The Effects of Increasing Competition and Uncertainty on Incentives and Extreme-Value Outcomes in Innovation Contests

Kevin Boudreau (London Business School)
Nicola Lacetera (University of Toronto)
Karim Lakhani (Harvard Business School)

Version: July 15, 2010
Working Paper – Please do not cite without permission

Abstract

Contests are an historically important and increasingly popular economic institution for eliciting innovations. A central concern in designing innovation contests is the number of competitors to admit to a contest. Using a unique data set of 9,661 contests we provide empirical evidence of two coexisting and opposing forces that operate when the number of competitors increases. Greater rivalry reduces the incentives to expend effort by individual solvers across competitors of all skill levels. Despite this overall reduction in the expected performance, adding competitors increases the likelihood that at least one competitor will find a high-quality solution—an extreme outcome. We show that the effort-reducing effect of greater rivalry dominates for less uncertain problems, whereas the effect on the extreme value prevails for more uncertain problems. Adding competitors thus systematically increased overall contest performance for high uncertainty problems, but decreased performance for low uncertainty problems. We also find that higher uncertainty dampens the negative effect of added competitors on incentives. We discuss the implications of our findings for both the theory and practice of innovation contests.

Key words: Innovation contests, competition, institutions of innovation, open and distributed innovation, problem-solving, extreme values
1. Introduction

Contests are a well-established economic institution for eliciting innovation (Terwiesch and Ulrich, 2009; Terwiesch and Xu, 2008). A basic and long-standing question within the literature and practice has been: “Just how ‘big’ should an innovation contest be?” in terms of number of competitors (Che and Gale, 2003; Fullerton and McAfee, 1999; Taylor, 1995; Terwiesch and Xu, 2008). Research in economics suggests that allowing increasing numbers of competitors to participate in a contest will reduce the likelihood of any one competitor winning, reduce each competitor’s incentives to invest or exert effort in creating an effective solution, and thus lower overall innovation outcomes (Che and Gale, 2003; Fullerton and McAfee, 1999; Taylor, 1995).\(^1\) Similar predictions and findings on the negative incentive effect have been found in sociological and psychological research on contest performance (recent examples include Bothner et al., 2007; and Garcia and Tor, 2009). Beyond preserving incentives, another reason to limit entry in a contest is to decrease the cost to the contest organizer conducting and evaluating the competition (Fullerton and McAfee 1999). Thus, competitors may be pre-vetted and qualified before being allowed to participate.\(^2\) The economics literature has, hence, generally recommended against free and open entry into innovation contests, with some models specifically determining the ideal number of competitors to be two (Che and Gale, 2003; Fullerton and McAfee, 1999). If adding competitors reduces incentives and raises organizing costs, there would seem to be little reason for contest organizers to encourage large numbers of competitors to enter their innovation contests.

Although there are, indeed, numerous examples of contest sponsors deliberately restricting the size of their contests (cf. McKinsey and Company, 2009; Nasar, 1999), historical and modern examples of innovation contests include many instances of large numbers of competing solvers—from tens to hundreds to even thousands of competitors in a single contest. Going back to the 15\(^{th}\) century, the office responsible for the construction of Florence’s new cathedral, Santa Maria del Fiore, announced on August 19, 1418 a contest to solve a fifty-year-old architectural puzzle: the creation of the world’s widest and tallest dome with an open invitation for anyone to participate. The organizers received more than a dozen design proposals and deliberated for more than a year before selecting, from an unexpected source, goldsmith and clockmaker Filippo Brunelleschi. He produced a radical approach that eschewed the then

---

\(^1\) Although we now have theories in other sorts of institutional settings that suggest that more competition may be associated with higher effort and greater incentives under certain conditions (e.g., Aghion et al. (2005)), all models of one-shot contests predict a negative effect (so long as there is at least some minimal initial level of competition).

\(^2\) This tends to be the case in architectural competitions in which limited and invited-only contests are often recommended (Nasar, 1999).
typical use of centering and internal supports for the construction of the massive dome (King, 2000). When Canadian mining company Goldcorp announced in 2000 a $500,000 contest aimed at discovering new gold targets in a low performing Northern Ontario mine, it, too, encouraged widespread entry. The contest attracted more than 1,400 participants and led to the discovery of a remarkable 44 productive new targets (Tapscott and Williams, 2006). The goal of the 2006 Netflix Prize contest was the development of software that would achieve a 10% improvement in the DVD rental firm’s algorithm-based movie recommendation system. A prize of $1M was set and a sweeping invitation extended to anyone to participate. The contest ultimately received 44,014 valid programming submissions from 5,169 teams.² Apart from these ad hoc contests, firms have begun to set up contest platforms as an on-going business model. InnoCentive.com, for example, routinely attracts roughly 250 individuals to contests involving R&D-related scientific problem solving on behalf of its clients (Jeppesen and Lakhani, 2010). Thus, rather than restrict entry, the tendency has been to open innovation contests to all comers. Why is this so? This would appear to contradict mainstream theory.

One possible explanation relates to the simple fact that the quality of any one solution, including the solution developed by the eventual winner, depends not just on how much effort that competitor exerts, or even just on the competitor’s skills or aptitude. There may remain substantial uncertainty regarding precisely how best to solve an innovation problem. It may inherently require a novel solution, one that has to be discovered. Precisely who will win and the approach of the winning solution may be hard to anticipate before the fact. Thus, a large number of competitors in an innovation contest might simply increase the likelihood that one will develop a solution that far outperforms all other attempts, in other words: an extreme value outcome. This perspective is consistent with the literature on innovation that posits that innovation efforts are fraught with uncertainty, and that increasing number of independent experiments pursuing parallel paths along the technical frontier improves the chances of finding an extreme outcome (Abernathy and Rosenbloom, 1969; Dahan and Mendelson, 2001; Nelson, 1961). Hence, we refer to the possibility that adding greater numbers of competitors will lead to a greater chance of extreme outcomes as the “parallel path” effect. This might be particularly important, as innovation managers in general, and contest organizers in particular, may care about the maximum or best innovation performance above anything else (Dahan and Mendelson, 2001; Girotra et al., 2010; Terwiesch and Ulrich, 2009).

Previously, analysis of parallel paths and incentive effects proceeded in largely independent literatures. It fell to Terwiesch and Xu (2008) to effectively merge the analysis of parallel path effects and

² Data obtained from http://www.netflixprize.com//leaderboard.
incentive effects within an integrated analytical framework. This required merging the order statistic modeling apparatus of parallel path models with systematic modeling of strategic interactions and incentives. Terwiesch and Xu (2008) argued that adding greater numbers of competitors should generate a tension between the negative effects on incentives and positive effects of parallel paths, leading to particular instances in which free entry or limited entry would generate better outcomes depending on the particular parameters in their model. The analysis also highlighted the importance of the maximum or winning score in a contest. If such a tension were to exist and be empirically relevant, it would imply an important reason why the optimal size of a contest should be larger than the mainstream economic analyses of incentives suggests. It might also imply a need for greater focus on the stochastic nature of the innovation processes when predicting the optimal design of innovation contests.

Given the potential importance of these effects on our central question of how big a contest should be, our goal is to empirically explore the most basic features of a possible tradeoff and the interplay between incentives and parallel path effects, thus providing an empirical foundation for recent theoretical advances. We study three fundamental and related issues. (1) Are incentive and parallel path effects truly of comparable magnitude and do they, consequently, need to be explicitly considered together when designing contests? (2) Do the incentive and parallel path effects work as simply as has been theorized, one effect dampening, the other stimulating, innovation? (3) Under what conditions might one effect dominate the other, perhaps changing how we think of the optimal number of competitors under different situations?

We analyze 9,661 competitions related to the solution of 645 problems from TopCoder, a contest platform on which elite software developers compete in regularly held competitions. This environment affords the opportunity to study multiple concurrent contests for the same problem with different numbers of direct competitors. Further, we are able to observe the skill level and quality of the solution for individual contestants. The contests are designed to challenge developers to create novel software algorithms for solving challenging abstract computational problems (the contests are, in fact, called “algorithm contests”) and are similar in nature to the earlier-mentioned Netflix Prize problem and InnoCentive computational challenges.

Our analysis begins by estimating the independent workings of both incentive effects and parallel path effects in the data. We confirm that these effects, when regarded on their own, appear to operate as straightforwardly as is typically predicted in the theoretical literature. We contribute a basic result to the existing empirical literature on contests by showing, through quantile regressions, that the entire distribution of expected outcomes shifted downward with added competitors, as is usually predicted of
incentive effects in one-shot innovation contests. We contribute a basic empirical result to the mostly theoretical work on parallel paths and stochastic modeling of innovation contests by showing that when more competitors were added the maximum score increased *relative* to the expected distribution of outcomes. These results collectively demonstrate that adding competitors, indeed, reduced outcomes (of all competitors) in expectation, but increased the “upside” that at least one competitor would achieve an extreme outcome. More important than this fact alone, our results demonstrate that these effects not only coexist, but were each empirically important and of *comparable magnitudes* in this context. Neither effect could be ignored, and both should be considered if we are to assess the net effect of varying the size of a contest on problem-solving performance (i.e., the best performance within the group of competitors). These findings on their own should serve as a call for greater research into integrating, and examining the interplay between, parallel path effects and strategic incentives in innovation contests.

We then highlight the key role played by uncertainty in moderating these effects and, therefore, in determining just how big an innovation contest should be. In our context, the extent of uncertainty related, in particular, to the most appropriate technical approach or path that should be taken to effectively solve a given problem, and closely associated uncertainty regarding which solver would successfully discover that approach. Most intuitively, added uncertainty positively moderated the parallel path effect, that is, the likelihood of attaining an extreme outcome with added competitors was increased in cases of higher uncertainty. We also found, however, that higher uncertainty dampened the negative effect of added competitors on incentives in this context. Thus, added competitors could lead to improved contest performance, but only in instances of high uncertainty problems that required a greater level of searching for the best approach or path to a solution.

As calls for the use of innovation contests are amplified in both the public and private sectors (Lindegaard, 2010; McKinsey, 2009; National Research Council, 2007; Tapscott and Williams, 2004; White House, 2010), the results described above suggest that considerable sensitivity to the relative importance of parallel path and incentive effects may be needed to properly design contests. On one hand, we might expect that the type of vexing scientific and societal problems that eventually are pushed out to contests (after perhaps failing to find solutions in more traditional problem-solving institutions) might be characterized by considerable uncertainty and thus benefit more from large than from small, focused contests. On the other hand, the proliferation of enduring platforms intended for repeated use might imply contests suited to a wider range of less uncertain problems for which a smaller number of competitors may be most desirable.

Our paper proceeds as follows. Section 2 reviews the relevant literature and develops basic
hypotheses to guide the empirical exploration of the data. Section 3 details the empirical context of our study. Section 4 describes the data and our estimation strategy. The results of the empirical analyses are reported in Section 5. Section 6 summarizes our contribution and offers concluding remarks.

2. Literature and Hypothesis Development

This section reviews the literature on innovation contests, particularly as it centers on the effects of varying numbers of competitors. Our objective in this section is to develop three basic empirical hypotheses that will serve as a baseline guiding set of predictions as we explore the nature of incentive and parallel path effects.

