

Various optimization methods are available in the literature for cricket team selection: genetic algorithm for selecting an optimal national cricket team [7] and integer programming for selecting playing XI team [8], T20 world cup team [9], and IPL team [3]. The multi-objective IPL team selection model has been solved using integer linear programming [3] and non-dominated sorting genetic algorithm-II (NSGA-II) [2], [10]. NSGA-II [11] is a population-based multi-objective optimization algorithm widely used to solve the multi-objective COPs [12], including multi-objective knapsack problems [13], [14].

The present study developed two NSGA-II variants, BNSGA-II and INSGA-II, for solving the proposed multi-objective IPL squad selection problem. BNSGA-II has a binary chromosome representation, and INSGA-II has an integer chromosome representation. The problem constraints are handled using the constraint dominance principle, and a repair mechanism is proposed to repair the infeasible solutions that occur in BNSGA-II due to the violation of a single constraint in order to produce a sufficient number of feasible solutions. The efficiency of the proposed algorithms is validated by comparing them with NSGA-II, an algorithm for the bi-objective IPL team selection problem [2] and three other popular multi-objective optimization algorithms: NSGA-II [11], non-dominated sorting differential evolution (NSDE) [15], and multi-objective particle swarm optimization with crowding distance (MOPSO-CD) [16], using hypervolume, inverted generational distance (IGD), number of Pareto optimal solutions (NPS), and computational time. Next, the trade-off squads obtained through the best-performed algorithm are analyzed based on their cost and the fielding performance. The performance of trade-off squad players in IPL 2020 validates the efficiency of the proposed model. A practical situation is also simulated where a franchise owner favours certain players irrespective of their cost.

In summary, the main contribution of this study is:

- Using the concept of knapsack problem for selecting an IPL team.
- Designing a new model for selecting a balanced squad for IPL.
- Developing two new algorithms, BNSGA-II and INSGA-II, for solving the proposed model.

The rest of this paper is organized as follows: Section 2 describes the related work. Section 3 defines the multi-objective knapsack problem. Section 4 formulates the proposed IPL Squad selection problem. Section 5 describes the proposed methodology. Section 6 discusses the experiment results and comparison. Section 7 shows the analysis and discussion. Finally, the conclusion is presented in Section 8.

II. RELATED WORK

This section discusses the optimization methods available in the literature to select IPL team/squad players. In addition to the optimization methods, researchers also used methods based on other techniques, such as data envelopment analysis [17], [18] and machine learning algorithms [19], [20].
However, this study is only focused on optimization methods for single and multi-objective optimization models. The comparative study between the related work on optimization methods for cricket team selection and the proposed work is given in Table 1.

Sathya and Jamal [7] proposed genetic algorithm for selecting eleven out of fifty national players for a one-day international (ODI). The factors such as the number of pacers and spinners, composition of left-hander and right-hander, and partnership records to select a more flexible, balanced, and diverse team had been considered for selection purposes.

Ahmed et al. [2] proposed NSGA-II with a novel gene representation and decision-making techniques for IPL T20 team selection. The resulting teams were compared with IPL 4th edition teams and found better theoretically. The authors also demonstrated a dynamic auction-based player selection to make the procedure more realistic.

Bhattacharjee and Saikia [9] selected an optimal squad of 15 players using binary integer programming and compared it with the Indian team selected by the International cricket council (ICC) for the corresponding T20 world cups. Again in 2016, authors [8] used binary integer programming to select the playing XI team, which took care of the area of specialization of the captain and other players.

Chand et al. [3] proposed an integer programming method for IPL team selection that guaranteed optimality and demonstrated its scalability using two-objective, three-objective, and five-objective formulations. The objectives of the problem were based on batting, bowling, fielding, cost, and star power. The construction of partial teams was done around the preferred players. In addition, the ranking of players was done using the players’ performance to aid the decision-making.

III. MULTI-OBJECTIVE KNAPSACK PROBLEM

Given a knapsack with weight capacity W and a set of n items in which ith item has weight \( w_i \leq W \) and profit per objective \( v_{i1} > 0, i = 1, 2, \ldots, n \), then the multi-objective 0/1 knapsack problem aims to fill the knapsack with given items within its capacity such that the total profit is maximized. Mathematically, a K-objective knapsack problem is defined as follows:

\[
\text{Maximize} \quad \left( \sum_{i=1}^{n} v_{i1} x_i, \sum_{i=1}^{n} v_{i2} x_i, \ldots, \sum_{i=1}^{n} v_{iK} x_i \right) \\
\text{s.t.} \quad \sum_{i=1}^{n} w_i x_i \leq W \\
\text{where} \quad x_i = \begin{cases} 
1, & \text{item } i \text{ is selected} \\
0, & \text{item } i \text{ is not selected} 
\end{cases}, \quad i = 1, 2, \ldots, n
\]

IV. IPL SQUAD SELECTION PROBLEM

The IPL squad selection problem is similar to the above-defined multi-objective 0/1 knapsack problem, as illustrated in Table 2. The number of players to be selected for an IPL squad is limited; a franchise owner will either select a player (Yes = 1) or not (No = 0). Therefore, the concept of the 0/1 knapsack problem is used in this paper for selecting a player for an IPL squad.

A. PROBLEM DEFINITION AND FORMULATION

A balanced squad contains players of each expertise in suitable proportions. The proposed squad selection problem aims to form a balanced IPL squad with maximum net batting and bowling performance. The objectives and constraints of the problem are detailed in the following subsections. The variables and parameters used in the paper are defined in Table 3.

1) BATTLING PERFORMANCE

The batting average\(^1\) indicates the run-scoring capability of a player and estimates its batting performance. However, for limited-overs cricket, slow batting will lead to defeat rather than victory [4]; therefore, players with high batting strike rate\(^2\) are also needed since it indicates the rate of scoring runs by a player.

It is not necessary that the players with high batting averages also have high strike rates; for example, the IPL 2020 auctioned players with high strike rates and low batting averages are shown in Table 4. The relation between these batting statistics can further be visualized using Fig. 1. Here, the value of these statistics for pure batters and batting all-rounders is normalized (due to much difference in their scale) and sorted in decreasing order of the batting averages. The figure shows that the batting averages of the players decrease continuously, but their strike rates fluctuate. Also, they do not have any relation such that considering one will count both of them. Therefore, both factors should be considered important for analyzing a player’s batting performance.

\[\text{Batting Average} = \frac{\text{Total runs scored}}{\text{Number of times out}}\]

\[\text{Batting Strike Rate} = 100 \times \left( \frac{\text{Total runs scored}}{\text{Total balls faced}} \right)\]

Researchers in [4] have proposed an efficient Batting performance factor (BPF) for calculating the player’s batting performance. BPF is the weighted product of the normalized values of batting strike rates and batting averages, as given in (1). The higher value of \( \alpha \) indicates a higher weightage to

\[\text{Normalized values of} \\
\text{Batting Statistics of Players} \\
\text{Auctioned players in IPL 2020} \]

**FIGURE 1. Relation between batting statistics of auctioned players in IPL 2020.**

\(^1\)Batting average = Total runs scored/Number of times out.

