Red Sox Order Optimized
By Dave St.

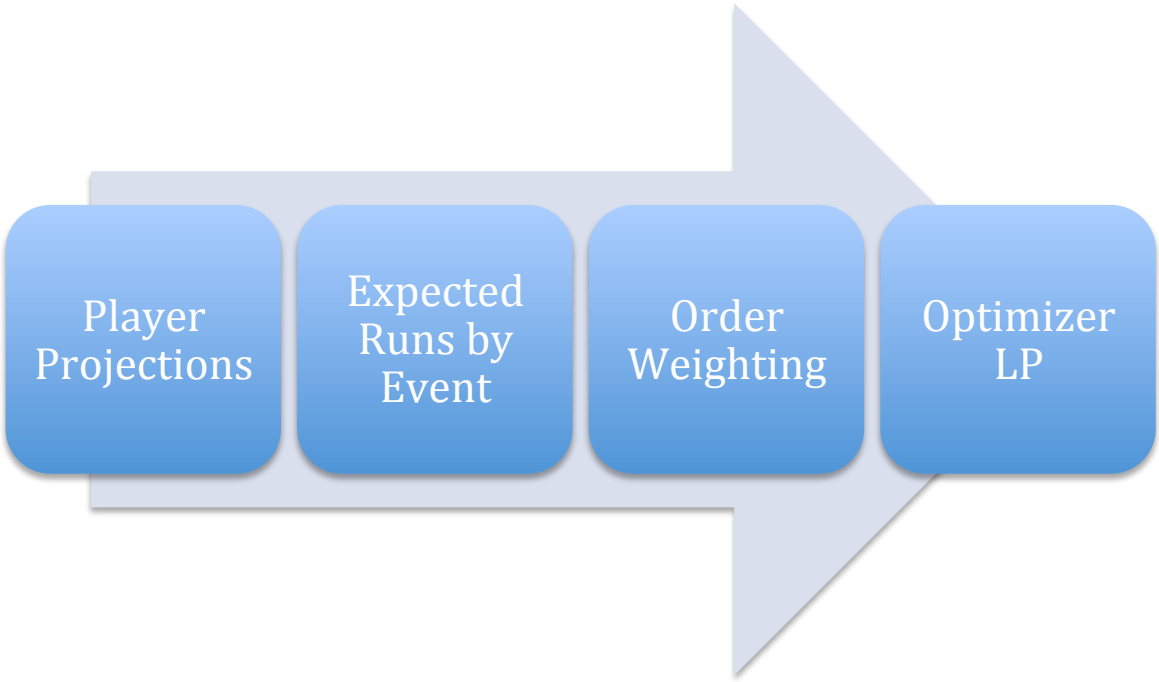
It's been a cold winter up here in New England with yet another snowstorm predicted for tonight. At least baseball season started a week or so ago thanks to the NY Jets (oh wait that was bad). The Hot Stove has been cooking big time up here for a variety of reasons.

The hottest topic of discussion in Red Sox Nation in the new year is the Sox's batting order. Believe me everybody has an opinion on the Sox's batting order. And virtually every opinion stated is driven almost entirely by something that was described in Cyril Morong's excellent post linked through "Batting Order philosophy… from a century ago" (find it linked here http://www.insidethebook.com/ee/index.php/category/Batting_Order/). You want to know how to get people to look at you cross-eyed, just ask them if "run-expectancy" should be a consideration.

I should mention that I am a management consultant and use regression analysis, simulation, and linear programming as part of my workday. I've also learned over the years that everybody loves to pontificate but that client's actually pay for analytically driven insights.

So here's my opportunity to put forth an opinion on the Red Sox's batting order that is driven by "run-expectancy" and pure analytics.

My approach was rather straightforward and should be very familiar to the readers of this website. A simple diagram of the process is shown below (remember the management consultant confession).



Each of the steps is described starting below.