2.1. Contests and Incentives

Contests and relative performance evaluation mechanisms have received considerable attention in the economics literature, with examples drawn from political decision making, internal labor markets, sales performance contests, and sporting outcomes (Casas-Arce and Martinez-Jerez, 2009; Holmstrom, 1982; Lazear and Rosen, 1981). Research on innovation contests closely follows this tradition (Che and Gale, 2003; Fullerton and McAfee, 1999; Taylor, 1995). A central question in this research is whether free entry or restricted numbers of participants should yield better outcomes. The main, intuitive message from existing models goes as follows. In winner-take-all contests with only one participant, contestants will, of course, have little incentive to exert effort to improve their work because there are no parties against whom they will be evaluated. Thus, adding some minimum level of competition should stimulate rivalry (Harris and Vickers, 1987). However, adding competitors beyond the point at which real rivalry exists also makes individual contestants less likely to win, which risks diluting their incentives to invest effort in improving their performance. Importantly, these predictions apply to all competitors in a contest. These basic arguments have been shown to apply in both winner-take-all payoffs as well as in cases in

---

4 The other major issue in contest design addressed by the theoretical economics literature is how to set the “prices” (prizes, fees, and penalties) for contestants. Larger prizes tend to stimulate higher performance. Single prizes are argued to be effective for homogenous and risk-neutral individuals, multiple prizes are optimal when contestants have asymmetric ability and are risk averse (see Sisak, 2009 for an extensive review of the theoretical literature), and penalties are useful for motivating further effort by top tier contestants (see, for example, Moldovanu and Sela, 2001; Nalebuff and Stiglitz, 1983). A small empirical literature related to managerial compensation and sports competitions offers support for the contention in this theoretical literature that contestants respond positively to prize size, and for findings regarding the conditions under which a single prize is preferable to multiple prizes and vice versa (see, among others, Ehrenberg and Bognanno, 1990; Eriksson, 1999; Harbring and Irlenbusch, 2003).
which payoffs are more continuous, with multiple prizes and payoffs that increase more continuously with performance (e.g., Konrad, 2007, 2009; Moldovanu et al. 2007; Moldovanu and Sela, 2001). This has led a number of scholars to argue that restricting the number of contestants improves contest outcomes in general, and outcomes for innovation contests in particular (Che and Gale, 2003; Fullerton and McAfee, 1999; Nalebuff and Stiglitz, 1983; Taylor, 1995). The few recent empirical papers on contests in settling like sales compensations (Casas-Arce and Martinez-Jerez, 2009) and test-taking (Garcia and Tor, 2009) have provided some evidence of an effort-reducing impact of increased numbers of contestants. Our first prediction simply follows this baseline view in the established literature.

**Hypothesis 1** Individual competitors will reduce their effort and the distribution of performance outcomes will drop with added competitors (“Incentive Effect”).

2.2. Innovation Contests as a Search Process

Whereas seminal works in economics have treated different types of contests, from those concerning top managers to procurement to innovation, with the same incentive-based theoretical toolkit, more recent work, notably within the innovation and product development literature, has taken steps to address explicitly the special character of innovation problems. This body of work places particular emphasis on innovation as a process of problem-solving, or “search” for solutions, subject to false steps, experimentation, serendipity, and uncertainty (e.g., Loch et al., 2001; Sommer and Loch, 2004). Innovation is more than a matter of applying effort towards a defined objective and along a pre-defined path or trajectory. Progress might potentially be made along multiple paths or design trajectories across a wide and imperfectly understood technological frontier. Therefore, stimulating innovation should involve not just racing or high-powered incentives, but also broad searching.

Because the search view of innovation shifts the focus from how any one competitor performs to how the best competitor, the winner of the contest, does, our greater concern may thus be to design contests that increase the likelihood of at least one extreme outcome rather than high outcomes for a large cross-section of competitors (Dahan and Mendelson, 2001; Girotra et al., 2010; Terwiesch and Loch, 2004; Terwiesch and Ulrich, 2009). Formally, if innovation attempts are independent across competitors,

---

5 Further, the inherently public nature of contests, which often play out among individuals in socialized contexts, has led sociologists to conjecture that non-cash prizes such as status and social comparison might also play a role in contest performance and the analogous reduction of effort with high levels of competition (Bothner et al., 2007; and Garcia and Tor, 2009).

6 This notion of innovation is a longstanding idea in the innovation literature. See, for example, Abernathy and Rosenbloom (1969), Dosi (1982), Nelson and Winter (1982), and Simon and Newell (1962).
we may think of competitors as providing a set of random draws from some underlying distribution of possible quality of outcomes (Dahan and Mendelson, 2001). If adding competitors implies adding independent (distinct, parallel, alternative) solution approaches, then adding more competitors implies a greater chance of uncovering an extreme outcome.\(^7\) Substantively, this may simply translate into assaying a wider expanse of the frontier of technological possibilities and paths, and is consistent with the view that when it comes to innovation “safety would seem to lie in numbers and variety of attack” (Jewkes et al., 1959: p. 246).

Terwiesch and Xu (2008), in bringing this perspective alongside the formal modeling of incentives in the study of innovation contests, point out a tension between stochastic parallel path effects and incentives, particularly when the focus is on the winning performance in a contest. Although Terwiesch and Xu (2008) examine several institutional arrangements and contest design details, we emphasize their basic insight about the fundamental tradeoff between incentives and parallel path effects as the driver of our second hypothesis. Our empirical prediction reflects the intuition that, although individual incentives may change with added competitors, more competitors should always increase the value of the extreme outcomes in relation to the baseline distribution.

**Hypothesis 2** The winning or maximum performance increases in relation to the expected distribution of outcomes with added competitors (“Parallel Path Effect”).

It should also be mentioned that beyond simply demonstrating the distinct response of the maximum performance to added competitors, it is also crucial to gauge the magnitudes of the shifting of the maximum outcomes relative to the shifting in the distribution of expected outcomes. The magnitudes will tell us just how important it is to consider both sets of effects when designing a contest.

### 2.3 A Moderating Effect of Uncertainty

Uncertainty is a crucial and ever present feature of the process of developing novel solutions to problems, in other words, of innovating (Abernathy and Rosenbloom, 1969; Dosi, 1982; Nelson and Winter, 1982).

---

\(^7\) Consistent with the importance of searching in innovation, experimental evidence produced by Girotra et al. (2010) shows that when groups are organized in a way that leads to a higher number of ideas being generated, the best ideas are of higher quality. This implies that with larger numbers of “draws” there may be a greater likelihood of finding an extreme outcome. Note, however, that these are effectively multiple “draws” by a single organization rather than added draws occasioned by adding competitors.
Before further hypothesizing about how uncertainty might affect the relationship between adding numbers of competitors and outcomes in innovation contests, we pause to review notions of uncertainty in the innovation literature in order to better clarify what we mean by it in this paper.

Uncertainty may shape innovation and surrounding strategic interactions in a number of ways. Standard ways of introducing uncertainty in economic models of contests include the assumptions of heterogeneous abilities or valuations by contestants (and asymmetric information or ignorance regarding them; see, for example, Konrad, 2009; Terwiesch and Xu, 2008). This sort of uncertainty regarding talents and inclinations effectively translates into uncertainty regarding the likelihood of any one competitor winning a contest (Konrad and Kovenock, 2010; Terwiesch and Xu, 2008).

Other scholars have suggested that uncertainty and its effects can be determined by focusing on the nature of a particular problem and of the knowledge required to solve it (Sommer et al., 2009). A long-standing interpretation of innovation as deriving from the recombination of different sets of knowledge and ideas (Fleming, 2001; Katila, 2002; Nelson and Winter, 1982; Schumpeter, 1943; Taylor and Greve, 2006; Weitzman, 1998) provides the conceptual building block for the notion of “recombinant uncertainty.” The greater the set of knowledge components or domains involved in addressing an innovation problem, the higher the expected uncertainty or variability of the outcomes (Fleming, 2001; Taylor and Greve, 2006). Kavadias and Sommer (2009) introduce explicitly, in a model of problem solving performance, the differences in the underlying problem being solved. Akin to the earlier reasoning, cross-functional problems, in particular, are defined as those requiring knowledge from different areas. Using simulation modeling, Kavadias and Sommer (2009) show that these problems are more likely to be solved when the diversity of the solvers is most fully exploited. Interestingly, even if the competitors solving an uncertain problem were identical (in contrast to the discussion in the previous paragraph), the effect of encountering an uncertain problem would similarly translate into uncertainty regarding the likelihood of any one competitor winning the contest.

This view of uncertainty has its roots in a notion of technological uncertainty akin to another view of uncertain “searching” along a frontier of different paths or trajectories of paradigms within which to search for and improve upon existing solutions (Dosi et al., 1988; Sahal, 1983). For challenging problems, there may be multiple, fundamental approaches with varying levels of feasibility and ultimate potential. When one approach is exhausted, we may need to seek a new pathway or trajectory or design approach to continue to make progress.8 This sort of uncertainty may go deeper than the earlier sorts; not

---

8 Much of this work does not even interpret uncertainty in statistical terms; in the Knightian sense (1921), uncertainty means that a distribution of outcomes cannot be determined. Other works consider uncertainty in
only may competitors’ ability to solve a problem differ (and the competitors not know it), and the problem solutions have inherently high variability, but it may not even be clear what sort of basic approach should be taken to the problem, how many possible approaches there are, and the return to pursuing any given approach. For purposes of considering the problem of adding competitors in an innovation contest, this sort of uncertainty similarly translates into uncertainty regarding the likelihood of any one competitor winning a contest, albeit at a perhaps more profound level.

Uncertainty should thus be understood to be an important ingredient as relates to innovation outcomes, one with many forms and degrees. For purposes of this article, we simply wish to emphasize that many forms of uncertainty in innovation often translate into uncertainty regarding precisely which competitor will achieve the best/extreme outcome (and how effective the solution of any one competitor will turn out to be). This effect is the basis for the “parallel path” effect, and directly implies that added uncertainty should simply amplify the parallel path effect: greater uncertainty increases how much adding competitors affects the maximum outcome relative to the expected distribution. Thus, the third basic hypothesis used to guide our empirical exploration is as follows.

**Hypothesis 3** The moderating effect of added uncertainty is positive on the parallel path effect.

It is important to note that the effect of greater uncertainty on incentives (and possible interactions with parallel paths) is a far subtler question, without a clear general prediction. (We also explore this relationship in the empirical tests.) A number of factors are implicated here including the shape of the knowledge distribution, number of competitors, skill levels of competitors, degree and scope of uncertainty, and so forth.\(^9\)\(^10\) For a preliminary and suggestive intuition, consider that if the eventual winner is from near the top of the true knowledge distribution, adding uncertainty might foster belief in a more “level playing field” than actually exists. On one hand, this might lead eventual winners to

---

\(^9\) For example, if we model uncertainty through a Gumbel distribution (see, for example, Terwiesch and Xu (2008), it can be shown that the (negative) relationship between an individual’s choice of level of effort and the number of competitors is affected by the degree of uncertainty, as expressed by the scale parameter of the distribution, in a non-monotonic way, depending on number of competitors, the particular skill level or “draw” of a given competitor, and the scale parameter itself.