\(^2\)Batting strike rate = 100 × (Total runs scored/Total balls faced).
batting strike rate than the batting average. Since this factor utilizes both batting statistics and one statistic can be prioritized over the other, we have used this factor to calculate the batting performance. The higher value of BPF indicates the higher batting performance of a player.

\[ BPF = \text{Batting strike rate}^\alpha \times \text{Batting average}^{1-\alpha} \quad (1) \]

where \( 0 \leq \alpha \leq 1 \)

The first objective of this study is to maximize the net batting performance (or minimize the –net batting performance) of the squad as given in (2). Here, the objective function \( f_1 \) denotes the negative of the sum of BPF factors of selected players.

\[ \text{Minimize } f_1 = - \sum_{i=1}^{n} x(i) \times BPF(i) \quad (2) \]

TABLE 1. Comparison of the related work and the proposed work.

| Year | Goal | Constraints | Batting & Bowling Statistics | Method | Format | Data | Number of objectives | Ref. |
|------|------|-------------|------------------------------|--------|--------|-----|---------------------|-----|
| 2009 | Selection of 11 players out of 50 national players | - | ICC rankings, statistics & cricket experts. | Genetic Algorithm | ODI | NA | Single | [7] |
| 2013 | IPL team selection | Captain, wicket-keeper, overseas player and budget | Batting, bowling, & fielding averages | NSGA-II = knee region approach | IPL | IPL 4th edition | Multi | [2] |
| 2014 | T20 world cup team selection | Openers, Middle-order batters, spin bowlers, fast bowlers, all-rounders, wicket-keeper | Batman’s average & strike rate; Bowler’s economy, average & strike rate | Binary Integer Programming | IPL | IPL 2009, 2010 & 2012 | Single | [9] |
| 2016 | Selection of playing XI team | Openers, Middle-order batters, spin bowlers, fast bowlers, all-rounders, wicket-keeper | Batman’s average & strike rate; Bowler’s economy, average & strike rate | Binary Integer Programming | IPL | IPL 5th edition | Single | [8] |
| 2018 | IPL team selection | Captain, wicket-keeper, overseas player, budget, bowling average, batting average, fielding average, star-power | Batting, bowling, & fielding averages | Integer Linear Programming | IPL | IPL 4th edition | Multi / Many | [3] |
| Proposed work | Selection of balanced squad for IPL (23 out of 185 players) | Captain, wicket-keeper, overseas player, budget, pure bowlers, pure bowlers, all-rounders, star batters, star bowlers, star all-rounders | Batting Performance Factor [4] (Batman’s average and strike rate); Combined Bowling Rate [5] (Bowler’s economy, average and strike rate) | Modified NSGA-II: BNSGA-II and INSGA-II | IPL | Franchise leagues & T20 international matches data up to August 2020 | Multi | - |

TABLE 2. Similarity between knapsack problem and IPL squad selection problem.

| Knapsack problem | IPL squad selection problem |
|------------------|----------------------------|
| Knapsack         | Squad                      |
| Item             | Player                     |
| Item’s weight    | Auctioned price of the player |
| Item’s profit    | Performance of the player  |
| Knapsack capacity | Maximum salary cap          |

TABLE 3. Notations.

| Notations | Definitions |
|-----------|-------------|
| \( m \)  | Total number of players |
| \( N \)  | Number of players in a squad |
| \( T \)  | Maximum salary cap |
| \( a_1 \) | Lower bound for the sum of the number of pure batters and batting all-rounders in a squad |
| \( a_2 \) | Lower bound for the sum of the number of pure bowlers and bowling all-rounders in a squad |
| \( m_1 \) | Lower bound for the number of pure batters |
| \( m_2 \) | Lower bound for the number of pure bowlers |
| \( m_3 \) | Upper bound for the number of all-rounders |
| \( n_1 \) | Lower bound for the number of star pure batters |
| \( n_2 \) | Lower bound for the number of star pure bowlers |
| \( n_3 \) | Lower bound for the number of star all-rounders |
| \( BPF \) | Batting performance factor |
| \( CBR \) | Combined bowling rate |
| \( \alpha \) | Weight \( \in (0,1) \) |
| \( P_m \)  | Mutation probability |
| \( P_c \)  | Crossover probability |

\[ x(i) = \begin{cases} 1, & \text{if player } i \text{ is chosen for the squad} \\ 0, & \text{otherwise} \end{cases} \]
| Players          | Strike rate (Max Value = 175.76) | Batting average (Max value = 64.44) |
|-----------------|---------------------------------|-----------------------------------|
| Andre Russell   | 171.48                          | 26.72                             |
| Krishnappa Gowtham | 166.67                       | 15.47                             |
| Fabian Allen    | 164.90                          | 25.25                             |
| Harshal Patel   | 153.82                          | 17.81                             |
| Sunil Narine    | 147.05                          | 14.84                             |

Here, $n$ is the total number of players, and $BPF(i)$ is the batting performance factor of the $i^{th}$ player, $i = 1, 2, \ldots, n$.

2) BOWLING PERFORMANCE

The bowling average,\(^3\) bowling strike rate,\(^4\) and bowling economy\(^5\) are the three factors that analyze a bowler’s performance. In T20, due to a limited number of overs, pure batters usually need to risk their wicket to score more runs because of which bowlers with an excellent economy rate (i.e., bowlers that do not provide runs to batters) produce wickets for other bowlers.

The relationship between bowling statistics is analyzed by normalizing and sorting them according to bowling averages, as shown in Fig. 2. The figure illustrates that a bowler with a relatively higher bowling average does not necessarily have a relatively higher economy rate and bowling strike rate.

![FIGURE 2. Relation between bowling statistics of auctioned players in IPL 2020.](image)

The second objective of this study is to maximize the net bowling performance (or minimize the -net bowling performance) of the squad as given in (4). Here, the objective function $f_2$ denotes the sum of the CBRs of selected players.

$$\text{Minimize } f_2 = \sum_{i=1}^{n} x(i) \times CBR(i) \quad (4)$$

where $x(i) = \begin{cases} 1, & \text{if player } i \text{ is chosen for the squad} \\ 0, & \text{otherwise} \end{cases}$

Here, $n$ is the total number of players, and $CBR(i)$ is the combined bowling rate of the $i^{th}$ player, $i = 1, 2, \ldots, n$.

