Optimal Lineups in Twenty20 Cricket

Harsha Perera, Jack Davis and Tim B. Swartz *

Abstract

This paper considers the determination of optimal team lineups in Twenty20 cricket where a lineup consists of three components: team selection, batting order and bowling order. Via match simulation, we estimate the expected runs scored minus the expected runs allowed for a given lineup. The lineup is then optimized over a vast combinatorial space via simulated annealing. We observe that the composition of an optimal Twenty20 lineup sometimes results in nontraditional roles for players. As a by-product of the methodology, we obtain an "all-star" lineup selected from international Twenty20 cricketers.

Keywords: Cricket, Relative value statistics, Simulated annealing.

*Harsha Perera and Jack Davis are PhD candidates, and Tim Swartz is Professor, Department of Statistics and Actuarial Science, Simon Fraser University, 8888 University Drive, Burnaby BC, Canada V5A1S6. Swartz has been partially supported by grants from the Natural Sciences and Engineering Research Council of Canada. The authors thank anonymous reviewers whose comments helped improve the manuscript.

1 INTRODUCTION

Twenty20 cricket (or T20 cricket) is a form of limited overs cricket which has gained popularity worldwide. Twenty20 cricket was showcased in 2003 and involved matches between English and Welsh domestic sides. The rationale behind the introduction of T20 was to provide an exciting version of cricket with matches concluding in three hours duration or less. There are now various professional T20 competitions where the Indian Premier League (IPL) is regarded as the most prestigious. Even in Canada (not exactly known as a cricketing country), every game of the IPL is telecast on live television.

Except for some subtle differences (e.g. fielding restrictions, limits on the number of overs per bowler, powerplays, etc.), Twenty20 cricket shares many of the features of one-day cricket. One-day cricket was introduced in the 1960s, and like T20 cricket, is a version of cricket based on limited overs. The main difference between T20 cricket and one-day cricket is that each batting side in T20 is allotted 20 overs compared to 50 overs in one-day cricket.

Consequently, many of the strategies used in one-day cricket have trickled down to Twenty20 cricket. This paper investigates the determination of T20 team lineups (i.e. team selection, batting orders and bowling orders). It is desirable that teams field their strongest sides, and doing so, requires a judgment on the relative value of batting versus bowling.

The question of team selection in cricket has been investigated by various authors. Barr and Kantor (2004) advocated the use of a nonlinear combination of the strike rate (runs scored per 100 balls faced) with the batting average (runs scored per dismissal) to select batsmen in one-day international (ODI) cricket. We note that the proposed metric involves an arbitrary weight α and that individual player selection does not account conditionally for players already selected. Swartz et al. (2006) obtained optimal batting orders in ODI cricket using batting characteristics from multinomial regression. However, the paper failed to look at the effect of bowling and did not address the initial team selection problem. Bretteny (2010) and Lemmer (2013) considered integer optimization methods for selecting players in fantasy league cricket and in limited overs cricket. Their methodology is based on performance measures computed from summary statistics such as the batting average, strike rate, etc. Integer programming constraints include various desiderata such as the inclusion of a single wicket-keeper and specified numbers of batsmen, all-rounders and bowlers. Two major drawbacks of the approach are the ad hoc choice of performance statistics (the objective function) and the lack of validation against "optimal" team selections.

In this paper, we investigate the problem of determining team lineups in T20 cricket from the point of view of *relative value statistics*. Relative value statistics have become prominent in the sporting literature as they attempt to quantify what is really important in terms of winning

and losing matches. For example, in Major League Baseball (MLB), the VORP (value over replacement player) statistic has been developed to measure the impact of player performance. For a batter, VORP measures how much a player contributes offensively in comparison to a replacement-level player (Woolner 2002). A replacement-level player is a player who can be readily enlisted from the minor leagues. As another example, the National Hockey League (NHL) reports plus-minus statistics. The statistic is calculated as the goals scored by a player's team minus the goals scored against the player's team while he is on the ice. More sophisticated versions of the plus-minus statistic have been developed by Schuckers et al. (2011) and Gramacy, Taddy and Jensen (2013).

In cricket, a team wins a match when the runs scored while batting exceed the runs conceded while bowling. Therefore, it is the run differential that is the holy grail of performance measures in cricket. An individual player can be evaluated by considering his team's run differential based on his inclusion and exclusion in a lineup. Clearly, run differential cannot be calculated from actual match results in a straightforward way because there is variability in match results and conditions change from match to match. Our approach in assessing performance is based on simulation methodology where matches are replicated. Through simulation, we can obtain long run properties (i.e. expectations) involving run differential. By concentrating on what is really important (i.e. run differential), we believe that our approach addresses the core problem of interest in the determination of T20 team lineups.

In Section 2, we begin with some exploratory data analyses which investigate batting and bowling characteristics of T20 cricketers. We observe that player characteristics are not clustered according to position. Rather, player skills appear to vary on the continuum. We then provide an overview of the simulator developed by Davis, Perera and Swartz (2015) which is the backbone of our analysis and is used in the estimation of expected run differential.

In Section 3, a simulated annealing algorithm is proposed to obtain optimal team lineups (i.e. the joint determination of team selection, batting order and bowling order). The algorithm searches over the vast combinatorial space of team lineups to produce an optimal lineup with the greatest expected run differential. This is a more complex problem than is typically considered in the literature where only team selection is discussed. We remark that we have not seen any previous work that addresses bowling orders. In the search for an optimal team lineup, the objective function is the run differential which is the quantity that is relevant to winning and losing matches. The simulated annealing algorithm requires fine tuning in order to effectively search the space and converge to the optimal lineup.

In Section 4, we provide two applications of the simulated annealing algorithm. First, we determine an optimal lineup for both India and South Africa in T20 cricket and we compare

these lineups to actual lineups that have been used in the recent past. Some comments are then made on the optimal composition of teams. Second, the simulated annealing algorithm is applied to an international pool of players to identify an "all-star" lineup. We compare the resulting team selection with some common beliefs concerning "star" players. We conclude with a short discussion in Section 5.