\(^10\) See also List et al. (2010) for a study of the effect of the “slope” of the density of the random component on the competition-outcome relationship in contests.
underestimate the probability of winning and shade their level of effort downward. On the other hand, creating a perception of closer rivalry could stimulate extra effort in leaders who might otherwise “rest on their laurels.” What can be argued, however, is simply that there should likely be some moderating effect on the incentive effects. Insofar as the moderating effect on the incentive effect could plausibly be negative, it is unclear whether added uncertainty, in the event competitors are added, should necessarily increase the net benefits to the extreme value.

3. Empirical Setting: TopCoder Software Contests

An ideal empirical setting in which to test the preceding hypotheses about innovation contests should satisfy a number of non-trivial requirements. First, it should contain precise measures of innovation outcomes (“problem solving success”). Second, it should allow for random variation in level of competition as well as in uncertainty. Thus, otherwise identical competitors should be assigned to different degrees of competition and problem uncertainty. Third, measures of competitive pressure as well as metrics that distinguish problems in terms of uncertainty are needed to translate conceptual constructs and hypotheses into empirical analyses. Finally, the availability of data on the whole distribution of outcomes rather than, for example, only on the best performance would enable us to distinguish more clearly between the effect of competitive pressure and uncertainty on the stochastic and effort components of the innovation outcomes, and thereby more precisely assess the effects of the incentives and parallel path views in different problem environments.

The data we analyze were provided by TopCoder. Established in 2001, TopCoder creates outsourced software solutions for IT-intensive organizations by encouraging independent programmers from around the world to compete in a regular stream of software development contests. TopCoder’s value proposition to its clients is that it can harness the value of large numbers of programmers working in parallel and let the competition determine the best solutions, without risking either a wrong hire or an incorrect solution. Over the years, TopCoder clients, including such firms as AOL, Best Buy, Eli Lilly, ESPN, GEICO, and Lending Tree, have benefited by having their internal software programming challenges resolved, and TopCoder members have had the opportunity to win cash prizes, obtain third-party assessments of their skills, and signal their talent in a global competition. Over the past ten years, TopCoder has run more than 7,500 competition events. In 2009, more than 11,122 programmers from around the world competed in 1,425 software development contests for 47 clients. Interviews conducted with TopCoder executives and community members during the course of the study to understand the
dynamics of the contest platform and various motivations that drive participation and performance inform our contextual explication.

In general, TopCoder works with client firms to identify software module requirements that it converts into contests for its community of programmers. The contests typically run for several weeks and target specific programming tasks like conceptualization, specification, architecture, component design, component development, assembly, and testing. Each submission is subjected to an evaluation by a peer review panel of three expert members or assessed by automatic test suites that check for accuracy and computation speed, or both. Winners are awarded pre-announced cash awards (range: $450-$1,300 per contest) for their contributions, and the performance of all participants is converted into a continually updated rating in each contest category. Of the more than 250,000 programmers from around the world who have signed up as members, well over 40,000 have obtained ratings by participating in contests.

Essential to TopCoder’s success is a growing stable of programmers willing to participate in software development contests, and the ability to convince client firms that its community of programmers has the skills and ability to deliver results and create high quality software solutions. Member recruitment is done through active outreach to college campuses worldwide and through joint sponsorship of programming competitions and events with high-profile technology firms. Members are encouraged to participate and demonstrate their skills through weekly to biweekly algorithm programming contests in which participants compete against each other to solve three software development problems in 75 minutes. The solutions to these problems are automatically scored via a large test suite custom tailored to each problem. Participation and performance data from the algorithm programming competitions provide the test bed for analyzing our hypotheses.

3.1 “Algorithm” Problems

TopCoder relies on dedicated internal staff and outside consultants to design the software challenges used in the algorithm contests. A central concern for designers is to create problems that members will find both interesting and demanding, and at the same time allow TopCoder to discern between mediocre, average, and great programmers. Explained Mike Lydon, chief technology officer for TopCoder and the principle designer of the algorithm contests framework:

Algorithm problems test participants’ ability to take an abstract problem statement and convert it into a working software program that will meet the requirements of the challenge. This requires creativity in developing solutions that rely on a broad knowledge of various algorithmic approaches, and the application of mathematical thinking in a severely time-limited context. While these problem are synthetic, the skills we assess and reward are directly applicable to diverse and demanding domains like computational biology and genomics, biomedicine, aerospace engineering, image processing, financial
fraud detection, graphical rendering, and text mining, amongst many others. These problems are micro versions of the Netflix Prize problem and performance in these challenges generally correlates well with our other client contests. Many of our top members have obtained employment in a wide variety of demanding software development roles based on their performance in our algorithm contests.

Our interviews with TopCoder problem designers revealed that they have to create challenges that have well-defined outcomes so that automated test suites can be used to assess performance (e.g., “find the most popular person in a social network of differing ethnicities in the least amount of computation time”), but the potential approaches to solve the problem can be varied.\(^\text{11}\) Thus, the preceding problem requires knowledge of both graph theory and string parsing to develop an effective solution. TopCoder problem designers, in their attempt to create challenges that test both ability and knowledge of a variety of algorithmic approaches, explicitly consider a variety of relevant knowledge domains that could be designed into a problem. Added Lydon: “Our problem designers have to constantly develop appealing problems that will pose tough intellectual challenges to our talented programmer community. We make our problems interesting by having them situated in multiple domains in the various fields of algorithms and mathematics.” Competing solvers simply access the problem statement, and do not know how many or which knowledge domains the designer has designated for a particular problem.

Once a problem has been developed, TopCoder designers create an elaborate automated test-suite, often consisting of hundreds of test cases per problem, to check for algorithmic accuracy. The test-suites consist of all obvious and non-obvious edge conditions a programmer must meet to create the right solution to a problem. Once the test suite is created, the problem designer and an experienced quality assurance engineer simulate the test conditions by trying to solve the problems themselves within the 75-minute time constraint. Based on their experience with the problems, a final points value is assigned to each problem. The appendix to this paper provides details on one contest problem statement, the participation patterns, and a solution synopsis by one of the members.

3.2 Algorithm Competition Format

Algorithm contests are held at different times and on different days of the week to accommodate TopCoder’s global membership. Contest dates and times are advertised well in advance to all registered members of TopCoder through personalized e-mails and on the company’s Web site. Competitions occur in two broad divisions, I and II, based on prior skill ratings in previous algorithm contests. Division I consists of participants who rank above a pre-determined rating score, the elite programmers in the

\(^{11}\) Terwiesch and Xu (2008) provide a typology of innovation projects that could be solved via contests on the dimensions of technical and market uncertainty. The algorithm problems in our context exhibit varying degrees of technical uncertainty, however, solvers are given only one attempt to solve them.
TopCoder stable, Division II of newcomers (i.e., those who do not yet have skill ratings) and those who rank below the Division I threshold score.

On the day of a contest, members are given a three-hour window in which to register their intent to compete. Five minutes before the start of the contest, registration is closed and the typically hundreds of entrants in any given contest are divided into groups, termed virtual “rooms,” of not more than 20 competitors. TopCoder chose the virtual room format to accommodate large numbers of competitors in the contest, typically several hundred, without making it so intimidating and large that competitors would be discouraged. Another reason for creating virtual rooms of 20 was to allocate prize money across the wider pool of participants. Each virtual room gets the same three problems in the division; competitors in different rooms thus solve the same problems, but direct competition largely takes place within a single room. This is because rank within an individual room determines cash prizes, if any, as well as public recognition for winning. As prizes are divided among different subsets of direct competitors by virtual room, there might be on the order of one to two dozen winners among several hundred entrants. Observed Lydon: “In the Algorithm competition our members care deeply about the competition that is occurring in the room that they have been assigned to. Winning in the room is as important as winning in the overall competition because the primary competitors are in your face.” Figure 1 illustrates the algorithm contest arrangement over multiple time periods.

In the early years, from 2001 to 2003, TopCoder experimented with a range of assignment procedures to the virtual rooms before finally settling on within-division random allocation to a room. Recalled Lydon:

As prize money is allocated based on room performance, there was significant concern amongst the participants about the potential biases in the room assignment procedures. We initially started with an “Ironman” style assignment procedure where participants were rank-ordered by rating and then sequentially placed in a room to capacity. Our membership reacted quite negatively to this approach, and so we adjusted by creating a pseudo-random allocation procedure. We eventually chose a simple random assignment procedure to allocate competitors to virtual rooms within divisions.

As of 2004, allocation to rooms was done on a fully random basis within each division.

Contests consist of two distinct phases, 75 minutes of programming, followed by 15 minutes of solution testing. In the programming phase, participants write and submit code for each of the three problems. Each problem is assigned a set amount of points visible at the start of the contest, typical values being 250, 500, and 1,000. As soon as a participant opens the problem, that is, gets the full problem statement, the available points for a successful submission start to decline based on the amount of time
between problem opening and submission of code. Hence, the faster the programmer finishes the submission, the greater the number of points available, subject to automated testing at the end. If participants open all three problems at the same time, all three will have the total number of points declining.

Competitors within individual virtual rooms are also provided with rich information about each other and the unfolding of the competition in the room. Included in a “heads-up” display in which coders complete their code is the full list of the competitors in the room (those that have logged-in post registration period), color-coded to facilitate quick assessment of their skill ratings. Clicking on any name reveals further information about, and a detailed history of the performance of, that competitor (Figure 2 presents what competitors see). As there are 20 or fewer competitors in a room, this information is easily navigable. The display also reveals who has submitted solutions to enable the progression of the contest to be observed in real-time. The ability to observe the submission of solutions by competitors gives participants an idea if they are ahead or behind in the competition.

[Figure 2 about here]

Final scores for each participant are determined in the testing phase by automatically compiling the software code for each problem and subjecting it to a barrage of from hundreds to thousands of automated test cases to determine the accuracy of the solution over a range of potential conditions and edge cases. Within each virtual room, participants have the right to examine any other competitor’s code and submit a test case they believe would cause their competitor to fail. If the challenge test case is successful, the challenger receives 50 additional points and the challenged participant loses all points for that problem. The test case is then made part of the full, automated test-suite for all participants. Challengers risk losing 25 points if unsuccessful in disqualifying their opponents. Performance over all the test cases is summed and the time taken to submit the answer converted into an objective final public score and ranking of each participant’s algorithm code-writing skills. Post-testing, the problem performance scores and ranking of each participant within the room and in the competition are publicly released.