Note: Since a lower $CBR$ is favourable, batters will be selected as a bowler due to their lower $CBR$ and not because of their excellent batting statistics. Therefore, the $CBR$ of a batter is taken 100 as a penalty.

3) CONSTRAINTS BASED ON IPL REGULATIONS

BCCI puts some constraints (5-9) on franchises for selecting players of an IPL squad. Constraint (5) limits the salary cap of franchises for purchasing players. Constraints (6) and (7) bounds the total number of players and the number of overseas players in a squad, respectively. Here $m$ is the number of players in a squad, and $T$ is the maximum salary cap. The other regulations (8-9) include the restrictions on the number of captains and wicket-keepers in the squad.

$$\text{Total cost of squad} \leq T \quad (5)$$
$$\text{Number of players in a squad} = m \quad (6)$$
$$\text{Number of overseas players} \leq 8 \quad (7)$$
$$\text{Number of captains} \geq 1 \quad (8)$$
$$\text{Number of wicketkeepers} \geq 1 \quad (9)$$

4) BOUND ON THE SUM OF THE NUMBER OF PURE BATTERS (BOWLERS) AND BATTING (BOWLING) ALL-ROUNDERS

The trade-off squads created by applying the above IPL regulations may contain mostly all-rounders as they improve both the batting and bowling performance of a squad. Also, due to budget limitation, bowlers, bowling all-rounders, and new players have a higher chance of being selected as these players have lower auction prices compared to pure batters and experienced players. Therefore, two additional constraints are considered, one bounding the sum of the number of pure batters and batting all-rounders (10) and the other bounding the sum of the number of bowlers and bowling all-rounders (11).

$$\text{Sum of number of batters} \& \text{batting all} \rightarrow \text{rounders} \geq a_1 \quad (10)$$
$$\text{Sum of number of bowlers} \& \text{bowling all} \rightarrow \text{rounders} \geq a_2 \quad (11)$$

Here, $a_1$ is the lower bound for the sum of the number of batters and batting all-rounders and $a_2$ is the lower bound for the sum of the number of bowlers and bowling all-rounders in a squad.
5) BOUND ON THE NUMBER OF PURE BATTERS, PURE BOWLERS, ALL-ROUNDERS, STAR BATTERS, STAR BOWLERS, AND STAR ALL-ROUNDERS

As the pure batters have higher CBR (as a penalty) than batting all-rounders, the trade-off squads still have the possibility of a shortage of batters. Therefore, instead of constraints (10-11), bounds are considered on the minimum number of pure batters (12), pure bowlers (13) and all-rounders (14).

\[
\begin{align*}
\text{Number of pure batters} & \geq m_1 \quad (12) \\
\text{Number of pure bowlers} & \geq m_2 \quad (13) \\
\text{Number of all - rounders} & \leq m_3 \quad (14)
\end{align*}
\]

where \(m_1\) is the lower bound for the number of pure batters, \(m_2\) is the lower bound for the number of bowlers, and \(m_3\) is the upper bound for the number of all-rounders in a squad.

The above three constraints can be used to obtain a balanced squad. However, every franchise wants to have a sufficient number of star players\(^6\) because a star player increases the squad’s performance and draws public attention owing to its outstanding performance. Consequently, we have considered the bounds on the number of star players also (15-17).

\[
\begin{align*}
\text{Number of star pure batters} & \geq n_1 \quad (15) \\
\text{Number of star pure bowlers} & \geq n_2 \quad (16) \\
\text{Number of star all - rounders} & \geq n_3 \quad (17)
\end{align*}
\]

where \(n_1\), \(n_2\) and \(n_3\) are respectively the lower bounds for the number of star pure batters, star pure bowlers, and star all-rounders in a squad.

B. DATASET

The proposed model is validated using the performances of IPL 2020 auctioned players in T20 tournaments, franchise leagues, and international matches. The data of 185 players up to August 2020 is taken from the website http://bigbashboard.com (a well-known web forum containing data on all cricket matches).

The details of the players are given in Table 5. The star players are subjectively considered based on the players’ records in T20I and IPL. A star batter is a player who has either scored \(\geq 1,000\) runs in T20I (i.e., T20 International) or has scored \(\geq 2,000\) runs in IPL. Likewise, a star bowler is a player who has taken either \(\geq 30\) wickets in T20I or \(\geq 60\) wickets in IPL. Further, for being a star all-rounder, a player should have scored \(\geq 300\) runs and taken \(\geq 15\) wickets in T20I or \(\geq 600\) runs and taken \(\geq 30\) wickets in IPL. The total number of star players is 70, out of which 24 are pure batters, 26 are pure bowlers, and 20 are all-rounders.

C. PROBLEM COMPLEXITY

The complexity of the proposed COP can be defined as the possible number of players combinations for a squad, \(c^n_m\), where \(n\) is the total number of selected players and \(m\) is the maximum number of players in a squad. The complexity varies with the number of players in the squad. For, \(m = 23\) and \(n = 185\), it is \(\sim 10^{29}\).

TABLE 5. Details of Auctioned players in IPL 2020.

| Role                      | Number of players |
|---------------------------|-------------------|
| Pure Batter               | 64                |
| Pure Bowler               | 74                |
| Wicket-keeper             | 24                |
| All-rounder               | 47                |
| Batting all-rounder       | 34                |
| Bowling all-rounder       | 33                |
| Captain                   | 23                |

V. PROPOSED ALGORITHMS

NSGA-II, introduced by Deb et al. [11], is a popular optimization algorithm capable of solving difficult multi-objective optimization problems. The present study develops two NSGA-II variants to solve the proposed IPL squad selection problem. The following are the basic operations of the proposed algorithms:

A. CHROMOSOME REPRESENTATION

The performance and functionality of an algorithm are affected by a solution representation. Therefore, two efficient solution representations, binary and integer chromosome representations, are defined to solve the proposed knapsack problem, as shown in Figures 3 and 4. Both of these representations denote a squad with \(m\) number of players. In Fig. 3, the player tag value corresponding to gene value ‘1’ represents the index of the player selected for the squad. This representation maintains a unique combination of genes for each squad; no operations cause the re-occurrence of players in a squad. In Fig. 4, the gene values corresponding to player tags represent the indices of the selected players.

![Figure 3. Binary chromosome representation.](image)

![Figure 4. Integer chromosome representation.](image)
B. POPULATION INITIALIZATION
In BNSGA-II, the initial population is randomly generated as binary chromosomes with \( m \) number of 1s. The initial population in INSGA-II is also randomly generated as integer chromosomes having \( m \) unique values out of \( n \).