2 PRELIMINARIES

To initiate discussion on T20 team composition, we begin with some exploratory data analyses. We consider T20 matches involving full member nations of the International Cricket Council (ICC). Currently, the 10 full members of the ICC are Australia, Bangladesh, England, India, New Zealand, Pakistan, South Africa, Sri Lanka, West Indies and Zimbabwe. The matches considered were those that took place from the first official match in 2005 until May 21, 2014. Details from these matches can be found in the Archive section of the CricInfo website (www.espncricinfo.com). In total, we obtained data from 282 matches.

For batting, we use familiar quantities. We let BA denote the batting average (runs scored per dismissal) and we let SR denote the batting strike rate (runs scored per 100 balls). Good batting is characterized by large values of both BA and SR. We define similar quantities for the bowling average BA (runs allowed per dismissal) and the bowling economy rate ER (runs allowed per over).

In Figure 1, we produce a scatterplot of BA versus SR for the 40 batsmen in our dataset who have faced at least 500 balls. As expected, players who are known as bowlers are not as proficient at batting (lower left section of the plot). We observe that batsmen have different styles. For example, V. Kohli of India has the best batting average but his strike rate is only slightly above average. On the other hand, Y. Singh of India scores many runs (i.e. has a high strike rate) yet his batting average is not exceptional. A. Hales of England and K. Pietersen of England are plotted deep in the upper right quadrant, and are the ideal combination of reliability (i.e. batting average) and performance (i.e. strike rate). We also observe a continuum in batting abilities along both axes with no apparent clustering. This is an important observation for the determination of optimal team selections. Consequently, players should be selected on their own merits rather than filling quotas of pure batsmen, all-rounders and bowlers.

In Figure 2, we produce a scatterplot of BA versus ER for the 33 bowlers in our dataset who have bowled at least 500 balls. Again, we observe that bowlers have different characteristics. For example D. Vettori of New Zealand concedes very few runs (i.e. low economy rate) but he is mediocre in taking wickets as highlighted by his middling bowling average. We also observe a

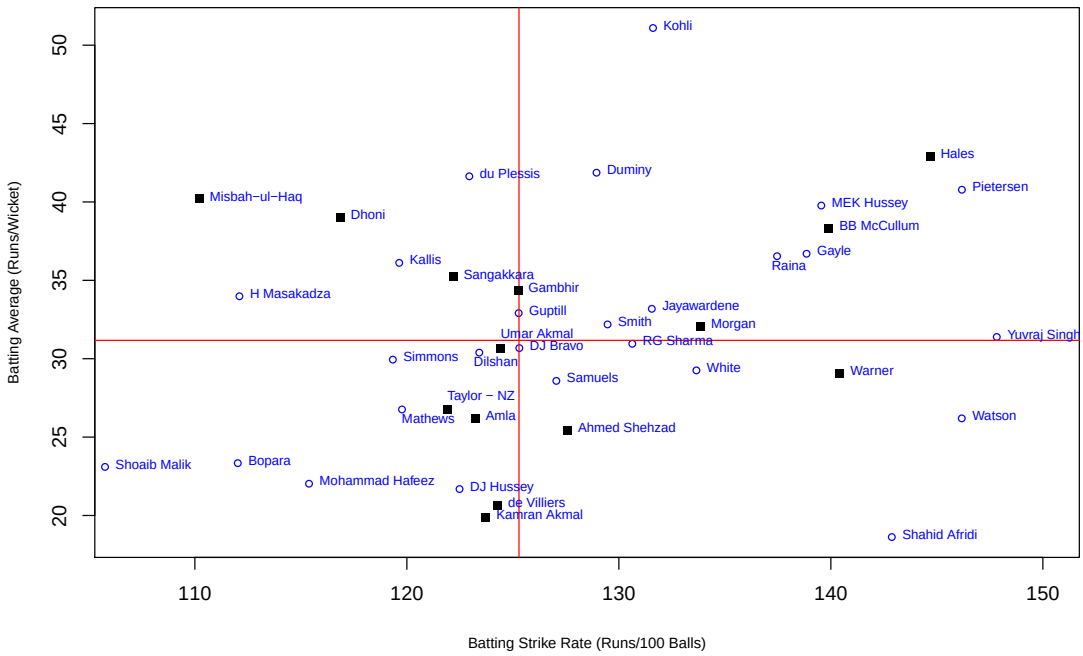

Figure 1: Scatterplot of batting average versus batting strike rate. Pure batsmen (i.e. those who never bowl) are indicated by black squares.

continuum in bowling abilities with no apparent clustering.

We now provide an overview of the simulator developed by Davis, Perera and Swartz (2015) which we use for the estimation of expected run differential. Ignoring extras (sundries) that arise via wide-balls and no-balls, there are 8 broadly defined outcomes that can occur when a batsman faces a bowled ball. These batting outcomes are listed below:

$$\begin{aligned}
\text{outcome } j = 0 &\equiv 0 \text{ runs scored} \\
\text{outcome } j = 1 &\equiv 1 \text{ runs scored} \\
\text{outcome } j = 2 &\equiv 2 \text{ runs scored} \\
\text{outcome } j = 3 &\equiv 3 \text{ runs scored} \\
\text{outcome } j = 4 &\equiv 4 \text{ runs scored} \\
\text{outcome } j = 5 &\equiv 5 \text{ runs scored} \\
\text{outcome } j = 6 &\equiv 6 \text{ runs scored} \\
\text{outcome } j = 7 &\equiv \text{dismissal}
\end{aligned} \tag{1}$$

In the list (1) of possible batting outcomes, we include byes, leg byes and no balls where the resultant number of runs determines one of the outcomes $j = 0, \ldots, 7$. We note that the outcome $j = 5$ is rare but is retained to facilitate straightforward notation.