3.3 Motivations to Participate in Algorithm Contests

A central practical and theoretical concern in innovation contests is the motivation. A chief lever available to motivate participation in the contests is the structure and form of prizes (Cason et al. 2010; Konrad 2007; Konrad and Kovenock, 2010; Moldovanu and Sela 2001; Sisak 2009). As noted

---

12 By virtue of the contests lasting a fixed 75 minutes, the effort exerted is the level of cognitive effort rather than, say, a discretionary level of working hours or capital investment.
earlier (in Section 2.1), the literature has examined both “winner-take-all” and more continuous prize structures. The TopCoder context in general, and the algorithm contest in particular, provide discrete payoffs for winners as well as more continuous payoffs across competitors. Winning cash is the most conspicuous motivation to participate in TopCoder. Between 2001 and 2009, TopCoder disbursed more than $1M in cash prizes for the algorithm contests alone. More continuous sources of payoffs (whereby higher ranking outcomes generate higher payoffs) relate to a wide range of other motivators. Here, public rankings and ratings resulting from contest performance are crucial. Placing high in an individual contest or achieving a high rating through sustained success are non-pecuniary sources of satisfaction that can also directly translate into career opportunities. High-profile firms like Intel, Facebook, Google, and Microsoft, for example, both sponsor the algorithm contests and encourage some prospective employees to get a TopCoder rating in order to be considered for programming positions. Although participation in one contest may not directly convert into an interview or job offer, sustained superior performance in these contests clearly signals ability and quality to prospective employers. High rankings and ratings are coveted, and members work hard to maintain and improve them over time. To many participants, the ratings are also a sort of status symbol. According to Justin Gaspar from the United States: “If you have a red rating (the highest possible), people look up to you.” Members have their own profile pages that track performance in every contest and provide a ratings measure and distribution on TopCoder (Figure 2). Dips and rises in performance and rankings after each contest are publicly discussed on the TopCoder community message boards, with individuals with both inflated and bruised egos participating. Our interviews revealed that members, especially those in the higher performing brackets, took it very personally if they did not come out on top in a competition. Remarked Mike Paweska from Canada: “To be successful at TopCoder you must ask yourself, ‘Are you a competitor?’ You need to be able to thrive on competition; you can’t be scared of it.” This point also surfaces what appears to be an intrinsic desire to compete in many members. Lydon notes: “Regardless of cash prizes, winning in the rooms and in the overall competition is everything to our top members.” Thus, those who do not rank first still may receive some “prize” related to their relative position. There is, however, a major discontinuity in the reputation effect in classifying first as opposed to any other position.

These various motivators beyond just cash incentives are consistent with a number of papers that have remarked the importance of sociological and behavioral motivators of various kinds in contests (Altmann et al., 2007; Konrad, 2009; Moldovanu et al. 2007). These more continuous sources of performance-based payoffs appear to be rather important in at least this context. TopCoder states that it sees little difference in performance whether a cash prize is offered or not, particularly now that the contest platform has grown and is internationally known by software developers. (Only roughly a third of algorithm contests have cash prizes).
4. Data, Measurement and Estimation Strategy

TopCoder executives granted us access to the full database records of 350 (roughly weekly) “algorithm contests” between 2001 and 2007. Our analysis focuses on the elite Division I, in which ratings were more reliable and individual solvers tended to compete more regularly than in Division II. We also focused on post 2001 contests, after the pseudo-random assignment procedure began. The sample covers 645 problems. Our empirical analysis focuses on the variation across rooms, the distinct groups of direct competitors that compete on each problem. There are 9,661 room-problem contests observations. We first describe our outcome variables, and then the key explanatory variables. Descriptions of all the variables used in our analysis are provided in Table 2, descriptive statistics and correlations in Table 3.

[Tables 2 & 3 about here]

4.1. Measuring Problem-Solving Performance

We measure innovation performance outcomes in terms of the final score assigned by TopCoder’s automated evaluation system to a given solution to a given problem, which we denote as Score. The Score per problem is based on the initial pre-set points allocation, which declines steadily once a competitor opens the problem during the contest up to the point of submission to the evaluation test suite. The faster a competitor codes, the higher the score, contingent on passing all challenges and system tests. Particularly relevant, given our research questions, are the average scores (Average Score) and maximum scores (Maximum Score) attained in a given room for a given problem. We supplement the Score measure with two other measures to assure its robustness. Recall that the final score is the result not just of an automated set of performance tests and barrage of test scenarios, that it is further adjusted if competitors find weaknesses in the solutions. To assure that the final score is, in fact, a good representation of the true merits of a solution rather than just representative of, say, strategic effects or a tit-for-tat challenge, we supplement the final score with the initial submission score and a dichotomous variable that distinguishes submissions considered incorrect (value of zero) and not incorrect (value of 1).

4.2. Key Explanatory Variables: Number of Competitors and Level of Uncertainty

The main explanatory variable is No. of Competitors, that is, the number of direct competitors facing each other in the same virtual room. For the regularly scheduled algorithm competitions, this number ranges between 10 and 20 (with the bulk of our sample between 15-20). However, for robustness checks (see section 4.4) we also include out-of-sample data from algorithm contests used in various TopCoder
elimination contests, both virtual and face-to-face, for qualification and participation in the annual TopCoder Open grand finale competition. This extends the range of our main variable from 3 to 25.

Our investigation is also interested in how the relationship between innovation performance and number of competitors might be affected by uncertainty (Section 2.3). As noted by Sommer et al. (2009), established empirical measures of (unforeseeable) uncertainty not being readily available, researchers have to rely on the empirical context for their derivation (p. 125).\(^\text{13}\) This measure thus requires special attention and care to properly motivate and interpret. In operationalizing and measuring uncertainty, we rely heavily on the particulars of our empirical context. Discussions and interviews with TopCoder managers during the roughly 18 months of this analysis led us to focus on a particular source of uncertainty that we and they believed to be most salient: the number of problem domains on which a given solution draws and associated uncertainty in the “approach” to solving a problem.

As relates to algorithm contests, TopCoder managers have long been sensitive to the need to make the contest problems continually interesting and challenging in order to maintain a high degree of participation. Apart from randomly mixing who appears in a given room of competitors, TopCoder’s problem designers also deliberately tune and adjust the degree of uncertainty in competition outcomes through the design of the problems.

The deliberate attention to problem design has led TopCoder to keep careful records of the nature of problems according to 16 canonical problem domains (Table 1). Roughly half of the problems included in competitions are single-domain problems, that is, they are classified as belonging to just one of these 16 categories. In conforming to a given problem type, these single-domain problems have canonical solution approaches. Although they remain non-trivial, a dominant approach or template can be used to approach the solution to the problem. Anecdotal accounts from competitors strongly corroborate this contention of TopCoder managers and problem designers. The competitors suggest that approaches to these problems can often be somewhat standardized, and even possibly “routinized,” at least to some extent. Observed Lydon: “Some competitors often modify a pre-built library of solutions to answer the single domain problems.”

TopCoder problem designers and executives suggested that the somewhat standardized approaches used for single-domain problems were far less likely in instances in which problems drew from multiple domains. Multiple domain problems often do not just “add” two sorts of problems together such that rote solutions might still be viable. These problems often involved what were described by

\(^{13}\) Sommer et al. (2009), for example, relied on survey-based self-reports by managers on a Likert scale to operationalize and quantify (unforeseeable) uncertainty.
Lydon as “tricks,” and the solutions were “much more about the sort of approach taken” and “required greater creativity.” It is in combining canonical problems to produce multi-domain problems that the problem designers attempt to inject greater uncertainty into performance outcomes.

A post-contest synopsis by one participant provides insight into the uncertainty faced by the competitors.\(^\text{14}\)

Picture yourself as an average Division 1 coder. You have just quickly finishedy [sic] quickly the easy problem and think that there's enough time left to take the medium slow. The 50 extra points contribute to this impression. After reading the problem statement, you write down some numbers and mathematical expressions, maybe think about a dynamicc [sic] programming approach, but nothing convinces you. After 15 minutes of doing this, you are mad at yourself and take a look at the division summary: Nobody has submitted! Not even one of the many coders that have several hundred rating points more than you…. Solving this problem required imagination and either faith or a good proof. The strange thing in this case is that almost everybody solved it differently.

These considerations of TopCoder problem designers and competitors are echoed and supported by research on “recombinant” problem-solving and innovation, according to which the presence of multiple knowledge domains should produce higher uncertainty and risk in the innovation process (Fleming, 2001; Kavadias and Sommer, 2009; Taylor and Greve, 2006). We thus use the number of problem domains from which a given problem design draws (No. Domains) as a proxy that should relate systematically to uncertainty with respect to both problem approach and who will win.

### 4.3. Estimation Approach and Control Variables

The objective of our empirical analysis is to estimate how exogenous variation in No. Competitors affects the distribution of performance outcomes in the room. In regressing measures of the distribution of Score on No. Competitors, our greatest concern is that coefficient estimates might be biased by spurious correlations associated with a long list of possible determinants of performance if we fail to control for those determinants.\(^\text{15}\) In principle, any number of factors might influence performance outcomes: whether a particular round had money prizes, the size of the prize(s), whether a given round received corporate sponsorship, how well-known TopCoder was at that time, how a given round corresponded to the calendar year or hiring cycles, and so forth. Further, the nature of the problem, maximum, theoretically attainable number of points, and so on should also have a direct influence on performance. Thus, to assure

---


\(^{15}\) Given that these are one-shot competitions and participants do not have details about the problems or direct competitors before a competition starts, we can reasonably rule out reverse causality.
that estimates are not biased by spurious correlations, we need to control for a wide range of imaginable “institutional” details or problem characteristics and changes over time.

Our approach is to simplify this set of issues by controlling outright for all differences across rounds, and for time and differences across problems by adding individual problem fixed effects in the regressions. In effect, we simply exploit the differences across rooms for a given problem to estimate the relationships of interest. This radically simplifies the estimation problem.

Consequently, the task of the control variables is simply to account for differences across rooms. The rooms themselves being identical, what varies across rooms is who is in them. Being interested in estimating the effect of the number of competitors in a room per se, what remains is to control for the composition of the individuals in a room. Fortunately, TopCoder provides an excellent measure of the skills of all participants based on their historical performance. Every competitor is evaluated and rated after each contest using the long-established “Elo” system used to evaluate, rate, and rank chess grandmasters (Van Der Mass and Wagenmakers, 2005). This system assesses skills based on the performance of a competitor relative to everyone else working on the same problem. Figure 3 shows the rating and relevant placement of a participant using the competitor’s pre-contest rating (Skill Rating) as the measure of ability. We use these measures of individual skill to construct multiple statistics that capture the skills distribution in each room based on different moments of the distribution (i.e., average, variance, skewness, and maximum).

Therefore, the basic intuition for the estimation approach is quite simple: we estimate how varying the number of competitors across rooms for a given problem affects the distribution of outcomes, controlling for differences in the distribution of skills across rooms. Given the importance of between-room differences, we present the distribution of these key variables across rooms. Figure 4 shows the distribution of Score. Whereas Score may vary appreciably from problem to problem, controlling for problem fixed effects yields a much smoother, single-peaked distribution. Figure 5 graphs the distribution of number of competitors before and after netting out problem fixed effects. As seen, there is little difference, as the number of competitors in a given competition is not strongly correlated with particular problems. Figure 6 plots the distribution of Skill Rating with and without problem fixed effects removed. As can be seen, the distribution does not change dramatically with and without problem fixed effects. Importantly, we also show this distribution for cases of different numbers competitors, as we are particularly sensitive to the possibility that there may be qualitatively different sorts of competitors in rooms with different numbers of competitors, and this could drive main regression results. Crucially, these differences are slight. Therefore, if there are systematic differences in the distribution of skills in
rooms with, say, 16 versus 17 competitors, then the list of statistics that captures the distribution of Skill Rating (i.e., the average, variance, skewness, and maximum, as noted above) needs account only for incremental differences in a similar distribution of skills rather than for qualitatively different distributions.

