NLP – races or tournament

# Overview

The design of an Open Innovation Contest has often to do with a choice between two archetypical architectures or formats: the race and the tournament.

Broadly speaking, races are competitions where the first contestant to solve a given problem or to hit a certain level of performance is awarded a prize. There are many obvious examples in sport competitions; the ``Netflix Prize’’ --- a popular contest to improve the performance of Netflix’s own recommendation system by 10% --- can perhaps be listed as the first example of a race in crowdsourcing competitions.

Tournaments are, instead, timed competitions where the participant who produced the ``best’’ performance (relative to other candidates and within the given time) wins. This format is very common and has been adopted by many crowdsourcing platforms such as *Innocentive, Kaggle, Threadless, TopCoder* which are hosting a stream of tournament-like competitions to produce all kinds of innovations (e.g., algorithms, artistic designs, software).

Yet, which architecture to choose is not a simple question. Tournaments and races would provide participants with very different incentives and possibly foster different outcomes.

Here we list a few key differences:

* **Incentives and quality**. Assuming contestants would exert effort to maximize the probability of winning the prize, tournament participants would care about their position in the final ranking but would not be necessarily worried about low-quality performances as soon as the others are doing poorly. On the contrary, participants in a race would be required to hit a certain quality level regardless of the performances of the others, but they may not want to improve their own performance any further once someone hit the desired level.
* **Management of time**. While a race would obviously impose participants to exert high effort from the first day, this is not necessarily the case in tournaments.
* **The role of feedbacks**. If partial scores are disclosed (e.g., through a leaderboard), a tournament might induce contestants to hide their progress until the last day of the competition as opposed to races where each attempt of reasonable quality should be submitted (and so disclosed) as soon as possible. Here the economic rationale would be that an early disclosure can spur higher effort from rivals.

So we propose to investigate the impact of these incentives on

* Performance
  + Both quality and the diversity of solutions
* Participation
  + Measures of effort (other than quality)

Our experimental design:

* We would recruit *N* participants from TopCoder’s member-base
* Participants would be competing to solve a common problem over a period of *T* weeks
* We would split them into homogenous rooms of different sizes (e.g., 8 and 30)
* Each room would then be randomly assigned to either a race or a tournament
* During the competition, we would keep truck of the extent of participation, the number of submissions, the quality, etc.

# TODO

1. Prepare 4 slides for the experimental design
   1. Races v.s. Tournaments
      1. Practical implications (hit a threshold)
      2. Theoretical implications (doing effort today or later?)
      3. Under what conditions (N-effect)
2. Simulate data with USPTO numbers
   1. Quality = b1 \* X + b2 \* X\_group + b3 \* Size + format (here suppose format has an effect)
   2. Random extract quality for 100 times
      1. if Q > bar then stop in the races
3. Discuss with Rinat actual implementation
   1. Using annotated

# Overview

The design of an Open Innovation Contest has often to do with a choice between two archetypical architectures or formats: the race and the tournament.

Races are competitions where the first contestant to solve a given problem or to hit a certain level of performance is awarded a prize. There are many obvious examples in sport competitions, but the ``Netflix Prize’’ --- a popular contest to improve Netflix’s own recommendation system by a fixed amount (i.e., 10%) --- can perhaps be listed as the first example of a race in the era of crowdsourcing competitions.

Tournaments are, by contrast, competitions where the winner is the participant who produced the ``best’’ performance (relative to other candidates) within a fixed term. This format is very common and has been adopted by many crowdsourcing platforms such as *Innocentive, Kaggle, Threadless, TopCoder* which are hosting a stream of tournament-like competitions to produce all kinds of innovations (e.g., algorithms, artistic designs, software).

Yet, which architecture to choose is not a simple question to make. Assuming participants would exert effort to maximize the probability of winning the prize, tournaments and races would provide participants with different incentives and possibly foster different outcomes.

Here we list a few differences:

* While tournament participants would care about their position in the final ranking but would not be necessarily worried about low-quality performances as soon as the others are doing poorly, on the contrary participants in a race would be required to hit a certain quality level (no matter what the others are doing) but this requirement comes at the cost of cutting off incentives to improve the performance beyond that level.
* Individual management of time would be different. While a race imposes to exert high effort from the first day, this is not necessarily the case in tournaments.
* If partial scores are disclosed in real-time (e.g., through a leaderboard), tournaments might induce contestants to hide their progress until the last day of the competition as opposed to races where each attempt of reasonable quality should be submitted (and so disclosed) as soon as possible.