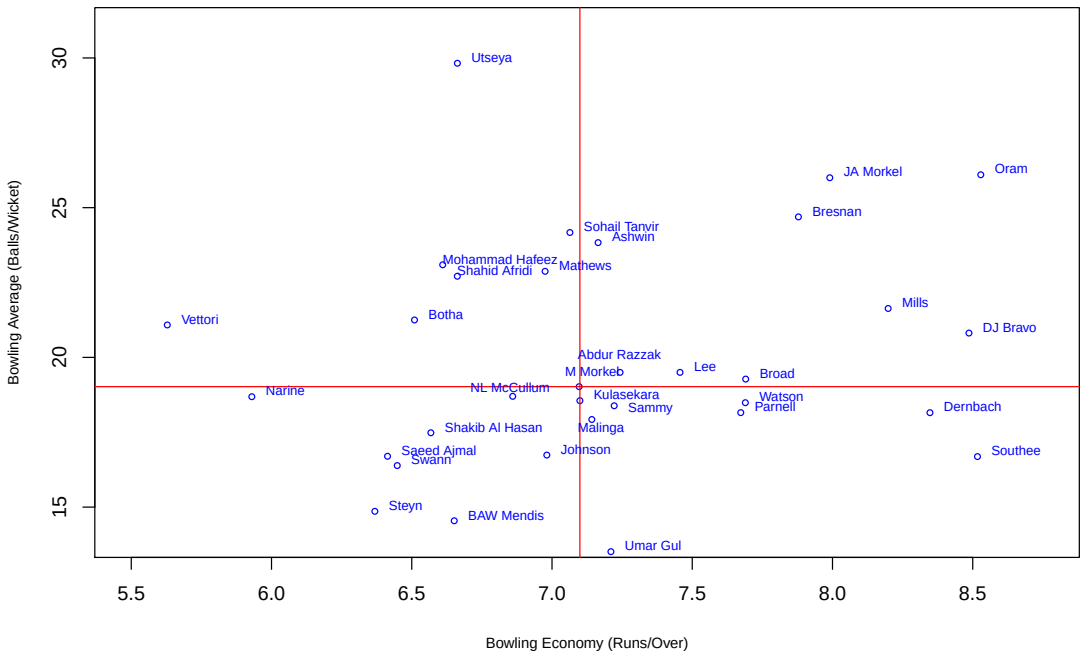

Figure 2: Scatterplot of bowling average versus bowling economy rate.

According to the enumeration of the batting outcomes in (1), Davis, Perera and Swartz (2015) suggested a statistical model for the number of runs scored by the ith batsman:

$$(X_{iow0}, \ldots, X_{iow7}) \quad \sim \quad \text{multinomial}(m_{iow};\ p_{iow0}, \ldots, p_{iow7}) \tag{2}$$

where X_{iowj} is the number of occurrences of outcome j by the ith batsman during the oth over when w wickets have been taken. In (2), m_{iow} is the number of balls that batsman i has faced in the dataset corresponding to the oth over when w wickets have been taken. The dataset is "special" in the sense that it consists of detailed ball-by-ball data. Typically, researchers study aggregate match data. The data were obtained using a proprietary parser which was applied to the commentary logs of matches listed on the CricInfo website.

The estimation of the multinomial parameters p_{iowj} in (2) is a high-dimensional and complex problem. The complexity is partly due to the sparsity of the data; there are many match situations (i.e. combinations of overs and wickets) where batsmen do not have batting outcomes. For example, bowlers typically bat near the end of the batting order and do not face situations when zero wickets have been taken.

To facilitate the estimation of the multinomial parameters, Davis, Perera and Swartz (2015)

introduced the simplification

$$p_{iowj} = \frac{\tau_{owj}\ p_{i70j}}{\sum_j \tau_{owj}\ p_{i70j}}\ . \tag{3}$$

In (3), the parameter p_{i70j} represents the baseline characteristic for batsman i with respect to batting outcome j. The characteristic p_{i70j} is the probability of outcome j associated with the ith batsman at the juncture of the match immediately following the powerplay (i.e. the 7th over) when no wickets have been taken. The multiplicative parameter τ_{owj} scales the baseline performance characteristic p_{i70j} to the stage of the match corresponding to the oth over with w wickets taken. The denominator in (3) ensures that the relevant probabilities sum to unity. There is an implicit assumption in (3) that although batsmen are unique, their batting characteristics change by the same multiplicative factor which is essentially an indicator of aggression. For example, when aggressiveness increases relative to the baseline state, one would expect $\tau_{ow4} > 1$ and $\tau_{ow6} > 1$ since bolder batting leads to more 4's and 6's.

Given the estimation of the parameters in (3) (see Davis, Perera and Swartz 2015), a straightforward algorithm for simulating first innings runs against an average bowler is available. One simply generates multinomial batting outcomes in (1) according to the laws of cricket. For example, when either 10 wickets are accumulated or the number of overs reaches 20, the first innings is terminated. Davis, Perera and Swartz (2015) also provide modifications for batsmen facing specific bowlers (instead of average bowlers), they account for the home field advantage and they provide adjustments for second innings simulation.

In summary, with such a simulator, we are able to replicate matches and estimate the expected run differential when Team A (lineup specified) plays against Team B (lineup specified).

3 OPTIMAL LINEUPS

Consider the problem of selecting 11 players from a pool of players for a Twenty20 cricket team and then determining the batting order and the bowling order for the selected players. We refer to this overall specification as a "team lineup". The objective is to select a team lineup that produces the greatest expected difference between runs scored R_s and runs allowed R_a

$$\begin{aligned} \mathrm{E}(D) &= \mathrm{E}(R_s - R_a) \\ &= \mathrm{E}(R_s) - \mathrm{E}(R_a) \end{aligned} \tag{4}$$

against an average opponent.

For a particular team lineup, the calculation of E(D) in (4) depends on the batting and bowling characteristics of the selected players. Clearly, this is not something that we can obtain analytically. Therefore, we simulate the batting innings and the bowling innings for a particular lineup and then take the average of the run difference D over many hypothetical matches. In calculating D, we simulate first innings runs for both teams. This seems reasonable since in practice, second innings batting terminates if second innings runs exceed the target. Since the performance measure E(D) is recorded in runs, it is desirable that our estimates are accurate to within a single run. We have found that simulating $N = 25000$ pairs of innings provides a standard error of less than 0.2 for the expected run difference E(D).