[Figures 3, 4, 5 and 6 about here]

As can be seen in Figure 5, even controlling for problem fixed effects there remains considerable variation in No. Competitors across rooms (a range, roughly, of 5, and robustness tests allow us to explore beyond this range). That said, we still need to assure that there exists some exogenous source of variation in the number of competitors across rooms (otherwise, our room-level controls will likely soak up all variation and there will be little exogenous variation with which to estimate coefficients). Two main sources of exogenous variation give rise to differences in the number of competitors in individual rooms. The first relates to “imperfect” room assignment. Although the goal is to fill each virtual room to 20 competitors during the competition, this rarely occurs in practice. The bulk of contests have between 15 and 20 competitors in each room. By definition, the average number of competitors in a room is given by the total number of participants in an event/round, divided by the number of rooms created for the event. A first and simplest reason for variation below 20 competitors is the constraint of not having participants arriving in multiples of 20. For example, a contest with 81 competitors would have five rooms, with an average of 81/5 competitors per room. The simple integer-counting problem, plus some noise created by the room assignment algorithm, would typically generate several rooms ranging from perhaps 15 to 18 participants.

At least as important, however, is the role of “no-shows,” individuals who have signed up and been assigned to a room, but fail to ever “check-in” and begin to actively participate in the contest. Importantly, no-shows do not know their competition or the nature of the problem before deciding not to show up. Nor do they activate their presence on the heads-up display in their rooms; they are effectively absent and invisible. We cannot directly observe the decision to not show up, but there are strong indirect indications. First, TopCoder managers and participants see this as a “fact of life” on the platform. We also speculated that we should see more no-shows on weekdays if it is simply harder to plan and predict one’s availability on weekdays. Consistent with this view, we found that average numbers of participants in rooms was lower on weekdays (whether controlling for total participation or not).

Specifics of the regression models are discussed further in the analysis below. Broadly, we estimate the relationship between performance outcomes and number of competitors using a series of ordinary least squares (OLS) and with fixed effects as well as quantile regressions using a weighted absolute deviation algorithm (Koenker and Basset, 1978; Koenker and Hallock, 2001).
4.4. Additional Robustness Issues
To assure that there are no subtle interactions working across the sample necessitates several additional tests. First, we extend the range of our main variable of interest, No. Competitors, to include competition rooms that ranged in size from 3 to 25 using data from algorithm competitions held as part of the qualifications and events for the annual TopCoder Open convention. Second, we would like to assure that there are no qualitative differences in the modeled relationship across different rounds or events (based on unobserved factors such as whether an effect received sponsorship). Although we do not observe these details, we do observe total attendance at a given event (the sum of competitors across rooms). Regressing the model on subsamples of widely attended events versus sparsely attended events should thus provide some indication of the robustness of the results across different sorts of events.

Finally, we might further be concerned that events for which there are no cash prizes might plausibly produce qualitatively different behaviors and responses to competition. Although our interviews with TopCoder executives and direct observation of these competitions suggest that this is not the case, we are able to directly test this possibility, as we can identify the events for which there were no cash prizes. Running regressions on this subsample yields similar results.

5. Findings
Our findings are reported in three subsections. First we assess the baseline model and its robustness in its simplest form. We then model how the distribution of problem-solving outcomes and maximum outcome are affected by varying the numbers of competitors. Finally, we assess the moderating effect of varying levels of uncertainty (implied by different sorts of problems) on these relationships.

5.1. The Baseline Model
We begin by assessing the baseline model on a most basic measure of the distribution of outcomes, Average Score. To the extent that it is properly specified, the model of Average Score quantifies how much the expected distribution of outcomes in a room shifts upward or downward, on average, in response to varying the number of competitors (No. Competitors). This is the average incentive effect. Our goal in the following analysis is to rigorously demonstrate the workings and robustness of the estimation approach before moving on to a more nuanced study of the changes in the broader distribution of outcomes. We specifically conclude here that the Average Score in a room drops by about 5 points for every added competitor in a room, controlling for the level and distribution of skills in the room. Results are presented in Table 4.
Model 4-1 regresses *Average Score* on *No. Competitors*, with problem fixed effects.\(^{16}\) Recall that problem fixed effects control for differences across not only individual problems, but also different rounds/events and time. The coefficient estimate on *No. Competitors* is negative and highly significant. To assure that differences across rooms are not somehow biasing the estimated coefficient, what remains is to control for compositional differences across rooms. Model 4-2 adds the simplest measure of differences in skills across rooms, *Average Skill Rating*. This changes the magnitude of the coefficient on *No. Competitors*, but the coefficient remains negative and, at -5.74 (\(p = 1\%\)), highly significantly different from zero. As an assessment of the effectiveness of controls for skill across rooms, we examine a couple of alternative specifications. First, we allow for the possibility that the effect of *Average Skill* may enter non-linearly. As reported in model 4-3, allowing for non-linear relationships with average skills across different rooms (the reported model implements non-linear controls with 20 dummies for levels of average skill at each 5-percentile) also does not change the results from a roughly -5 effect (-5.01, \(p = 1\%\)). As an additional test, we include a variety of other measures (rather than just the mean) of the distribution of *Skill Rating* in a room. In model 4-4 we add, in addition to the average, the variance, skewness, and maximum of *Skill Rating*.\(^{17}\) Adding this large set of statistics to describe the full distribution still yields a coefficient of roughly -5 (-4.63, \(p = 1\%\)). The lack of sensitivity of the coefficient on *No. Competitors* to controls for higher order moments is consistent with the clear similarity of distributions across rooms seen in Figure 6. Nonetheless, we continue to include these controls, given that their statistical significance suggests that they explain a measure of variation and therefore improve the precision of the estimates.

To provide greater assurance of the meaningfulness of the *Score* measure (which represents the final score conferred on a given solution), we performed several additional tests. Similar patterns were observed when we replaced our main final score measure with either initial submission score, as in model

\(^{16}\) The \(F\)-test for overall model fit is significant at \(p = 1\%\) for all models. Standard error estimates are robust to autocorrelation and heteroskedasticity.

\(^{17}\) Transforming the skills rating to the log of the skills rating and re-running these models yields similar estimates of the coefficient on *No. Competitors*. We also assessed the robustness of results with a completely different approach in which we estimated how individual competitors’ performance varied from round-to-round, controlling for individual competitor fixed effects, a series of covariates for round and problem covariates (using event time of day as an instrumental variable). This approach produced almost identical point estimates of the average effect of added competitors, but with much lower statistical significance.
4-5, or with an indicator for incorrect versus not-incorrect solutions, as in model 4-6.\textsuperscript{18}

We ran additional robustness checks to assess the results on sub-samples and out-of-sample data. Results are reported in Table 5. We began with an out-of-sample test to assess whether the result of a roughly -5 slope was valid beyond the particular range of No. Competitors in our sample. Using data from ad hoc contests provided considerable additional variation in that variable. Similar results were obtained whether we regressed the model on a particular low or high range, or the entire range, of No. Competitors. Given that the ad hoc contest data and sample data were largely complementary in their coverage of different levels of No. Competitors, we felt it most meaningful to report regression results for the pooled data, as in model 5-1. To more explicitly show the regular negative linear relationship of roughly -5, we plot the results of a model that included a quadratic term to allow for any convexity or concavity over the pooled data. As can be seen in Figure 7, the relationship is nearly linear and regularly negative.

To assure that the sample estimates do not confound or obscure distinct cases in which there were no monetary prizes for a contest, we also ran the regressions on just the competitions without prizes and found no differences in the estimates, as reported in model 5-2. Coefficient estimates are statistically indistinguishable between contests without prizes and the entire sample. To control for additional potential unobserved round characteristics influencing the relationships of interest, we separated the analysis between subsamples of events that had high total numbers of participants and those that had low total numbers of participants (relative to the median participation in events/rounds for a given year). The results in both cases appear similar to those for the entire sample, as reported in models 5-3 and 5-4.

[Figure 7 about here]

Finally, to account for changes in the room allocation process we looked for possible differences in patterns over time by comparing results in pre- and post-2004 subsamples. As reported in models 5-5 and 5-6, the results generally follow those for the full sample, with negative coefficients on No. Competitors in either case, and therefore broadly confirm a negative or downward shift in problem-solving performance with added competitors.\textsuperscript{19}

\textsuperscript{18} Modeling the dichotomous variable in a linear or binary framework (such as Logit or Probit) does not affect the results.

\textsuperscript{19} The magnitudes of these estimates are somewhat different, however. We examined these patterns more closely by studying data for individual years separately. We found that the divergence in magnitudes was created by differences in just two years. In 2007, the coefficient on No. Competitors is unusually low; in 2002, it is unusually high. In both years, however, the estimates are themselves not statistically significant. Estimates for all other years are of similar magnitude and statistically significant. Given that we have no a priori reason to reject the data for
5.2. Number of Competitors, Distribution of Outcomes and Extreme Value Performance

Having observed a general negative shift in performance outcomes with added competitors (about -5, on average, with each added competitor), we now examine the effects on the overall distribution of outcomes. We demonstrate in this section that there is a marked difference between how the expected distribution of outcomes the maximum outcome/score are affected by adding to the number of competitors. We also find the negative effect on expected outcomes to be consistent across the entire distribution of outcomes (there are no cases of increasing performance with added competitors). The analysis in this section, therefore, most closely relates to Hypotheses 1 and 2, which emphasize, respectively, the negative incentive effect of added competitors and how, in contrast, the maximum score should benefit from “greater numbers of draws.”

To document how the wider distribution of outcomes, beyond just the average, shifted in response to added competitors, we assessed how different quantiles of scores were affected by adding competitors while controlling for the underlying distribution of skills. In Table 6, quantile regression results for the 25th, 50th (median), 75th, and 90th percentiles of the distribution of outcomes are reported, as in models 6-1 through to 6-4. The results of different quantiles to varying numbers of competitors are consistent with the earlier findings of an overall downward shift, the coefficient on No. Competitors being negative in each case, showing that the downward shift in the average reflects a more general downward shift across the entire distribution. Moreover, it appears that the upper tail of performance, in particular, shifts downward more than the rest of the distribution. This can be seen from the larger (negative) magnitude of coefficients on No. Competitors for higher quantiles, as in models 6-3 and 6-4. This pattern is plausibly consistent with top competitors making a more strategic response to added competition. It might also simply reflect that the lower quantile is bounded from below by the minimum, zero score.

Having demonstrated that the negative response to added competitors is general across the entire distribution of expected outcomes, and is particularly negative for the highest quantiles, we examine how adding competitors affects the maximum score. Hypothesis 2 predicts that the maximum score should

---

2002 and 2007, we continue to include them in the reported results. This makes no difference in the results we report.

20 An alternative way to demonstrate similar patterns is to simply regress both the mean and variance of room scores on No. Competitors. This approach produces results in line with those of the quantile regressions.
shift upward in relation to the expected distribution. This is simply because the maximum score should reflect not just shifting incentives and effort, but also the stochastic parallel path effect, which can create additional upside for the maximum score attained in the distribution. In this context, in which the distribution shifts downward, this means an at least less negative if not more positive response to No. Competitors than is seen in the expected distribution of outcomes.

Model 6-5 essentially repeats the earlier preferred model 4-4 for average score, but replaces the dependent variable with the maximum score attained in each room. Consistent with our hypothesis, the response of the maximum score is substantially less negative relative to the distribution of expected outcomes. For example, whereas the upper tail of the expected distribution decreases by almost -9.39 with added direct competitors in a room (model 6-4), and the average is -4.63, as in model 4-4, the maximum score decreases by only -.88 (model 6-5), an upward shift relative to the distribution of expected outcomes.