C. CROSSOVER AND MUTATION OPERATORS
In BNSGA-II, the two-point crossover and swap mutation are used, as shown in Fig. 5. The two-point crossover exchanges the genes between two random crossover points \( k_1 \) and \( k_2 \) of the parent chromosomes to generate the offspring chromosomes, where \( k_1 \) and \( k_2 \) are random integers in \([1, \lfloor n/2 \rfloor]\) and \([k_1 + 1, n]\), respectively. Similarly, the swap mutation exchanges two random genes of the offspring population.

D. CONSTRAINT HANDLING
The constraint dominance principle [11] is used to take care of the constraints. According to this principle, if there are two solutions \( x^1 \) and \( x^2 \), then the selection is made as per the following three possibilities:
1) If both \( x^1 \) and \( x^2 \) are feasible, where one that dominates the other is selected.
2) If one solution is infeasible, then the feasible solution is considered.
3) If both \( x^1 \) and \( x^2 \) are infeasible, then the one with a lesser constraint violation is said to dominate the other and is selected.

E. REPAIR MECHANISM
In BNSGA-II, initially, each squad has \( m \) number of players, but after the first generation, the constraint (6) of maintaining the fixed number of players in each squad starts violating. Due to this constraint, the infeasibility in the population results in fewer feasible population members despite taking a large population size. Therefore, a repair mechanism is proposed to repair the offspring population in each generation to produce sufficient feasible population members for the next generation. More clearly, the repair mechanism helps remove the large infeasibility due to a single constraint. The pseudocode for the repair mechanism is given in Algorithm 1.

Algorithm 1 Repair Mechanism

Parameters: Mutated chromosome \((x)\), Number of variables \((n)\), Squad length \((m)\)

1: Find the number of non-zero elements in \( x \) and their indices set: \( N \) and \( \text{Index} \)
2: IF \( N > m \), then
   \( x(\text{Index}) = 0 \)
   Randomly select \( m \) indices from the set \( \text{Index} \): \( \text{Index1} \)
   \( x(\text{Index1}) = 1 \)
ELSE IF \( N < m \), then
   Take the compliment of the set \( \text{Index} \): \( \text{Index2} \)
   Randomly select \( m-N \) indices from \( \text{Index2} \): \( \text{Index3} \)
   \( x(\text{Index3}) = 1 \)
END IF

A brief summary of BNSGA-II and INSGA-II operators is provided in Table 6. The other procedures, including the non-dominating sorting, crowding distance calculation, and selection procedures, are the same as in conventional NSGA-II [11]. The algorithmic steps of the BNSGA-II are shown in Algorithm 2.

VI. EXPERIMENTAL RESULTS AND COMPARISONS
This section designs experiments considering two cases for IPL squad selection. Case-I is an optimization problem with objectives (2) and (4) subject to constraints (5-11), and case-II is the optimization problem with objectives (2) and (4) subject to constraints (5-9) and (12-17). The first experiment in which a random gene (a random player) is replaced with a gene different from the chromosome genes (player not present in the squad).
TABLE 6. Proposed algorithms.

| Operators                  | BNSGA-II    | INSGA-II    |
|----------------------------|-------------|-------------|
| Chromosome representation  | Binary      | Integer     |
| Crossover                  | Two-point   | Partially   |
|                           | crossover   | mapped      |
|                           | Swap mutation| Random      |
| Mutation                  | Constraint  | Constraint  |
|                           | dominance   | dominance   |
|                           | principle + Repair mechanism | principle |

validates the efficiency of the proposed algorithms and the second experiment compares the cases for a balanced squad. The setup for the experiments and the results are detailed in the following subsections.

A. EXPERIMENTAL SETUP
This section introduces the other algorithms used for comparison, parameter settings, and performance metrics.

1) ALGORITHMS FOR COMPARISON
The efficiency of BNSGA-II and INSGA-II is validated using four prominent algorithms. One of them is the modified NSGA-II for the bi-objective IPL team selection problem (naming convention used in this study: ‘MNSGA-II’) [2], which used a chromosome representation with fixed positions of the captain and wicket-keeper. MNSGA-II was originally coded for an 11 players team, and the current study extends it to a squad of 23 players.

2) PARAMETER SETTING
The parameters for the proposed model and algorithms are given in Table 7. The population size and the number of generations are taken after fine-tuning the algorithm. Several combinations were tried to optimize the performance of the algorithm. It was observed that the population size of 200 and the stopping criteria of 500 generations gave the best possible results. Likewise, the other parameters were also fine-tuned so that the Pareto optimal solutions cover the whole Pareto front and with good convergence. Considering the brevity of space only, the best parameters are shown in the study.

Algorithm 2 BNSGA-II
Algorithm parameters: Population size (pop_size), Number of variables (n), Maximum number of generations (max_gen), Mutation probability (pm), Crossover probability (pc)
Problem parameters: chromosome (x), Squad length (m)
//Population initialization
1: FOR i = 1: pop_size
1.1: Generate a matrix [A]_{1 \times m} with unique elements a\_i \in \{1, 2, \ldots, n\}
1.2: FOR j = 1: n
   IF j \in A
   x(i, j) = 1
   ELSE
   x(i, j) = 0
   END IF
END FOR
END FOR
// Generating next generations P, t=2, 3, ..., max_gen
2: FOR t=1: max_gen
2.1: Perform two-point crossover on P\_t using p\_c
2.2: Perform swap mutation and generate the offspring population using
2.3: Perform the repair mechanism on the offspring population
   using p\_c
2.4: Combine parent population and repaired offspring population
2.5: Put the combined population into different non-dominated fronts
   F = \{F_1, F_2, \ldots, \} using non-dominated sorting procedure
2.6: Set the parent population P\_{t+1}=\varnothing, k=1
2.7: WHILE (P\_{t+1} \leq pop_size) DO
   IF (P\_{t+1} \cup F_k \leq pop_size) then
      Include the k-th non-dominated front F_k in the P\_{t+1}(P_{t+1}= P_{t+1} \cup F_k)
   ELSEIF (P\_{t+1} \cup F_k > pop_size) then
      Calculate the crowding distance of members of the front F_k
      Sort the members of F_k in decreasing order of crowding distance
      Add the top (pop_size - |P\_{t+1}|) members of F_k to P\_{t+1}
   END IF
END WHILE
END FOR
3: Output: A set of Pareto optimal solutions F_k

The model parameters \(a_1, a_2, m_1, m_2, m_3, n_1, n_2, n_3\) are fixed for the squad length \(m = 23\), as shown in Table 7 and can be changed with the value of \(m\). The value of parameter \(\alpha\) is set using sensitivity analysis as follows:

Sensitivity analysis of parameter \(\alpha\): The parameter \(\alpha\) calculates the batting performance of the squad by prioritizing one batting statistic over the other; therefore, a sensitivity analysis is performed to decide its value. For

FIGURE 7. Real chromosome representation.