We now formalize the solution space of team lineups. Although the set of possible team lineups is discrete and finite, it is a vast combinatorial space. Let M denote the number of players who are available in the team selection pool. Then the first step in obtaining an optimal lineup is the specification of the 11 active players. There are $\binom{M}{11}$ potential selections. Once team selection is determined, there are $11! \approx 40$ million potential batting orders alone. And with a potentially different bowler for each over in a bowling innings, there is an upper bound of $(11)^{20}$ bowling orders. The number of bowling orders is an upper bound since the rules of T20 cricket prohibit a bowler for bowling more than four overs. Therefore a simple upper bound for the cardinality of the solution space is given by

$$\binom{M}{11}(11!)(11)^{20} = \frac{M!\ (11)^{20}}{(M-11)!}\ . \tag{5}$$

Our problem is therefore computationally demanding. We need to optimize team lineups over an enormous combinatorial space where the objective function E(D) itself requires many simulations of innings. For example, if $M = 15$, the upper bound (5) yields 3.67×10^{31}. In the Appendix, we provide a more nuanced description of the solution space taking into account some detailed aspects of the game.

To carry out the optimization, we employed a simulated annealing algorithm (Kirkpatrick, Gelatt and Vecchi 1983). Simulated annealing is a probabilistic search algorithm that explores the combinatorial space, spending more time in regions corresponding to promising team lineups. Successful implementations of simulated annealing typically require careful tuning with respect to the application of interest.

For our problem, simulated annealing proceeds as follows: Denote the current team lineup at the beginning of step i of the algorithm as lineup c_{i-1} where $i = 1, 2, \ldots$ The algorithm has a prescribed starting lineup c_0 where a typical T20 lineup is straightforward to specify and is recommended. During each step of the algorithm, a candidate lineup c^* is generated (as described

in the fine tuning of the algorithm - Section 3.1). Then N_i matches are generated with the lineup c^* which provides the estimated run differential $\hat{\mathrm{E}}(D_{c^*})$. The candidate lineup is accepted as the current lineup, i.e. we set $c_i = c^*$ if

$$\hat{\mathrm{E}}(D_{c^*}) > \hat{\mathrm{E}}(D_{c_{i-1}}) \ . \tag{6}$$

The candidate lineup is also accepted, i.e. we set $c_i = c^*$ if both

$$\hat{\mathrm{E}}(D_{c^*}) \leq \hat{\mathrm{E}}(D_{c_{i-1}}) \tag{7}$$

and if a randomly generated $u \sim \mathrm{Uniform}(0, 1)$ satisfies

$$u < \exp\left\{\frac{\hat{\mathrm{E}}(D_{c^*}) - \hat{\mathrm{E}}(D_{c_{i-1}})}{t_i}\right\} \tag{8}$$

where t_i is a specified parameter referred to as the "temperature". If the proposed lineup c^* is not accepted, then the current lineup is carried forward, i.e. we set $c_i = c_{i-1}$. The temperature is subject to a non-increasing "cooling schedule" $t_i \to 0$. We therefore observe that the simulated annealing algorithm goes "up the hill" (6) to states corresponding to preferred lineups but also occasionally goes "down the hill" (7) allowing the algorithm to escape regions of local maxima in the search for a global maximum. Intuitively, condition (8) says that as the temperature cools (i.e. gets closer to zero) and the system is more stable, then it becomes more difficult (probabilistically) to escape a state (i.e. a lineup) for a state with a lower expected run differential. Under suitable conditions, the simulated annealing algorithm converges to an optimal lineup. Practically, the algorithm terminates after a fixed number of iterations or when there are infrequent state changes. The asymptotic results suggest that the final state will be nearly optimal.

The fine tuning of the algorithm corresponds to the cooling schedule and the mechanism for generation of candidate lineups. A necessary condition of the asymptotic theory is that all team lineups are "connected". In other words, it must be possible for every team lineup to be reached from any other team lineup in a finite number of steps. A guiding principle in the development of our simulated annealing algorithm is that we enable large transitions (moving to "distant" team lineups) during the early phases of the algorithm and we restrict transitions to neighbouring (i.e. close) lineups during the latter phases of the algorithm.

With the combinatorial space described above, we note that our problem has similarities to the longstanding travelling salesman's problem (TSP) where the potential routes of a salesman consist of permutations of the order of visited cities. In the TSP, given n cities, there are n! orderings in

which the cities may be visited. As this is likewise a large discrete space for even moderate n, there are some useful heuristics in proposing candidate routes. For example, total distance travelled is unlikely to vary greatly if the order of the visited cities is only slightly changed. This introduces a concept of closeness where total distance travelled is close if there are only small changes to the route. This corresponds to small changes in the permutations such as interchanging the order of two adjacent cities. Also, in simulated annealing we want to explore the solution space widely in the initial phase of the search so that with high probablity, we are eventually in the neighbourhood of the global optimum. In the TSP, exploring wide neighbourhoods corresponds to more extreme changes in the permutations of cities visited. Although our problem is more complex than the TSP, we borrow ideas from Aarts and Korst (1989) and Swartz et al. (2006) where simulated annealing has been successfully utilized to address optimization problems for related discrete spaces.

We note that there is considerable flexibility in the proposed procedure. For example, suppose that the captain wants to try out a new player in the 4th position in the batting order. In this case, the proposal distribution of the simulated annealing algorithm can be hard-coded such that the new player is forced into the 4th position, and the remaining lineup is optimized according to this constraint. The introduction of such constraints provides a systematic approach for experimentation with lineups, and is particularly useful with new players who do not have much of a batting/bowling history.