Figure 8 presents graphically quantile (in 5% increments) and maximum responses along with 90% confidence intervals to clarify that the results in Table 6 are part of a relatively smooth shifting and reshaping of the distribution of expected outcomes with added competitors, conditional on the skills distribution. The sizeable differences between these responses to No. Competitors can be seen graphically in Figure 8. We thus find support for Hypotheses 1 and 2.

[Figure 8 about here]

5.3. The Effect of Uncertainty

We now examine how varying levels of uncertainty affect the earlier relationships, a question most closely related to Hypothesis 3. Greater uncertainty should increase the parallel path effect, but the effect on competitors’ incentives is theoretically ambiguous (Section 2.3). By implication, the net of these effects on the maximum score is also an empirical question. Therefore, we are concerned with characterizing the direction and magnitudes of the independent workings of these effects and how added uncertainty affects the net result of added competitors. The results confirm that added uncertainty amplifies the (positive) parallel paths effect and dampens (i.e., makes less negative) the incentive effect. Thus, added uncertainty has an unambiguously positive moderating effect. The net effect of added competitors on Maximum Score depended on the nature of the problem: the net effect of adding competitors on the winning score is negative in the case of canonical single-domain (low uncertainty) problems and positive in the case of multi-domain (high uncertainty) problems.

We assess the moderating effect of uncertainty by first contrasting the simple Average Score and
with the *Maximum Score* to discern key patterns. We then show broader patterns for the wider distribution of expected outcomes. Results are reported in Table 7, beginning with models of *Average Score* to examine the average shift in the distribution of expected outcomes, followed by models of *Maximum Score* to examine the shift in the best score. The models effectively modify the earlier regression specifications by including an interaction term between *No. Competitors* and our proxy for uncertainty in solution approaches, *No. Domains*. (Note that the coefficients on the “direct” terms for *No. Domains* cannot be independently estimated because we are controlling for problem fixed effects.) We have no a priori reason to expect a particular functional form of this interaction. Therefore, we assess several functional forms for both average and maximum room scores. The table includes model results without the moderating effect for purposes of comparison. Preferred models for *Average Score* (4-4) and *Maximum Score* (6-5) are included.

We begin by interacting *No. Competitors* with a discrete indicator for multiple domains (i.e., *No. Domains* > 1), as in models 7-1 and 7-5. The interaction term for the *Maximum Score*, model 7-5, is large and positive (7.65), that for *Average Score*, model 7-1, also large and positive (5.75). These models suggest the main results of the analysis to follow: the effect of added competitors on expected outcomes, as in *Average Score*, is to shift outcomes downward. The effect may be less negative in the case of multi-domain (high uncertainty) problems, but remains absolutely negative. In contrast, the effect of multi-domain problems is to make the effect on the *Maximum Score* go from negative to positive. Not surprisingly, the results involving the (noisy) *Maximum Score* are less precise than those involving the *Average Score*. Models specified as a linear interaction, as in models 7-2 and 7-6, or a quadratic interaction, as in models 7-3 and 7-7, were less significant. We take the discrete interaction with Multi-Domains to therefore be preferred.

The interaction effect on *Maximum Score* is larger than that on *Average Score*, surprisingly, however, not significantly larger. We might have expected that with a strong parallel paths effect spurred by greater uncertainty, the gulf between the response of *Maximum Score* and *Average Score* would have widened appreciably, the idea here being that the response of *Average Score* is about the expected outcomes, accounting for skill or an incentive effect. The difference between the response of *Maximum Score* and the average is thus a rough indication of the magnitude of the parallel paths effect. So it would appear that, in this case, the moderating effect of uncertainty was at least as important on incentives as on the parallel paths effect.

[Table 7 about here]
To show more explicitly how added competitors reshaped distributions of outcomes and the maximum scores, and show that the average score results again reflect general shifts across the distribution of outcomes, we plot in Figures 9 and 10 results of the 10th, 25th, 50th, 75th, and 90th quantile regressions and the maximum score linear regression results. These figures explicitly show the response of these quantiles and the maximum score to varying numbers of competitors, divided by single- and multi-domain problems. Results of the linear estimates of the reshaping of the distribution with No. Competitors are presented. We ran several non-linear models and found no differences. These plots emphasize that the effects of added competitors are not so much to fundamentally transform the distribution of outcomes, but rather to incrementally shift the distributions downward while slightly compressing outcomes. This negative effect and compression are less pronounced in the case of multi-domain problems. More generally, the plots immediately reveal greater variation of outcomes in the case of the multi-domain problems, consistent with greater uncertainty in approach in these cases. Hence, we find support for Hypothesis 3, and we further note that uncertainty in the problem being solved also shifts the incentive response.

6. Discussion and Conclusions

Why do innovation contest organizers typically invite and encourage widespread entry? Canonical economic models suggest that widespread entry should diminish competitors’ incentives to make investments and exert effort (Che and Gale, 2003; Fullerton and McAfee, 1999; Taylor, 1995). Other work has predicted a positive impact of wider competition through “parallel paths effects”, whereby added competitors may increase the chance that at least one will achieve an extreme outcome (Abernathy and Rosenbloom, 1969; Dahan and Mendelson, 2001; Nelson, 1961). These arguments go to the very heart of a key question in the design of innovation contests: how many competitors to let in? For the most part, these economic mechanisms have been examined in separate literatures and with distinct modeling approaches. In a notable exception, Terwiesch and Xu (2008) take important steps to begin to integrate these views within a single analytical framework, with the suggestion that there should be a tension between these effects.

Given the profoundly divergent implications of these distinct views, and limited empirical analysis thereof, the goal of the present analysis was to investigate and document the most basic features of the workings and interplay of these effects—and thus to begin to provide an empirical foundation in understanding these effects at once. We sought to estimate each of these distinct effects empirically,
assess their relative importance, and understand under which conditions one effect might dominate the other.

Our paper makes several contributions to the emerging literature on innovation contests. First, we showed, consistent with theories that focus on incentive effects in one-shot innovation contests, that adding numbers of competitors led to a downward shift in performance outcomes. To past literature that documents a negative effect of competition we add that this effect operates across *all individuals* of varying skills, as we saw the *entire distribution* of outcomes shift downward with greater numbers of competitors. Our results also document this negative incentive effect in the case of a cognitively demanding algorithmic problem, whereas much past empirical research on contests has focused on professional sports, lab conditions, and other contexts more distant from challenging innovation problems.

Second, we demonstrated that, consistent with theories on the parallel paths effect (Dahan and Mendelson, 2001; Girotra et al., 2010; Terwiesch and Xu, 2008), adding competitors resulted in the maximum (winning) score in a competition shifting upward in relation to the distribution of expected outcomes, the parallel paths effect. Essentially, adding competitors generated more “upside” potential to achieve an extreme outcome (in relation to the distribution of expected outcomes). Although abundant theory presumes this effect, we contribute to a nascent literature (see for example Girotra et al., 2010) that quantifies the effect under controlled conditions.

Third, and revealing the key tradeoff between the incentive and parallel paths effects, our empirical isolation and measurement of these effects revealed that whereas added competitors pushed the maximum score in a group of competitors upward in relation to the expected distribution of outcomes (the parallel paths effect), the entire distribution of expected outcomes itself shifted downwards (the incentive effect). That these effects were both large and of comparable magnitude implies the crucial conclusion that neither should be ignored when modeling or designing contests, at least within this context.

Fourth, our analysis shows that whether the (negative) incentive effect or (positive) parallel paths effect dominated the other depended on the contest, and particularly on the amount of uncertainty surrounding the solution to the problem, which could be observed and codified for each contest, ex ante. Thus, not only were the parallel paths and incentive effects of comparable magnitude, but their relative magnitudes differed from contest to contest such that added competitors increased the maximum outcome in the case of problems with higher uncertainty and decreased the maximum outcome in the case of problems with lower uncertainty. Even in this one particular empirical context, we can thus see that the optimal contest design strategy for promoting the quality of the best solution would vary across contests. Our findings begin to explain why at times we observe wide-open contests with extraordinarily large numbers admitted and at other times contests with restricted entry (often practically effected by
“prequalifying” contestants). Uncertainty is thus a key parameter in innovation contest design, affecting both incentive and parallel paths effects.

Fifth, we found that the more positive effect of adding competitors in the case of problems with higher uncertainty was a function of both the parallel paths and incentive effects. Intuitively, added uncertainty positively moderated the parallel paths effect, the likelihood of attaining an extreme outcome with added competitors being greater in cases of higher uncertainty. But we also found that higher uncertainty dampened the negative effect of added competitors on incentives. Although the overall distribution of expected outcomes still shifted downward with added competitors for problems with higher uncertainty, it did not shift downward quite as much. Whereas the moderating effect of uncertainty on incentives is, in principle, ambiguous in the case of a one-shot contest with competitors of heterogeneous skill, this empirical result demonstrates at least one instance in which the reduced incentive effect was as important as a heightened parallel paths effect in generating higher outcomes with added competitors.

Our empirical context presents some elements of novelty, because to date, and to the best of our knowledge, it has been difficult to find innovation contest settings that accommodate the estimation of both incentive and parallel paths effects in which one could claim that, controlling for a series of observable factors, the sources of variation exploited in the econometric analysis are exogenous and, therefore, a basis for estimating causal effects. Moreover, the availability of information on the performance not only of the winner, but of all competitors, allows for more elaborate tests of the response to changes in competition and uncertainty, and facilitates identification of the independent existence of the two effects.

Our results also have direct implications for managers organizing one-off innovation contests or managing platforms for ongoing contests. At the most basic level, managers need to be aware that contests set in motion opposing incentive and parallel paths effects with regard to the number of competitors allowed to participate. Whereas the practitioner literature has mostly celebrated the virtues of contests and open entry (e.g., Lindegaard, 2010; Tapscott and Williams, 2006), realizing that, by definition, most participants lose, and that increasing competition decreases individual incentives, even at the highest quantiles of the performance distribution, should cause managers to have realistic expectations of both the benefits and drawbacks of innovation contests. One implication is that to attract and retain solvers that otherwise may not be “winners,” managers might do well to explicitly create ancillary benefits of participation such as learning, career signaling, and community identification. Managers might also want to consider changing the design of contests such that participation information is revealed strategically, perhaps after the contest, so that the incentive effect does not dominate the parallel paths effect. Our findings about the role of uncertainty in mediating between the incentive and parallel paths
effects also highlight the essential role managers must play in selecting and/or designing the innovation problems to be resolved through contests. In particular, managers need to design contests such that the free entry criteria is reserved for problems with a high degree of uncertainty.

A few important limitations of our study should be taken into account when considering broader questions of innovation contests. The first is the nature of the problems being solved in our empirical context. Although clearly challenging problems intended to demand considerable cognitive effort of elite software developers, we should bear in mind that these were synthetic rather than “naturally occurring” innovation problems. Their synthetic nature enabled us to usefully exploit their observable design as a proxy for uncertainty. In this context, the number of canonical problem-solving domains a problem was to draw from affected the uncertainty of finding an appropriate solution approach. We might expect many naturally occurring innovation problems, perhaps particularly those selected for open contests, to also be considerably more uncertain with respect to the technical approaches to their solutions. What we wish to emphasize in our findings is thus the usefulness of observing a range of uncertainty levels as a way to infer general effects on economic mechanisms and the optimal institutional design with greater or lesser levels of problem uncertainty.