The other popular multi-objective optimization algorithms that are used for comparison with the proposed algorithms are real-coded NSGA-II (naming convention used in this study: ‘RNSGA-II’)[11], NSDE [15], and MOPSO-CD [16]. The chromosome representation used for these three algorithms is given in Fig. 7. Here, the continuous representation is mapped to binary representation with the help of a rounding function. Suppose \((x_1, x_2, \ldots, x_n)\) is the chromosome with continuous representation; then it gets round to 0-1 representation \((b_1, b_2, \ldots, b_n)\) using (18). The rounding function selects \(m\) players (or nearly \(m\)) out of \(n\) players to satisfy the constraint (6).

\[ b_i = \begin{cases} 
1, & x_i \leq (m/n) \\
0, & \text{otherwise} 
\end{cases}, i = 1, 2, \ldots, n \quad (18) \]

Algorithm 1 repairs the infeasible chromosomes obtained using these three algorithms.
3) PERFORMANCE METRICS

The performance metrics NPS, hypervolume, and IGD are used for comparing the algorithms [22]. These are the most popular metrics for comparing the multi-objective optimization algorithms and are defined as:

- **NPS**
  
  \[ \text{NPS} = \left| \mathcal{S}_{\text{ND}} \right| \]

- **Hypervolume**
  
  \[ H = \text{volume} \left( \bigcup_{i=1}^{\left| \mathcal{S} \right|} v_i \right) \]

- **IGD**
  
  \[ \text{IGD} = \frac{1}{|\mathcal{S}|} \sum_{y \in \mathcal{S}} \min_{x \in \mathcal{S}} \| d_{xy} \| \]

where \( \mathcal{S} \) is the solution set, \( \mathcal{S}_{\text{ND}} \) is the non-dominated solutions set, \( v_i \) is the hyper-cube constructed with a reference point \( W \) and solution \( x_i \) as diagonal corners of the hypercube, \( d_{xy} \) is the Euclidean distance between solution \( x \) and \( y \), and \( \mathcal{S}^* \) is the set of all reference solutions.

4) COMPUTATIONAL DETAILS

The algorithms are coded in MATLAB and executed on an Intel (R) Core (TM) i7-3770 with a 3.40 GHz processor with 12 GB of RAM.

**FIGURE 8.** Sensitivity analysis of alpha parameter.

This purpose, we first obtain the trade-off squads for different values of \( \alpha \). Then, the players who occur in these squads are given ‘BA Rank’ (‘SR Rank’) according to their batting averages (batting strike rates) such that the higher the batting average (batting strike rate) of a player, the better will be its BA Rank (SR rank). The player with the highest batting average (batting strike rate) has 1st BA rank (1st SR rank). The relation between BA rank and SR rank for \( \alpha = 0.1, 0.5, 0.6, \) and \( 1 \) is shown in Fig. 8. For lower \( \alpha \), there is not much difference in both ranks of players, but this difference increases as the value of \( \alpha \) increase from 0.5 to 1. As the T20 is a shorter format than ODI, more hitters, i.e., batters with superb strike rates, are needed to achieve a good score. Therefore, the value of \( \alpha \) is set to 0.6 for the proposed model.

**TABLE 7.** Parameter settings for algorithms and problem model.

| Algorithms   | Population Size | Maximum Generations | Crossover Probability | Mutation Probability | Mutation Rate | Other Parameters | Parameter | Value |
|--------------|-----------------|---------------------|-----------------------|----------------------|--------------|-----------------|-----------|-------|
| MOPSO-CD     | 200             | 500                 | -                     | -                    | 0.5          | Inertia rate = 0.4 | \( \alpha \) | 0.6   |
| NSDE         | -               | -                   | -                     | -                    | 0.5          | Scaling factor = 0.2 | \( m \)   | 23    |
| RNSGA-II     | 1               | 0.1                 | -                     | -                    | -            | \( a_1, a_2 \) | 11        |
| BNSGA-II     | 1               | 0.1                 | -                     | -                    | -            | \( m_1, m_2 \) | 8         |
| MNSGA-II     | 0.9             | 0.05                | -                     | -                    | -            | \( n_1, n_2, n_3 \) | 3         |
|              |                 |                     |                       |                       |              | Crossover rate = 0.1 | \( T \)   | ₹ 8500 L |

Let \( S_i \) be the \( i \)th solution set, and \( S_{\text{ND}} \) are the non-dominated solutions in \( S_i \), then the NPS in \( S_i \) is \( |S_{\text{ND}}| \). This measure compares the NPS of different solution sets.

**b: HYPERVOLUME**

This is the volume of the region dominated by the solution set \( S \) in the objective space. Mathematically, \( \forall x_i \in S \), a hyper-cube \( v_i \) is constructed with a reference point \( W \) and solution \( x_i \) as diagonal corners of the hypercube, the reference point \( W \) can be found by constructing a vector of worst objective values. After that, a union of all hypervolumes is calculated as:

\[ HV = \text{volume} \left( \bigcup_{i=1}^{\left| S \right|} v_i \right) \]

**c: IGD**

This measure evaluates the proximity of the solution set \( S \) to the solution set \( S^* \) and is defined as the average distance of each reference solution \( y \in S^* \) from its nearest solution in \( S \) as:

\[ \text{IGD} = \frac{1}{|S^*|} \sum_{y \in S^*} \min_{x \in S} \| d_{xy} \| \]

where, \( d_{xy} \) is the Euclidean distance between solution \( x \in S \) and \( y \in S^* \).

**B. EXPERIMENTAL RESULTS**

This section compares the solutions obtained using various algorithms for the proposed problem. The simulation is also performed by fixing the preferred players and building partial squads around them.

1) PERFORMANCE COMPARISON OF ALGORITHMS

An experiment is designed in which six algorithms, MNSGA-II, NSDE, RNSGA-II, MOPSO-CD, INSGA-II, and BNSGA-II, are compared using three performance metrics (hypervolume, NPS, IGD), and computational time.
Before this experiment, for both cases, all the algorithms were run independently 40 times such that each run started from different populations to avoid biased and unfair comparisons.