So we propose to investigate the impact of these diverse incentives on

* Performance
* Participation

And to study any variation in the magnitude of the effects by changing the number of contestants.

Our experimental design:

* We would recruit 400/600 participants on TopCoder to solve a common problem.
* We would split them into homogenous rooms of different sizes (e.g., 8 and 30).
* Each room would then be randomly assigned either to a race or to a tournament.
* Contestant would compete over a period of N weeks
  + We would keep truck of the extent of participation, the number of submissions, the quality, etc.

Possible theoretical issues:

* Assume that
* **In a tournament** contestants would care about their performance’s position in the final ranking, but not about the level of the performance per se
  + Output should be a function of the perceived ``intensity of competition’’
  + If competition is ``low’’, there might not be enough incentives to max. quality
* **In a race**, the perceived intensity of competition would affect participation rates and the decisions on when exerting effort but not necessarily quality (as now participants need to hit a certain quality threshold to be awarded a prize)
  + At the same time, once someone hit the threshold the competition ends

In the race we sacrifice a certain freedom in the quality level to […]

We might argue that even in the absence of any specific benefit from an early innovation, a race would guarantee quality at the cost of some uncertainty.

As N increases

[…]

Suppose that you

would not be willing to improve their solutions any further whenever someone else hit the target quality level. Similarly, contestants would not be willing to improve their solutions as they are the best relative to others.

In the race scenario contestants would hardly decide to improve the solution beyond the stated threshold and we would not be able to see what would be the quality if we had not set a threshold.

Under the tournament scenario, contestants care to be first relative to the other participants. So that if participation is not … we would not be able to tell what … would do.

Here we face a risk that with the same prize money contesta

Even in the absence of any precise criterion for evaluating the benefits from having a good solution sooner rather than later, and so even disregarding time as of any value at all, the incentives provided under these two … are very different.

In particular there is a trade-off from the uncertainty of the solution and the quality of the solution.

Under uncertainty of the skills, the

NOTE, the first to hit a threshold but if no one is hitting the threshold then all would be awarded a prize on the basis of the rank-ordered of solutions.

The N-effect.

As N grows, competition in the tournament is tougher for high-types.

As N grows, competition in the race is tougher for high-types but also the likelihood that the threshold is hit is increasing. (a problem is that you are forced to do things quicker 🡪 less experimentation).

We want to do *a prior* power analysis to figure out the right sample sizes.

The goal of the project is to test the contest architecture on the outcomes of the crowd.

Specifically, **races** or **tournaments**?

## Phase 1 – qualification round

* A ``race’’ is such that the first N\* participants that would have hit a quality threshold would qualify for the second phase of the competition. The others will be excluded (if less than the right number of submissions, then those with the highest skill rating would qualify).
* A ``tournament’’ is such that within a week of time the best N\* submissions would qualify for participation to the next phase.
* Besides qualification, contestants will also compete for room-prizes (e.g., cash)
* Depending on participation we randomly form rooms of equal size
  + For example, 500 participants are sorted into rooms of 20 (e.g., 50 rooms)
  + Half of the rooms are assigned to a race

## Phase 2 – final round

Qualified coders will go on with standard marathon match for grand prizes.

We ask participants (when they submit) to ``tag’’ their solutions with the technique they are using.

Why we need the first stage. To keep separate incentives for qualification and racing from the final incentives. Second is that the threshold might be miss-specified there is uncertainty (we can set different thresholds if we have multiple rooms)

This first qualification stage would allow us to eventually change the problem statement

## Phase 3 – analysis of outcomes

Outcomes to evaluate (at the room level)

* Number of submissions
* quality of submissions
  + it could be that the tournament will provide higher scores for the top guys but lower for those who qualify
* how to care about time?
* Selection of coders for the next phase. We can look at the phase two submissions to evaluate the selection of good coders into the competition
* Qualification stage has effect
* Diversity of approaches in races vs tournament

Alternative is ``milestones’’ in a four week marathon match

* After every week we will be awarding a milestone prize within each competition room
  + To the first to hit a threshold (e.g., race)
  + To the best solution of the week (e.g., tournament)
* The problem is that then is just a matter of getting more effort

Industry of crowdsourcing has now set a standard of open innovation contest.

* Yet, under certain circumstances it is not obvious whether the