By the same token, it should be emphasized that problems were on an entirely different scale than typical innovation problems, specifically, 75 minutes in duration. Their small scale and repeated nature enabled the systematic comparisons in the foregoing analysis. The costs of participating remain largely fixed and known in these “micro-challenges,” whereas naturally-occurring innovation problems often do not have this character. Further, in these contests we do not observe, say, varying work hours, capital investments, or other discretionary “levels” of investment. Rather, the systematic causal relationships between levels of competition and problem-solving performance relate to effects on cognitive effort, exertion, and effectiveness. Although the general patterns observed conform to theory (whether in relation to cognitive effort or more mundane costly investments), the results more directly reflect contests in which we observe the behavior of individual people who compete rather than a context in which, say, firms decide investment levels more generally. The central point and emphasis of the paper is that incentive effects (whatever their nature and origin) appear to coexist with the parallel paths effect. Analogously, the motivations in this context extend beyond cash rewards. A range of winner-take-all and more continuous higher-is-better rewards plays a role in these contests. Several of these are more particular to individuals rather than, say, to larger, profit-seeking organizations.

Finally, this article began by emphasizing that game theoretic models of innovation focused on strategic interactions generate the prediction that entry should be restricted, and that models of stochastic outcomes based on order statistical arguments tend to argue the opposite. The results presented in this article suggest that neither order statistic arguments related to parallel paths nor game theoretic arguments
related to strategic incentives should be ignored in modeling or designing innovation contests. This is not just a substantive finding in its own right, but suggests that current traditions of modeling innovation contests, modeling just one set of mechanisms but not the other, may largely ignore key interactions and tradeoffs. To our knowledge, only Terwiesch and Xu (2008) have begun to make progress in integrating these issues thus far.
References


Table 1 The Canonical Problem Types

<table>
<thead>
<tr>
<th>Knowledge Category</th>
<th>No. of Problems Tagged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption/Compression</td>
<td>19</td>
</tr>
<tr>
<td>Advanced Math</td>
<td>63</td>
</tr>
<tr>
<td>Greedy</td>
<td>84</td>
</tr>
<tr>
<td>Sorting</td>
<td>99</td>
</tr>
<tr>
<td>Recursion</td>
<td>117</td>
</tr>
<tr>
<td>Geometry</td>
<td>119</td>
</tr>
<tr>
<td>String Parsing</td>
<td>128</td>
</tr>
<tr>
<td>Simple Search, Iteration</td>
<td>148</td>
</tr>
<tr>
<td>Graph Theory</td>
<td>151</td>
</tr>
<tr>
<td>Simulation</td>
<td>157</td>
</tr>
<tr>
<td>Search</td>
<td>170</td>
</tr>
<tr>
<td>String Manipulation</td>
<td>192</td>
</tr>
<tr>
<td>Math</td>
<td>202</td>
</tr>
<tr>
<td>Simple Math</td>
<td>213</td>
</tr>
<tr>
<td>Dynamic Programming</td>
<td>245</td>
</tr>
<tr>
<td>Brute Force</td>
<td>251</td>
</tr>
</tbody>
</table>

Notes. The number of problems associated with different problem types exceeds the count of problems in the population, as roughly the half the problems are tagged as belonging to multiple categories.
### Table 2 Variable Definitions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Score</td>
<td>The final score awarded to a given solution to a problem</td>
</tr>
<tr>
<td>(2) No. Competitors</td>
<td>Number of competitors directly competing with one another in a room</td>
</tr>
<tr>
<td>(3) Average Score</td>
<td>Total number of points awarded to competitors in a given room for a given problem, divided by No. Competitors</td>
</tr>
<tr>
<td>(4) Maximum Score</td>
<td>Highest or winning score within a room</td>
</tr>
<tr>
<td>(5) Skill Rating</td>
<td>Numerical evaluation of a competitor's skill, based on history of performance</td>
</tr>
<tr>
<td>(6) Average Skill Rating</td>
<td>Total Skill Rating in a room, divided by No. Competitors</td>
</tr>
<tr>
<td>(7) Variance Skill Rating</td>
<td>Standard deviation (second moment) of Skill Rating in a room</td>
</tr>
<tr>
<td>(8) Skewness Skill Rating</td>
<td>Skew (third moment) of Skill Rating in a room</td>
</tr>
<tr>
<td>(9) Maximum Skill Rating</td>
<td>Highest Skill Rating of all competitors in a room</td>
</tr>
<tr>
<td>(10) No. Domains</td>
<td>Count of the number of canonical problem/solution types that are part of the problem</td>
</tr>
</tbody>
</table>
Table 3 Descriptive Statistics and Correlations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
<th>(7)</th>
<th>(8)</th>
<th>(9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Score</td>
<td>300.1</td>
<td>300.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) No. Competitors</td>
<td>18.5</td>
<td>1.2</td>
<td>-.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3) Average Score</td>
<td>283.4</td>
<td>129.6</td>
<td>.44</td>
<td>-.09</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) Maximum Score</td>
<td>313.7</td>
<td>212.7</td>
<td>.20</td>
<td>-.02</td>
<td>.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) Skill Rating</td>
<td>1689.2</td>
<td>412.1</td>
<td>.58</td>
<td>-.05</td>
<td>.17</td>
<td>.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6) Average Skill Rating</td>
<td>1751.6</td>
<td>239.2</td>
<td>.15</td>
<td>-.05</td>
<td>.29</td>
<td>.17</td>
<td>.26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7) Variance Skill Rating</td>
<td>422.9</td>
<td>164.8</td>
<td>.09</td>
<td>-.02</td>
<td>.21</td>
<td>.13</td>
<td>.15</td>
<td>.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(8) Skewness Skill Rating</td>
<td>.9</td>
<td>.5</td>
<td>-.05</td>
<td>.07</td>
<td>-.13</td>
<td>-.01</td>
<td>-.13</td>
<td>-.22</td>
<td>.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(9) Maximum Skill Rating</td>
<td>2366.4</td>
<td>438.5</td>
<td>.11</td>
<td>.00</td>
<td>.27</td>
<td>.16</td>
<td>.20</td>
<td>.74</td>
<td>.84</td>
<td>.06</td>
<td></td>
</tr>
<tr>
<td>(10) No. Domains</td>
<td>1.8</td>
<td>.8</td>
<td>-.04</td>
<td>.05</td>
<td>-.09</td>
<td>-.06</td>
<td>.01</td>
<td>.02</td>
<td>.00</td>
<td>.00</td>
<td>.01</td>
</tr>
</tbody>
</table>
### Table 4 Baseline OLS Regression Results of Overall Shifts in Performance Outcomes (*Average Score*) on Numbers of Competitors (*No. Competitors*)

<table>
<thead>
<tr>
<th>Dependent Variable:</th>
<th>Average Score</th>
<th>Alternative Dependent Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model:</td>
<td>4-1</td>
<td>4-2</td>
</tr>
<tr>
<td></td>
<td>Problem Fixed Effects</td>
<td>Control for Avg Room Skills</td>
</tr>
<tr>
<td><em>No. Competitors</em></td>
<td>-.924*** (.84)</td>
<td>-.508*** (.73)</td>
</tr>
<tr>
<td><em>Skills Rating Distribution</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Average</em></td>
<td>.18*** (.00)</td>
<td>.14*** (.01)</td>
</tr>
<tr>
<td><em>Average Skill Rating (Nonparametric)</em></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><em>Variance</em></td>
<td>.02*** (.01)</td>
<td>.04*** (.01)</td>
</tr>
<tr>
<td><em>Skewness</em></td>
<td>-22.16*** (.53)</td>
<td>-26.03*** (.58)</td>
</tr>
<tr>
<td><em>Maximum</em></td>
<td>.02*** (.01)</td>
<td>.00</td>
</tr>
<tr>
<td>Problem Fixed Effects</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Adjusted R-Squared</td>
<td>.60</td>
<td>.70</td>
</tr>
</tbody>
</table>

*Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; autocorrelation-heteroskedasticity robust standard errors reported; number of observations = 9,661 room-problems.*
Table 5 Regression of Baseline Model on Subsamples and Out-of-Sample Data

<table>
<thead>
<tr>
<th>Explanatory Variables</th>
<th>5-1</th>
<th>5-2</th>
<th>5-3</th>
<th>5-4</th>
<th>5-5</th>
<th>5-6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Extended Range of N (Out of Sample)</td>
<td>No Prize</td>
<td>&quot;Big&quot; Rounds</td>
<td>&quot;Small&quot; Rounds</td>
<td>Post-2004</td>
<td>Pre-2004</td>
</tr>
<tr>
<td>No. Competitors</td>
<td>-5.39***</td>
<td>-5.74***</td>
<td>-4.21***</td>
<td>-5.56***</td>
<td>-3.73***</td>
<td>-9.66***</td>
</tr>
<tr>
<td></td>
<td>(.88)</td>
<td>(.88)</td>
<td>(.87)</td>
<td>(1.26)</td>
<td>(.74)</td>
<td>(1.73)</td>
</tr>
<tr>
<td>Skills Rating Distribution</td>
<td>.16***</td>
<td>.15***</td>
<td>.14***</td>
<td>.14***</td>
<td>.13***</td>
<td>.19***</td>
</tr>
<tr>
<td>Average</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
</tr>
<tr>
<td>Variance</td>
<td>-.01</td>
<td>.02**</td>
<td>.03***</td>
<td>.00</td>
<td>.02**</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.02)</td>
<td>(.01)</td>
<td>(.02)</td>
</tr>
<tr>
<td></td>
<td>(1.51)</td>
<td>(1.83)</td>
<td>(1.85)</td>
<td>(2.73)</td>
<td>(1.60)</td>
<td>(3.25)</td>
</tr>
<tr>
<td>Maximum</td>
<td>.02***</td>
<td>.02***</td>
<td>.01**</td>
<td>.01***</td>
<td>.02***</td>
<td>.00</td>
</tr>
<tr>
<td></td>
<td>(.00)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
</tr>
<tr>
<td>Problem Fixed Effects</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Adjusted R-Squared</td>
<td>.72</td>
<td>.72</td>
<td>.73</td>
<td>.67</td>
<td>.72</td>
<td>.72</td>
</tr>
<tr>
<td>Number of Observations</td>
<td>13156</td>
<td>7219</td>
<td>4831</td>
<td>4830</td>
<td>7484</td>
<td>2177</td>
</tr>
</tbody>
</table>

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; autocorrelation-heteroskedasticity robust standard errors reported. "Big" rounds have greater total participation across all rooms in a given round than the median of other rounds during the same calendar year; "Small" have less than or equal to the median.
Table 6 Regression Results of Distribution of Performance Outcomes (*Score*) and Winning Performance (Maximum Score) on Numbers of Competitors (*No. Competitors*)

<table>
<thead>
<tr>
<th>Dependent Variable:</th>
<th>Score</th>
<th>6-1</th>
<th>6-2</th>
<th>6-3</th>
<th>6-4</th>
<th>6-5</th>
<th>Maximum*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explanatory Variables</td>
<td>q25</td>
<td>q50</td>
<td>q75</td>
<td>q90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. Competitors</td>
<td>-1.74***</td>
<td>-1.68***</td>
<td>-5.08***</td>
<td>-9.39***</td>
<td>-.88</td>
<td></td>
<td>(1.52)</td>
</tr>
<tr>
<td></td>
<td>(.56)</td>
<td>(.55)</td>
<td>(1.18)</td>
<td>(1.82)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Skills Rating Distribution