3.1 Fine Tuning of the Algorithm

Recall that the determination of a team lineup has three components:

(a) the selection of 11 players from a roster of available players

(b) the specification of the batting order for the selected 11 players

(c) the specification of the bowling order for the selected 11 players

Our mechanism for the generation of candidate lineups takes these three components into account. In each step of simulated annealing, we generate a random variate $u \sim \text{Uniform}(0, 1)$. If $0 \leq u < 1/3$, we address issue (a). If $1/3 \leq u < 2/3$, we address issue (b). If $2/3 \leq u < 1$, we address issue (c). Specifically, we generate candidate lineups as follows:

(a) Team Selection - Denote the roster of M potential players as $\{i_1, \ldots, i_M\}$ where $i_1, \ldots, i_{11}$ are the players in the current lineup. We randomly choose a player from $i_1, \ldots, i_{11}$ and swap this player with a randomly chosen player from $i_{12}, \ldots, i_M$. The swapped-in player replaces the

swapped-out player in both the batting order and the bowling order. If the swapped-out player is a bowler who is active in the bowling order and the swapped-in player is a pure batsman, then the excess bowling overs are randomly distributed to the current bowlers in the lineup. We do not allow swaps that result in lineups with fewer than five players who can bowl nor do we allow swaps that result in lineups without a wicketkeeper. The resultant lineup is the candidate lineup.

(b) Batting Orders - Denote the batting order corresponding to the current lineup by $i_1, \ldots, i_{11}$ where i_j corresponds to the batsman in the jth batting position. We randomly choose a batsman i_j from $i_1, \ldots, i_{10}$ and then interchange i_j with i_{j+1}. The resultant lineup is the candidate lineup.

(c) Bowling Orders - Suppose that we have M_B potential bowlers in the current lineup. The generation of a candidate bowling order begins with an ordered list of $4M_B$ bowler symbols where each bowler appears in the list four times. The setup therefore enforces one of the T20 rules that no bowler may bowl more than four overs in a match. The first 20 entries in the list define the current bowling order where i_j corresponds to the bowler who bowls during over j. We randomly choose a bowler from $i_1, \ldots, i_{20}$ and then swap him with a randomly chosen player from one of $i_1, \ldots, i_{4M_B}$ According to T20 rules, bowlers are not permitted to bowl in consecutive overs, and we do not allow such a swap. The resultant lineup is the candidate lineup.

In the early stages of the algorithm ($i = 1, \ldots, 500$), we permit double swaps instead of single swaps in steps (a), (b) and (c) to facilitate larger transitions to escape local neighbourhoods. To complete the description of the algorithm, we use an exponential cooling schedule defined by a sequence of temperature plateaux

$$t_i = 0.5(0.9)^{\lfloor i/100 \rfloor} \ . \tag{9}$$

We introduce a provision to the cooling schedule (9) whereby we move to the next temperature plateau if there are more than 20 state changes at any temperature. The rationale is that we do not want to "waste" time at temperatures where we are regularly accepting the candidate lineup (i.e. moving both up and down the hill). At such temperatures, we would not be moving in the direction of an optimal lineup.

As the algorithm proceeds, we also want to make sure that the objective function $\mathrm{E}(D)$ is estimated accurately. We therefore increase the number of innings simulations N_i as the algorithm proceeds. Specifically, we set

$$N_i = \min(25000, 1000 + 16i) \ .$$

For example, $N_1 = 1016$, and when our algorithm has typically converged, $N_{1500} = 25000$.

4 APPLICATIONS

We have run the simulated annealing algorithm for various teams and have obtained sensible yet provocative results in each case. In our testing, we have found that 1500 iterations are sufficient for convergence and this requires approximately 24 hours of computation on a laptop computer.

A large fraction of the computational cost comes from the simulation of innings. Since the number of simulations increases as the algorithm proceeds, so does the cost of each successive iteration. We eliminate unnecessary simulations wherever possible. The expected run differential and the lineup are stored for each proposal that is accepted, so only the newly proposed lineup needs to be simulated during any given iteration. Furthermore, if the batting order is unchanged between the current lineup and the proposed lineup in an iteration, then only the bowling innings is simulated and the current lineup's batting average is reused. Likewise, if the bowling assignment is unchanged in a proposal, then only the batting innings needs to be simulated.

Since the algorithm simulates a large number of independent innings during each iteration, the simulation step is embarrassingly parallelizable, meaning it requires $1/n$ as much time to complete using n identical CPU cores as it would on one such core. We use the Snow package (Tierney et al. 2013) and the Snowfall package (Knaus 2013) in R to parallelize simulation on four cores using an Intel i7 processor. Other steps in the algorithm such as creating candidate lineups and checking them for validity are not parallelizable, and there is a computational cost associated with work assignment and aggregation of output between cores.

In Figure 3, we illustrate a path taken by the simulated annealing algorithm in the search for the optimal Indian lineup described in more detail in Section 4.1. We observe that the starting lineup is far from optimal (i.e. $\mathrm{E}(D_1) \approx 0$) but very quickly iterates to good lineups (i.e. $\mathrm{E}(D_i) > 10$) for larger values of i. Recall that our computational strategy was to not spend much time exploring unsuitable lineups. Hence we see that confidence intervals corresponding to the estimates of $\mathrm{E}(D_i)$ are wider in the early stages of the algorithm where N_i is small. We note that we have confidence in the algorithm since different starting values (i.e. lineups) all lead to the same neighbourhood of solutions. We say "neighbourhood of solutions" since we have found that small lineup changes (e.g. exchanging the batting order of the 9th and 10th batsmen) lead to changes in $\mathrm{E}(D)$ that are not meaningul (i.e. less than 0.5 runs). As the temperature decreases, candidate lineups continue to be proposed but fewer are accepted. Also, as the temperature decreases, the magnitude of the dips in Figure 3 decrease as it becomes less probable that condition (8) is satisfied.