For experiment purposes, we obtain a Pareto optimal set (POS) for each run of the algorithms. Every POS is normalized by normalizing each \(k\)th objective function value \(f_k\) using the relation (21), where \(F\) is the combination of POS of the six algorithms \((6 \times 40 = 240\text{POSs})\). The normalized objective function values \(f'_k\) lies in the interval \([0,1]\); therefore, \((1,1)\) is the reference point for calculating the hypervolume. For IGD, the reference set \(\mathcal{S}^*\) is the set obtained after normalizing the non-dominated objective values of the set \(\mathcal{F}\).

\[
  f'_k = \frac{f_k - \min(F_k)}{\max(F_k) - \min(F_k)}, \quad k = 1, 2
\]  

The performance metrics for each algorithm are calculated using the corresponding normalized POSs. The median, average, maximum or minimum, and standard deviation of these performance metrics for case-I and case-II are shown in Table 8. The best values are marked in bold. The above comparison can also be visualized using line charts in Fig. 9, which plot the Average hypervolume, Average IGD, Average NPS, and Average time of each algorithm. The results suggest that BNSGA-II is the best performing algorithm in all performance metrics. The Pareto fronts using the six algorithms for case-I and case-II are shown in Fig. 10-11. The run of each algorithm with maximum hypervolume is used to plot these Pareto fronts. Here each Pareto optimal solution represents a trade-off squad. The \(x\)-coordinate \((y\)-coordinate\) represents the normalized -net batting (bowling) performance of a squad. The extreme (toward origin) squad on the \(x\)-axis \((y\)-axis\) is best in batting (bowling) performance. The figures also show that BNSGA-II dominates the other algorithms, especially MNSGA-II, MOPSO-CD, and RNSGA-II.

### Table 8. Comparison of algorithms’ performance for case-I and case-II.

| Algorithm  | Case-I |   |   | Case-II |   |   |
|------------|--------|---|---|---------|---|---|
|            | NPS    | Time (in seconds) | Hypervolume | IGD    |   |   |
|            | Avg. value | Avg. value | Std. Dev | Max. Value | Avg. Value | Median. Value | Std. Dev | Min. Value | Avg. Value | Median. Value | Std. Dev |
| MNSGA-II   | 58.7250 | 109.6087 | 5.1356 | 0.8716 | 0.8178 | 0.8196 | 0.0255 | 0.0323 | 0.0712 | 0.0675 | 0.0210 |
| MOPSO-CD   | 34.8500 | 388.9935 | 30.8479 | 0.8903 | 0.8427 | 0.8460 | 0.0322 | 0.0260 | 0.0509 | 0.0472 | 0.0233 |
| NSDE       | 159.0250 | 495.6788 | 44.1512 | 0.9382 | 0.9279 | 0.9278 | 0.0053 | 0.0050 | 0.0115 | 0.0109 | 0.0034 |
| RNSGA-II   | 123.9500 | 515.8244 | 32.9480 | 0.9335 | 0.9202 | 0.9218 | 0.0082 | 0.0062 | 0.0115 | 0.0110 | 0.0028 |
| INSGA-II   | 164.3000 | 108.9525 | 8.6757 | 0.9381 | 0.9266 | 0.9263 | 0.0063 | 0.0052 | 0.0125 | 0.0117 | 0.0046 |
| BNSGA-II   | 165.0250 | 129.0056 | 13.5195 | 0.9504 | 0.9442 | 0.9454 | 0.0044 | 0.0015 | 0.0038 | 0.0032 | 0.0022 |

### Table 9. Friedman ranking based on hypervolume and IGD, and its test statistics.

| Algorithm  | Case-I |   |   | Case-II |   |   |
|------------|--------|---|---|---------|---|---|
|            | Hypervolume | IGD |   | Hypervolume | IGD |   |
|            | 1/HV | IGD | 1/HV | IGD |   |   |
| MNSGA-II   | 5.80  | 5.80 | 5.33 | 5.28 |
| RNSGA-II   | 3.53  | 3.00 | 3.45 | 3.40 |
| INSGA-II   | 2.80  | 3.08 | 1.85 | 1.88 |
| BNSGA-II   | 1.83  | 1.13 | 1.20 | 1.20 |
| NSDE       | 2.65  | 2.80 | 3.50 | 3.53 |
| MOPSO-CD   | 5.20  | 5.20 | 5.68 | 5.73 |

This result is further investigated statistically by performing the Friedman test [23], which provides the algorithms' ranking and analyses whether the results evaluated by different algorithms demonstrate any inequality. The null hypothesis assumes that the performance of all the algorithms is equivalent or that none of the algorithms performs significantly differently. Therefore, the rejection of the null hypothesis shows a significant difference in the performance of the algorithms. IBM SPSS is used to perform this test with critical statistical significance \(\alpha = 0.05\). Here, the Friedman test is performed for ranking the six algorithms (for each case separately) on the basis of their hypervolumes\(^{-1}\) and IGDs on 40 runs. The Friedman mean rank and the corresponding test statistics are shown in Table 9. As all \(p\) values are less than 0.05, we reject the null hypothesis and accept that a significant difference exists in the performance of the algorithms. BNSGA-II has the highest mean rank in terms of hypervolume and IGD. NSDE has the second-highest mean rank in both metrics for case-I, and INSGA-II has the second-highest
mean rank for case-II. However, INSGA-II is more computationally efficient than NSDE, as can be seen in Tables 7-8. Further, MNSGA-II and MOPSO-CD have the lowest mean rank among all the compared algorithms.

2) SQUADS WITH PREFERRED PLAYERS
The franchise owners usually prefer some particular players they want in the squad regardless of the players’ cost. For instance, suppose ‘Rohit Sharma’ (the captain of the maximum times IPL winning team) and ‘Jasprit Bumrah’ (one of the best death-overs bowlers) are the preference of a franchise owner for building a squad. Then, it is possible to obtain balanced trade-off squads along with these preferred players using BNSGA-II. For this purpose, we use a repair mechanism modifying Algorithm 1 such that the index of these two players has ‘1’ in the chromosome representation (Fig.3) to maintain these two players in the squad, as shown in Table 10. In this way, BNSGA-II helps incorporate the preference of the franchise owner in squad selection which is helpful in a dynamic environment, such as an auction.

VII. ANALYSIS AND DISCUSSION
This section analyses case-I and case-II on the basis of the trade-off squads obtained using BNSGA-II. It also analyses the trade-off squads based on cost, fielding performance, and players’ performance in IPL 2020.

A. ANALYSIS OF DIFFERENT CASES
Before analyzing the trade-offs for case-I and case-II, we first obtain the trade-off squads that maximize the net batting and bowling performance and satisfy the constraints (5-9) that are based only on IPL regulations. The majority of players in almost all squads are bowlers or all-rounders, as shown in Fig. 12. Another concern is that some squads have no bowlers, which is impossible in the actual situation as an IPL squad should have the right proportion of pure batters, pure bowlers, and all-rounders.

Further, case-I bounds the sum of pure batters (bowlers) and batting all-rounders (bowling all-rounders); however, as shown in Fig. 13, in most squads, more than 60% of players are all-rounders. The reason for this is that all-rounders contribute to both the batting and bowling performance of the squad. So, the result suggests that the squads using case-I are also not balanced concerning players’ expertise.
Unlike this, a suitable proportion of pure batters, pure bowlers, and all-rounders are present in each squad for case-II, as shown in Fig.14. This result is promising since an efficient playing XI team can be made through these squads. The above analysis shows the motivation for developing a new model for IPL squad selection.