Player Projections

I used player projections from The Hardball Times for 2011 as my starting point (1/26/2011 – The Hardball Times sells this info so I'm not including the actually table here). I converted the player projections into percentages based on plate appearance for each of the batting related outcomes and developed a matrix of run expectancies for each of the expected members of the Red Sox starting line-up. The matrix included percentages for each plate appearance for singles, doubles, triples, homeruns, walks, hit by pitch, strikeouts, and non-k outs. For each player the percentages sum to one (or one thousand in baseball speak). Sorry Jacoby Ellsbury but in this pass I ignored stolen bases (same to Carl Crawford and Dustin Pedroia). I suspect the stolen base issue will not be a major impact but will be discussed later.

Expected Runs by Event and Order Weighting

The expected runs by event with order weighting are straight from ***The Book*** (Table 51 page 129 - buy the book to see the table). Order weighting reflects that the leadoff batter is expected to have more plate appearances per game than the ninth batter. Is there an update to this table? Would an update be any different?

Optimizer LP

The last step involves multiplying the player expectation (in percentages) by the run value per event for the appropriate batting position and summing the outcomes to arrive at a team run expectancy for the order. Simulation (making up your own order) allows you to see the team run expectancy for your selected order.

Unfortunately there are 362,880 potential batting orders (9! As in nine factorial or 9x8x7x6x5x4x3x2x1=362,880) to consider (ignoring all players beyond the starting nine). Linear programming allows you to find the best combination (the objective function is highest team run expectancy) very quickly. I did all of this in Excel using the SOLVER add-in (free version from Frontline – www.solver.com).

Highest Expected Run Producing Red Sox Batting Order

The magic "black box" result provided the following batting order:

Batting Position	Player	Bats
1	Kevin Youkilis	Right
2	Adrian Gonzalez	Left
3	David Ortiz	Left
4	Dustin Pedroia	Right
5	Carl Crawford	Left
6	J.D Drew	Left
7	J. Saltalamacchia	Switch
8	Marco Scutaro	Right
9	Jacoby Ellsbury	Left

Many observations occurred to me during this process (yup gonna use bullet points – remember the management consultant thing):

- Some players have positive run expectancies (e.g., Adrian Gonzalez and Kevin Youkilis) and some have negative run expectancies per plate appearance.
- Traditionalists (i.e., those that think "Batting Order philosophy… from a century ago" is the best current thinking) will object to a lot of things in this batting order (e.g., the number of back-to-back lefties, slower runners at the top of the order, using "base stealers" to force pitchers to throw fastballs to fastball hitters, etc.).
- It is easy to add constraints in the linear programming process to find the best solution given a "managerial" commandment (e.g., Ellsbury must lead off).
- You really should have your "best" hitters/on base machines at the top of the order.
- I was surprised that the highest expected run production came with David Ortiz batting third. That's so 2007. I suspect Dustin Pedroia would like to see that he is slotted to hit clean-up, although for all the wrong reasons.
- Batting order matters. You can minimize run expectancy as well using linear programming and the difference is dramtatic.

Arguing that Crawford and Ellsbury should bat higher in the order may be putting too much value on base stealing. My optimization ignores base stealing. Simple attempts to include the impact of base stealing (adding the run expectancy of base stealing less caught base stealing based on forecast percentages of base stealing attempts to singles or increasing forecast singles to doubles less a run expectancy factor did not change the order. My optimization also ignores left/right batting splits and, therefore, does not factor in whether the opposing pitcher is a lefty or righty.

My main goal was to use a "process" for optimizing the batting order to maximize run production. I wonder how many teams actually such an approach?

It also occurred to me that linear programming could be used to optimize the use of relief pitchers. Relief pitcher usage is a classic limited resource with variable value allocation issue (high leverage situations occurring during a season versus limited "warm-ups" or batters faced constraints of each pitcher). Linear Programming is also fast enough that it could potentially be used as an input to tactical in game decision making.

Thoughts?