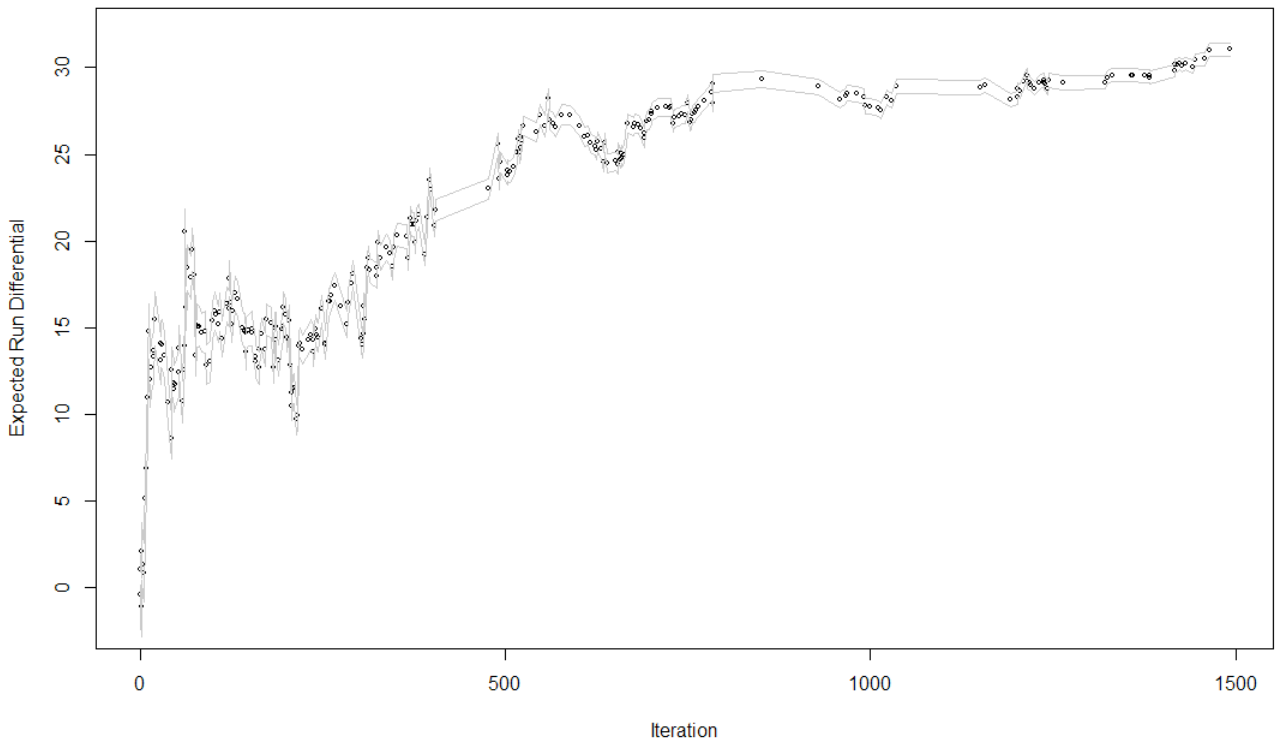

Figure 3: A plot of the estimated run differential versus the iteration number in simulated annealing corresponding to India. Confidence intervals for the estimates are provided.

The algorithm has been designed such that all batsmen are drawn from the same pool. This prevents double drawing of players, and also allows for multiple wicketkeepers to be in the lineup. It is therefore possible that a wicketkeep such as A.B. de Villiers could be selected in his non-core role.

4.1 Optimal T20 Lineup for India

To investigate our methodology, we considered 17 currently active players for India who have a playing history in T20 ICC matches. The optimal lineup (i.e. team selection, batting order and bowling selection) based on the proposed simulated annealing algorithm is given in the first column of Table 1. For comparison, columns 2, 3 and 4 provide actual lineups used by India in the T20 World Cups of 2010, 2012 and 2014, respectively. In general, we see many similarities between the optimal lineup and the lineups used in practice. Also, there seems to be as much variation between the optimal lineup and the actual lineups as between the actual lineups themselves.

	Optimal Lineup	May 11/10 Lineup	Oct 02/12 Lineup	Apr 06/14 Lineup
	01. V Kohli (1)	KD Karthik	G Gambhir	RG Sharma
	02. G Gambhir	G Gambhir	V Sehwag	AM Rahane
	03. S Raina (3)	S Raina	V Kohli	V Kohli
	04. Y Singh (4)	MS Dhoni*	RG Sharma (1)	Y Singh
	05. V Sehwag	Y Singh (1)	Y Singh (4)	MS Dhoni*
	06. RG Sharma	YK Pathan (3)	S Raina	S Raina (4)
	07. YK Pathan	RG Sharma	MS Dhoni*	R Ashwin (4)
	08. H Singh (4)	PP Chawla (4)	IK Pathan (3)	RA Jadeja (1)
	09. MS Dhoni*	H Singh (4)	R Ashwin (4)	A Mishra (4)
	10. A Mishra (4)	V Kumar (4)	L Balaji (4)	B Kumar (3)
	11. B Kumar (4)	A Nehra (4)	Z Khan (4)	MM Sharma (2)
	NS Z Khan			
	NS R Ashwin			
	NS R Jadeja			
	NS V Kumar			
	NS S Dhawan			
	NS AM Rahane			
E(D)	32.4	2.9	7.4	1.1

Table 1: Optimal lineup for India and three typical lineups that were used on the specified dates. The vertical numbering corresponds to the batting order where the players labelled NS were not selected. In parentheses, we provide the number of overs of bowling and the asterisk denotes the wicketkeeper.

However, there are a number of interesting discrepancies between the optimal lineup and the actual lineups used by India. For example, we observe that India may not be positioning M.S. Dhoni properly in the batting order. Whereas India places him in the middle batting positions (No 4, 5 and 7), they may be better served to have him bat later, such as in the 9th position as specified in the optimal lineup. This is suggestive that Dhoni (who is both captain and wicketkeeper) may be overrated as a batsman. It is also interesting to observe that the optimal lineup has only one fast bowler, namely B. Kumar. In each of the actual lineups there are two fast bowlers. In cricket, there is a tradition of mixing the bowling sequence between fast and spin bowlers. However, from an analytics point of view, if your team does not have many good fast bowlers (for example), then maybe the composition should be reexamined to include more spin bowlers. Sometimes it is useful to question tradition and folklore in the presence of hard numbers.