B. TRADE-OFF SQUADS ANALYSIS

Case-II provided balanced squads; therefore, the trade-off squads obtained using BNSGA-II for this case are further analyzed based on the cost, fielding performance, and players’ performance in IPL 2020.

1) SQUADS COST

The ‘-net batting performance versus price’ and ‘-net bowling performance versus price’ is shown in Fig.15. The figure illustrates that a squad with higher batting (bowling) performance does not need to be costlier than the squad with comparatively lower batting (bowling) performance. More clearly, two squads with similar costs may have immense differences in their batting and bowling performances. Another finding is that the squad having high net batting performance is mostly costlier than the squad with high net bowling performance, as auction prices of pure batters are higher than pure bowlers.

Fig. 16 shows the total price versus the number of star players in a squad. The presence of star players increases a squad’s net batting and bowling performance. However, it is clear from the figure that a squad with more star players not necessarily be more expensive than a squad with fewer star players. It also demonstrates that many squads have the same number of star players but different total costs. The trade-off squads of the run with maximum hypervolume are used for this analysis.

| Wicket-Keeper    | Fielding average |
|------------------|------------------|
| AB de Villiers   | 0.77             |
| KL Rahul         | 0.49             |
| Rishabh Pant     | 0.75             |
| Tom Banton       | 0.71             |
2) FIELDING PERFORMANCE

A player’s fielding performance accounts for the number of catches, stumped out, and wickets. However, there is no data regarding the number of runs saved by a player. With this incomplete information, we cannot say about the fielding performance of a player. However, this data can estimate a wicket-keeper’s performance in terms of the fielding average (22), as the higher the fielding average of a wicket-keeper, the better its fielding performance. Table 11 shows the wicket-keepers for a trade-off squad; here, AB de Villiers is the best choice for wicket-keeping.

\[
\text{Fielding average} = \frac{\text{Number of runs out} + \text{Number of catches} + \text{Number of stumpouts}}{\text{Number of matches played}}
\]

\[
(22)
\]

3) PLAYERS’ PERFORMANCE IN IPL 2020

The trade-off squads used for this analysis (BNSGA-II’ run with maximum hypervolume) have total 44 players occurring in different combinations, as given in Table 12.

A player’s frequency defines the number of times they occur in all the possible trade-off squads. The players with higher frequency are the key players for a squad. First, the frequency of each of the 44 players is calculated to identify the top players. Next, the frequency percentage of each player is calculated using (23).

\[
f(\%) = \frac{\text{Frequency of player}}{\text{Total number of trade-off squads}} \times 100
\]

\[
(23)
\]

Out of 44 players, 10 with 100% frequency in the trade-off squads are Akash Singh, Chris Gayle, Darshan Nalkande, Devdutt Padikkal, Imran Tahir, KL Rahul, M Siddharth, R. Sai Kishore, Rashabh Pant, and Virat Kohli. 14 of them did not play in IPL 2020. The 30 players who played are marked in bold, and their performance records are obtained from iplt20.com. The players with one of the top ten records in IPL 2020 (in terms of most runs, most runs (over), best batting average, best batting strike rate, fastest fifties, most fifties, fastest centuries, most sixes, highest score (innings), best bowling average, best bowling economy, best bowling strike rate, most wickets, most dot balls, most maiden overs and player points) are shown in Table 13. The remaining players are mainly in the top twenty of these records. The above analysis supports the claim that the players selected based on performance could perform well in the upcoming matches.

VIII. CONCLUSION AND FUTURE SCOPE

An efficient playing XI team for IPL implies a balanced team where a sufficient number of players with different expertise are present. The model proposed in the present study tries to form a balanced squad for IPL and has the following potential:

- It ensures the presence of a sufficient number of pure batters, pure bowlers, and all-rounders.
- It evaluates the performance of a batter by estimating his batting performance through a batting performance factor which takes count of both batting average and batting strike rate and can prioritize one over the other.
- It evaluates the performance of a bowler through a combined bowling rate, which includes all three bowling statistics, i.e., bowling average, bowling strike rate, and economy rate.
- It assures that there are a sufficient number of star pure batters, star pure bowlers, and star all-rounders under the
same budget for better performance and to meet franchise business prospects.

- The performance of the players is assessed not only through the IPL data but also through the data of other tournaments played in the T20 format.

This study identifies the squad selection problem as a knapsack problem due to the nature of the decision for a player selection (yes or no) and the objective of maximizing squad performance under a fixed budget. On this basis, two combinatorial optimization algorithms BNSGA-II and INSGA-II, are developed to obtain its trade-off solutions. The proposed algorithms are validated by comparing them with the existing optimization algorithm for IPL team selection and with the three other prominent algorithms, RNSGA-II, NSDE, and MOPSO-CD, using the data of players auctioned in IPL 2020. Overall, this comparison demonstrates BNSGA-II as the best-performed algorithm in terms of hyper-volume and IGD and reports similar performances of NSDE and INSGA-II, though INSGA-II is computationally efficient than NSDE.

Further, it is observed that many of the players selected through the proposed model played well in IPL 2020. This model can be an additional tool for the selectors and the franchise owners to make a more concrete, unbiased, and systematic decision for the selection of the squad. Other than cricket, there can be other sports also where such a model can be implemented. At a broader level, this being a selection problem can be implemented under any scenario where a limited selection is to be done out of the given sample.

The present study tries to develop a model that is appropriate for selecting players an IPL squad. However, there are some limitations that may be addressed in a future study:

- In this study, the players’ past information has been used to obtain the trade-off squads. Therefore, some amount of risk is involved in making such a decision because a player’s performance may not be the same in the upcoming matches. So whether more weightage is to be given to the in-form player in comparison to the out-of-form player may also be considered while designing the model.

- The most prominent factors, like batting average, bowling average etc., are considered in this study for designing a well-balanced team. However, there are other important features also, like the condition of the pitch, home ground etc., that may also be considered while designing the model for a well-balanced team.

- In cricket, there are several forms of bowling and batting. Like in the case of bowling, there are fast bowlers, medium-pace bowlers, and spinners, and for batting, there are openers and middle-order batters. Further studies may be done to include all these types while developing a model.

In addition, it will be interesting to design a graphical user interface for the selection of a balanced squad.

REFERENCES

[1] V. Lunawat, “Sport development in India: Professional sport league systems,” in Sport and Development in Emerging Nations, London, U.K.: Routledge, Apr. 2021, pp. 77–93, doi: 10.4324/9781003203400-R-2.

[2] F. Ahmed, K. Deb, and A. Jindal, “Multi-objective optimization and decision making approaches to cricket team selection.” Appl. Soft Comput., vol. 13, no. 1, pp. 402–414, Jan. 2013, doi: 10.1016/j.asoc.2012.07.031.