One question of interest is whether the optimal T20 lineup has a different composition in terms of the number of pure batsmen, all-rounders and bowlers that are used in actual T20 matches

and actual ODI matches. In analyzing actual matches, we found that there is considerable variability in this regard. The variability is both across matches and across teams. Complicating the analysis is the definition of "all-rounder". An all-rounder is an ambiguous term as it suggests that a player is both good at batting and bowling. But technically, any player who bowls is forced to bat occasionally, and therefore, whether or not he is considered an all-rounder is subject to some debate. Therefore, in terms of team composition, the main take-away point is that teams should play their optimal lineup even though it may sometimes result in nontraditional uses of players. The continuum of abilities suggested in Figure 1 and Figure 2 lead credence to the idea that there are not always clear delineations between pure batsmen, all-rounders and bowlers.

It is worth commenting that E(D) for the optimal T20 lineup is remarkably larger than for the actual lineups in Table 1. While this is a promising reason to consider the proposed methodology, there may be various nuances involved in this observation. First, there are important changes that collectively explain the 30-run gap between the optimal lineup and the ones that were actually used. For example, M.M. Sharma was unknown before the 2014 World Cup match, and his limited international T20 information implies that he is worse than an average bowler. Hence, his inclusion in the April 6/14 lineup reduced E(D). India included Sharma in their roster despite his limited international experience. It was Sharma's stellar performance in other leagues (e.g. IPL) and one-day cricket which earned him the right to bowl in a Twenty20 World Cup final. Other notable inclusions in the optimal lineup who did not play in 2014 are Gambhir, Sehwag, Pathan and Singh. Also, we recall that the timeframe of our dataset used for estimating player characteristics was 2005-2014. It is therefore likely that some of the players who were selected in the actual lineups were experiencing a good run of form, perhaps better than their historical characteristics which were used in obtaining E(D). Related to this comment, G. Ghambir and V. Sehwag were included in our optimal squad but not in 2014. By 2014, they were likely perceived as aging players with sub-optimal performance.

4.2 Optimal T20 Lineup for South Africa

Similarly, we carried out the optimization procedure for South Africa. This time, 15 current players were considered for team selection. The optimal lineup and the actual lineups (used in the three most recent World Cups) are given in Table 2. The optimal lineup ranges from 16.8 runs to 21.9 runs superior to the actual lineups.

A significant difference between Table 2 and Table 1 is that the South African optimal lineup has three fast bowlers (Morkel, Tsotsobe and Steyn) compared to India's one fast bowler. This implies that the South African fast bowlers are more effective than their Indian counterparts. And,

	Optimal Lineup	May 10/10 Lineup	Oct 02/12 Lineup	Apr 04/14 Lineup
01.	H Amla	HH Gibbs	H Amla	Q de Kock*
02.	AB de Villiers*	GC Smith	JH Kallis (3)	H Amla
03.	JP Duminy	JH Kallis (4)	AB de Villiers*	F du Plessis
04.	F du Plessis (4)	AB de Villiers	F du Plessis (1)	JP Duminy (3)
05.	F Behardien	JP Duminy	JP Duminy (1)	AB de Villiers
06.	D Miller	MV Boucher*	F Behardien	D Miller
07.	JA Morkel	JA Morkel (4)	RJ Peterson (4)	JA Morkel (2)
08.	M Morkel (4)	J Botha (2)	JA Morkel	D Steyn (3)
09.	LL Tsotsobe (4)	RE van der Merwe (2)	J Botha (3)	BE Hendricks (4)
10.	I Tahir (4)	D Steyn (4)	D Steyn (4)	I Tahir (4)
11.	D Steyn (4)	CK Langeveldt (4)	M Morkel (4)	WD Parnell (3)
NS	Q de Kock			
NS	BE Hendricks			
NS	WD Parnell			
NS	AM Phangiso			
E(D)	36.6	14.7	19.8	14.7

Table 2: Optimal lineup for South Africa and three typical lineups that were used on the specified dates. The vertical numbering corresponds to the batting order where the players labelled NS were not selected. In parentheses, we provide the number of overs of bowling and the asterisk denotes the wicketkeeper.

this is again suggestive that teams should consider the best players available for team selection, and not be tied to a tradition of having an even mix of fast and spin bowlers.

Perhaps the largest discrepancy between the optimal lineup and the 2014 actual lineup is that Q. de Kock was not selected to the optimal lineup whereas he is the opening batsman in 2014. We note that de Kock did not play all that well in the World Cup scoring only 64 runs in five innings. Also, the optimal lineup appears to place more value on A.B. de Villiers as he is selected as an opener compared to No 5 in the 2014 lineup.

4.3 All-Star Lineup

To our knowledge, the determination of an "all-star" team which attracts interest in many team sports (e.g. the National Basketball Association, the National Hockey League, etc.) is not something that is common to cricket. We have therefore carried out this novel exercise by including a larger pool of 25 widely recognized and current T20 players for team selection. This provided a further test of the algorithm to see that the size of the player candidate pool did not impose an impediment to obtaining an optimal lineup.

The optimal all-star lineup is shown in Table 3. Although there is no way of assessing whether the optimal lineup is "correct", our cricketing intuition is that the resultant all-star team is very strong. It is interesting to note that four of the selected players are from the West Indies, of whom three are all-rounders and are well-known power hitters. The West Indies have been a strong T20 team in recent years, winning the 2012 World Cup and reaching the semi-finals in the 2014 World Cup.

The fact that M.S. Dhoni was not selected to the all-star team reinforces the observation that he was placed down at No 9 in the optimal Indian batting order. We note that although V. Kohli is thought by some to be the best current Indian cricketer, he was not selected to the all-star team, yet his teammate, S. Raina was selected. It was also interesting to observe that B.B. McCullum was selected as the wicketkeeper instead of A.B. de Villiers who is known as a great power hitter. The all-star team had an impressive expected run differential of $\mathrm{E}(D) = 61.6$.