[3] S. Chand, H. K. Singh, and T. Ray, “Team selection using multi-/many-objective optimization with integer linear programming,” in Proc. IEEE Congre. Evol. Comput. (CEC), Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477945.

[4] G. D. I. Barr and B. S. Kantor, “A criterion for comparing and selecting batsmen in limited overs cricket,” J. Oper. Res. Soc., vol. 55, no. 12, pp. 1266–1274, Dec. 2004, doi: 10.1057/palgrave.jors.2601800.

[5] H. H. Lemmer, “The combined bowling rate as a measure of bowling performance in cricket,” South Afr. J. Res. Sport, Phys. Educ. Recreation, vol. 24, no. 2, pp. 37–44, Feb. 2002, doi: 10.4314/sajr.v24i2.25839.

[6] D. Connolly, “Knapsack problems: Algorithms and computer implementations,” J. Oper. Res. Soc., vol. 42, no. 6, p. 513, Accessed: Nov. 19, 2020, doi: 10.1057/jors.1991.99.

[7] S. S. Sathy and M. S. Jamal, “Applying genetic algorithm to select an optimal cricket team,” in Proc. Int. Conf. Adv. Comput., Commun. Control, 2009, pp. 43–47, doi: 10.1109/ICACC.2009.5313954.

[8] D. Bhattacharjee and H. Saikia, “An objective approach of balanced cricket team selection using binary integer programming method.” OPSEARCH, vol. 53, no. 2, pp. 225–247, Jun. 2016, doi: 10.1007/s12597-015-0228-3.

[9] D. Bhattacharjee and H. Saikia, “On performance measurement of cricketers and selecting an optimum balanced team,” Int. J. Perform. Anal. Sport, vol. 14, no. 1, pp. 262–275, Apr. 2014, doi: 10.1080/24748668.2014.1165520.

[10] F. Ahmed, A. Jindal, and K. Deb, “Cricket team selection using evolutionary multi-objective optimization,” in Proc. 2nd Int. Conf. Swarm, Evol., Memetic Comput., in Lecture Notes in Computer Science, vol. 7077. Kanpur, India: Indian Inst. Technol., 2011, pp. 71–78, doi: 10.1007/978-3-642-27242-4_9.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Aug. 2002, doi: 10.1109/4235.996017.

[12] S. Verma, M. Pant, and V. Snaezel, “A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems,” IEEE Access, vol. 9, pp. 57757–57791, 2021, doi: 10.1109/ACCESS.2021.3070634.

[13] S. Ghasemi-Falavarjani, M. Nematabakhsh, and B. S. Ghalafarokhi, “Context-aware multi-objective resource allocation in mobile cloud,” Comput. Elect. Eng., vol. 44, pp. 218–240, May 2015, doi: 10.1016/j.compeleceng.2015.02.006.

[14] A. Kaushik and D. P. Vidyarthi, “An energy-efficient reliable grid scheduling model using NSGA-II,” Eng. Comput., vol. 32, no. 3, pp. 355–376, Jul. 2016, doi: 10.1007/s00366-015-0419-9.

[15] A. W. Iorio and X. Li, “Solving rotated multi-objective optimization problems using differential evolution,” in AI 2004: Advances in Artificial Intelligence. Cairns, Australia: Springer-Verlag, 2004, pp. 861–872, doi: 10.1007/978-3-540-30549-1_74.

[16] C. R. Raquel and P. C. Naval, “An effective use of crowding distance in multiobjective particle swarm optimization,” in Proc. Conf. Genetic Evol. Comput. (GECCO), 2005, pp. 257–264, doi: 10.1145/1068009.1068047.

[17] G. R. Amin and S. K. Sharma, “Cricket team selection using data envelopment analysis,” Eur. J. Sport Sci., vol. 14, no. 1, pp. 836–8376, Jan. 2014, doi: 10.1080/17461391.2012.705333.

[18] R. Chaudhary, S. Bharadwaj, and S. Lakra, “A DEA model for selection of Indian cricket team players,” in Proc. Amity Int. Conf. Artif. Intell. (AICAI), Feb. 2019, pp. 224–227, doi: 10.1109/AICAI.2019.8701424.

[19] A. Ghosh, A. Sinha, P. Mondal, A. Roy, and P. Saha, “Indian premier league player selection model based on Indian domestic league performance,” in Proc. IEEE 11th Ann. Comput. Commun. Workshop Conf. (CCWC), Jan. 2021, pp. 277–282, doi: 10.1109/CCWC51732.2021.9376011.

[20] C. D. Prakash, C. D. Prakash, C. Patwardhan, and C. V. Lakshmi, “Team selection strategy in IPL 9 using random forests algorithm article,” Int. J. Comput. Appl., vol. 139, no. 12, pp. 1–7, 2016, doi: 10.5120/ijca201609516.
[21] D. E. Goldberg, “Alleles, loci, and the traveling salesman problem,” in Proc. 1st Int. Conf. Genetic Algorithms Appl., 1985, pp. 154–159.

[22] N. Riquelme, C. Von Lucken, and B. Baran, “Performance metrics in multi-objective optimization,” in Proc. Latin Amer. Comput. Conf. (CLEI), Oct. 2015, pp. 1–11, doi: 10.1109/CLEI.2015.7360024.

[23] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” Swarm Evol. Comput., vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/J.SWEVO.2011.02.002.

SHANU VERMA received the M.Sc. degree in mathematics from the Central University of Rajasthan, India. She is currently a Ph.D. Scholar with the Department of Applied Mathematics and Scientific Computing, IIT Roorkee, India. Her research interests include multi-objective optimization, combinatorial optimization, evolutionary algorithms, swarm intelligence-based algorithms, and their applications to complex real-life problems.

VIVEKANAND PANDEY received the M.Sc. (Tech.) degree in mathematics and computer science from the Central University of Rajasthan, India. He is currently a Ph.D. Scholar with the Department of Applied Mathematics and Scientific Computing, IIT Roorkee, India. His research interests include agri-food supply chain, blockchain technology, machine learning algorithms, and evolutionary algorithms.

VACLAV SNASEL (Senior Member, IEEE) is currently a Professor with the Department of Computer Science, VSB—Technical University of Ostrava, Czech Republic. He works as a Researcher and a University Teacher. He is also the Dean of the Faculty of Electrical Engineering and the Computer Science Department. He is also the Head of the Research Program IT4 Knowledge Management, European Center of Excellence IT4 Innovations. His research and development experience includes over 30 years in the industry and academia. He works in a multi-disciplinary environment involving artificial intelligence, social networks, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, neural networks, web intelligence, nature and bio-inspired computing, data mining, and applied to various real-world problems.