Optimal Lineup	Pure Batsmen not Selected	Bowlers & All-Rounders & Wicketkeepers not Selected
01. AJ Finch, Aus	G Bailey, Aus	S Afridi, Pak
02. BB McCullum, NZ*	MJ Guptill, NZ	S Ajmal, Pak
03. NLTC Perera, SL (2)	AD Hales, Eng	K Akmal, Pak*
04. G Maxwell, Aus (2)	R Levi, SA	C Anderson, NZ
05. C Gayle, WI	D Miller, SA	DJ Bravo, WI
06. K Pollard, WI	A Shehzad, Pak	SCJ Broad, Eng
07. D Sammy, WI (4)	R Taylor, NZ	AB de Villiers, SA*
08. S Raina, Ind		MS Dhoni, Ind*
09. F du Plessis, SA (4)		JP Duminy, SA
10. S Badree, WI (4)		U Gul, Pak
11. D Steyn, SA (4)		M Johnson, Aus
		V Kohli, Ind
		L Malinga, SL
		S Narine, WI
		D Ramdin, WI*
		KC Sangakkara, SL*
		S Watson, Aus

Table 3: All-Star lineup including players not selected. In parentheses, we provide the number of overs of bowling and asterisks denote wicketkeepers.

5 DISCUSSION

This paper provides a powerful methodology for the joint problem of determining team selection, the optimal batting order and the optimal bowling order in Twenty20 cricket. No such work has been previously carried out on this signficant problem. Moreover, the approach is based strictly on analytics, avoiding opinion, folklore and tradition.

One of the features of the proposed approach is that teams are free to modify player characteristics. Perhaps a player is in particularly good form and it is believed that his probability of dismissal is reduced. It is a simple matter of lowering his value of p_{i707} in (3) and observe what optimal lineup is obtained. Alternatively, one can hard-code a subset of players into the lineup and build the remainder of the roster around them.

It is our hope that papers like this will help promote the adoption of analytics in cricket. Big money is now being spent in leagues such as the IPL, and it is only sensible to make use of the knowledge contained in data.

6 REFERENCES

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines.* New York: Wiley.

Barr, G.D.I. and Kantor, B.S. (2004). A criterion for comparing and selecting batsmen in limited overs cricket. *Journal of the Operational Research Society*, 55, 1266-1274.

Bretteny, W. (2010). Integer optimization for the selection of a fantasy league cricket team. *MSc Thesis*, Nelson Mandela Metropolitan University.

Davis, J., Perera, H. and Swartz, T.B. (2015). A simulator for Twenty20 cricket. *Australian & New Zealand Journal of Statistics*, 57, 55-71.

Gramacy, R.B., Taddy, M.A. and Jensen, S.T. (2013). Estimating player contribution in hockey with regularized logistic regression. *Journal of Quantitative Analysis in Sports*, 97-112.

Kirkpatrick, S., Gelatt Jr. C.D. and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.

Knaus, J. (2013). Easier cluster computing (based on snow), version 1.84-6.

Lemmer, H.H. (2013). Team selection after a short cricket series. *European Journal of Sport Science*, 13, 200-206.

Schuckers, M.E., Lock, D.F., Wells, C., Knickerbocker, C.J. and Lock, R.H. (2011). National Hockey League skater ratings based upon all on-ice events: An adjusted minus/plus probability (AMPP) approach. Unpublished manuscript.

Swartz, T.B., Gill, P.S., Beaudoin, D. and de Silva, B.M. (2006). Optimal batting orders in one-day cricket. *Computers and Operations Research*, 33, 1939-1950.

Tierney, L., Rossini, A.J., Li, N. and Sevcikova, H. (2013). snow: Simple Network of Workstations, version 0.3-13.

Woolner, K. (2002). Understanding and measuring replacement level. In *Baseball Prospectus 2002*, J. Sheehan (editor), Brassey's Inc: Dulles, Virginia, 55-66.

7 APPENDIX

The major goal of this paper is the determination of optimal lineups for Twenty20 cricket sides. In (5), we provide a simple upper bound for the cardinality of the solution space for the optimization problem. For various reasons, there are some lineups which we exclude as possibilities from the solution space. We discuss this here and provide an improved upper bound for the cardinality of the solution space.

We expand on our earlier notation and let $M = M_1 + M_2 + M_3$ denote the number of players that are available in the team selection pool where M_1 is the number of wicketkeepers, M_2 is the number of non-wicketkeepers who are pure batsmen and M_3 is the number of non-wicketkeepers who are able to bowl. From these players, an optimal lineup of 11 players is chosen where $m_1 + m_2 + m_3 = 11$ using the obvious notation for m_1, m_2 and m_3.

Although there is no formal rule in cricket that prevents two wicketkeepers from playing at the same time, we assume that $m_1 = 1$. This assumption is in accordance with the way that cricket is played in practice. And since only one wicketkeeper is selected, it follows that this wicketkeeper does not bowl (for if he did, there would be no available wicketkeeper). Also, we recall that no player may bowl more than four overs (i.e. this implies that there must be at least five players who are able to bowl). Therefore, the selection of the 11 active players can be chosen in

$$\sum_{A} \binom{M_1}{1}\binom{M_2}{m_2}\binom{M_3}{m_3}$$

ways where $A = \{\ (m_2, m_3)\ :\ 1 + m_2 + m_3 = 11,\ m_2 \leq M_2,\ 5 \leq m_3 \leq M_3\ \}$.

For batting, there are 11! possible batting orders given the team selection. For bowling, given that m_3 bowlers have been selected, let i_j denote the number of overs bowled by bowler j. Then $i_1, \ldots, i_{m_3}$ are restricted according to the set $B = \{\ i_j \leq 4\ :\ i_1 + \cdots + i_{m_3} = 20\ \}$. Given $i_1, \ldots, i_{m_3}$, there are $\frac{20!}{i_1! \cdots i_{m_3}!}$ indistinguishable orderings of the bowlers. However this term is an upper bound for the number of bowling orders given $i_1, \ldots, i_{m_3}$ because it does not take into account the restriction that no bowler is allowed to bowl in consecutive overs. Unfortunately, we were unable

to derive a combinatorial expression for the number of distinct orderings of $i_1 + \cdots + i_{m_3} = 20$ symbols where i_j symbols are of type j and no two symbols may be ordered consecutively.

Putting all three lineup components together, an improved upper bound for the cardinality of the solution space is given by

$$\sum_{A} \left[\binom{M_1}{1}\binom{M_2}{m_2}\binom{M_3}{m_3} \, (11!) \sum_{B} \, \frac{20!}{i_1! \cdots i_{m_3}!} \right] .$$