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>.09***</th>
<th>.12***</th>
<th>.23***</th>
<th>.33***</th>
<th>.11***</th>
<th>(.01)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(.00)</td>
<td>(.00)</td>
<td>(.01)</td>
<td>(.01)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(.01)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-.02**</td>
<td>.00</td>
<td>.07***</td>
<td>.19***</td>
<td>.04**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.02)</td>
<td>(.02)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(.02)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-15.01***</td>
<td>-21.13***</td>
<td>-35.52***</td>
<td>-19.86***</td>
<td>6.25*</td>
<td>(3.32)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1.14)</td>
<td>(1.14)</td>
<td>(2.47)</td>
<td>(3.75)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>.00</td>
<td>.00</td>
<td>-.01</td>
<td>-.02</td>
<td>.02**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.00)</td>
<td>(.00)</td>
<td>(.01)</td>
<td>(.01)</td>
<td>(.01)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem Fixed Effects</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constant</td>
<td>-178.59***</td>
<td>-55.33***</td>
<td>117.45***</td>
<td>333.26***</td>
<td>372.10***</td>
<td>(32.36)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(54)</td>
<td>(53)</td>
<td>(1.15)</td>
<td>(1.77)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; models (1-4) are estimated with weighted least absolute deviations, with standard errors following Koenker and Bassett (1978); model (5) is estimated with ordinary least squares and autocorrelation-heteroskedasticity robust standard errors; (+) the constant in the regression model with Maximum Score as dependent variable with problem fixed effects included is set to zero on account of the inclusion of problem fixed effects – the reported constant is the mean level of this variable without problem fixed effects to allow more direct comparison with constants in the quantile regression; models (1-4) number of observations = 162,561 round-problem-individuals; model (5) number of observations = 9,661 round-problem-rooms.
Table 7 Regression Results on the Moderating Effect of Varying Levels of Uncertainty (No. Domains)

<table>
<thead>
<tr>
<th>Dependent Variable:</th>
<th>Average Score</th>
<th>Maximum Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model: 4-4</td>
<td>7-1</td>
<td>7-2</td>
</tr>
<tr>
<td>No Interactions</td>
<td>Interaction with Multi-Domain Problems</td>
<td>Interaction with Multi-Domain Problems</td>
</tr>
<tr>
<td>No. Competitors</td>
<td>-4.63*** (.72)</td>
<td>-8.07*** (1.16)</td>
</tr>
<tr>
<td>No. Competitors x (Multiple Domains)</td>
<td>5.75*** (1.46)</td>
<td></td>
</tr>
<tr>
<td>No. Competitors x No. Domains</td>
<td>4.02*** (.83)</td>
<td>3.94 (.39)</td>
</tr>
<tr>
<td>No. Competitors x No. Domains²</td>
<td>.2 (.79)</td>
<td></td>
</tr>
</tbody>
</table>

**Skills Rating Distribution**

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Variance</th>
<th>Skewness</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.14*** (.01)</td>
<td>0.02** (.01)</td>
<td>-22.16*** (1.53)</td>
<td>0.02*** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.14*** (.01)</td>
<td>0.02** (.01)</td>
<td>-22.04*** (1.53)</td>
<td>0.02*** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.14*** (.01)</td>
<td>0.02** (.01)</td>
<td>-22.07*** (1.53)</td>
<td>0.02*** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.14*** (.01)</td>
<td>0.02** (.01)</td>
<td>-22.07*** (1.53)</td>
<td>0.02*** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.11*** (.01)</td>
<td>0.04* (.02)</td>
<td>6.25* (3.32)</td>
<td>0.02** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.11*** (.01)</td>
<td>0.04* (.02)</td>
<td>6.40* (3.32)</td>
<td>0.02** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.11*** (.01)</td>
<td>0.04* (.02)</td>
<td>6.33* (3.32)</td>
<td>0.02** (.01)</td>
</tr>
<tr>
<td></td>
<td>0.11*** (.01)</td>
<td>0.04* (.02)</td>
<td>6.38* (3.32)</td>
<td>0.02** (.01)</td>
</tr>
</tbody>
</table>

**Problem Fixed Effects**

<table>
<thead>
<tr>
<th></th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
</tr>
</thead>
</table>

**Adjusted R-Squared**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.72</td>
<td>.72</td>
<td>.72</td>
<td>.72</td>
<td>.53</td>
<td>.53</td>
<td>.53</td>
</tr>
</tbody>
</table>

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; autocorrelation-heteroskedasticity robust standard errors reported; number of observations = 9,661 room-problems.
Figure 1 - Structure of Weekly Events or “Rounds”

<table>
<thead>
<tr>
<th>Competition Round 38</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Tues Jan 23 2001)</td>
<td>“Room”</td>
<td>“Room”</td>
<td>Prob 231</td>
<td>Prob 232</td>
<td>Prob 233</td>
<td></td>
</tr>
<tr>
<td>11:15am EST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Competition Round 39</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Fri Feb 2 2001)</td>
<td>“Room”</td>
<td>“Room”</td>
<td>“Room”</td>
<td>“Room”</td>
<td>Prob 234</td>
<td>Prob 235</td>
</tr>
<tr>
<td>3pm EST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Competition Round 40</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Wed Feb 14 2001)</td>
<td>“Room”</td>
<td>“Room”</td>
<td>“Room”</td>
<td>Prob 237</td>
<td>Prob 238</td>
<td>Prob 239</td>
</tr>
<tr>
<td>1pm EST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 2 - Typical Public Profile of a TopCoder Competitor
Figure 3 – Skills Distribution at TopCoder
Figure 4 – Distribution of Score

I. Score Histogram

II. Score Distribution (FE's)
Figure 5 – Distribution of No. Competitors
Figure 6 – Distribution of *Skill Rating*, Stratified by *No. Competitors*
Figure 7 - Relationship between Score and No. Competitors across Wider Domain (Out-of-Sample Data)
Figure 8 – Change across the Distribution of Performance Outcomes with an Added Competitor

Co-efficient estimates from quantile regressions performed at 5 percentile intervals presented with 90% confidence intervals. The response of maximum score is shown at the position of the 100th quantile.
Figure 9 - Change in Response Due to Added Competitor for Single and Multi-Domain Problems

Quantile Response
Multi-Domain Problems

Black relates to single-domain problems; grey relates to multi-domain problems.
Figure 10 Distribution of Performance Outcomes (Score) versus Numbers of Competitors (No. Competitors)

Lines are q10, q25, q50 (bold), q75, q90 and max (red).
Appendix A
Contest Vignette – Basketball Strategy Problem

The following vignette details contest dynamics for a sample problem in the TopCoder algorithm contests. The vignette shows the competition room setup with varying numbers of competitors working on the same problem, the performance distribution, the exact problem statement, and a solution synopsis from one of the contestants.

1) Competition Room Setup
This problem was in a TopCoder algorithm contest conducted on July 26, 2006 and was internally classified as straddling the following three knowledge domains: Geometry, Graph Theory, and Math. There were 397 competitors in Division I assigned to solve this problem. The participants were distributed among 20 project rooms according to the distribution shown in Figure A1.

![Figure A1: Number of Competition Rooms by Size](image_url)

2) Results and Performance Distribution
Overall, 338 individual actually opened the problem. Only 66 submitted solutions and 47 passed the system tests. Indicating that this problem posed a significant challenge to even the most elite TopCoder developers, Figure A2 shows the distribution of participants’ scores.
3) Problem Statement
The following is the exact problem statement as seen by the contest participants. We have permission from TopCoder to present it below.

**Basketball Strategy**
In a simplified version of basketball the goal is to score by getting the ball in a special scoring place. There are two teams, and each team contains the same number of players. When a player has possession of the ball, he has two choices: take a shot, or pass the ball to a teammate. When taking a shot, the player throws the ball in a straight line to the scoring place. When passing the ball, he throws the ball in a straight line to the target teammate. In both cases, at most one of the rival players will try to intercept the shot or pass.

The probability of a pass being successful is:

\[ \text{Cp} \times (1 - (\text{Is} / 150)^2) \times \frac{\text{dr}}{\text{dr} + 1} \]

And the probability of a shot being successful (score) is:

\[ (\text{Cs} \times \frac{\text{dr}}{\text{dr} + 1}) \ln(\text{Is}) \]

Where Cp and Cs are constants defined for the problem instance, Is is the length of the shot or pass, dr is the distance between the intercepting rival and the ball trajectory and ln is the natural logarithm (logarithm in base e).
When trying to intercept a shot or a pass, only the best suitable player of the other team to do so (i.e., the one that produces the lowest $dr$) will try. If no player on the other team can do it, the factor $dr/(dr+1)$ in the formula is considered to have a value of 1 (i.e., it is ignored). A player of the rival team is only allowed to try to intercept the ball if the line that passes through him and is perpendicular to the ball trajectory intersects the trajectory at some point between the two endpoints of the trajectory, inclusive.

For example, in this picture:

There are 3 players in each team, green players are your team and red players are rivals. Player 0 has the ball and has 3 options marked as blue lines, 2 passes and taking a shot. The shot, if taken, can be intercepted by any of the rivals, but only number 2 will try because he is clearly the nearest. The pass to player 1 is impossible to intercept for the rivals, because any player that can intercept that pass should be inside the gray area. The pass to player 2 can be intercepted by rivals 1 or 2. Rival player 0 is not on an intersecting perpendicular line, so he cannot try to intercept it. In this last case, rival 1 will try to intercept because he is nearer than rival 2.

You will be given two String[]s team and rivals with the same number of elements representing the members of each team. Each element of team and rivals will be in the format "X Y" where X and Y will be positive integers with no leading zeroes representing the x and y coordinates of that player in the field. You will also be given $C_p$ and $C_s$, the constants for the probability calculations of each type of movement. When the game starts, the ball is in possession of the player on your team with index 0. The scoring place is at X=50, Y=0. Your team is only allowed to take one shot, and you are to determine and return the probability that your team will score if it follows the best strategy. A strategy consists of zero or more passes followed by a shot. If your team loses the ball at any point during the strategy, you will not score.

Definition
Class: BasketballStrategy
Method: scoreProbability
Parameters: String[], String[], double, double
Returns: double
Method signature: double scoreProbability(String[] team, String[] rivals, double $C_p$, double $C_s$)
(be sure your method is public)

Notes
- The returned value must be accurate to within a relative or absolute value of $1E^{-9}$.
- Pictures are just approximations. The players are considered to be perfect points with 0 surface and 0 length and trajectories and other lines are perfect lines with 0 surface.
- The same rival may try to intercept many passes along the game (see example 3 for further clarification).

**Constraints**
- Team and rivals will each contain exactly $N$ elements, where $N$ is between 1 and 50, inclusive.
- Each element of team and rivals will be two integers between 1 and 99, inclusive, with no leading zeroes, separated with exactly one space character, with no leading or trailing spaces.
- All elements of team and rivals together will be distinct.
- $C_p$ and $C_s$ will each be greater than 0 and less than or equal to 1.


**4) Solution Synopsis by Member “soul-net”**

The following is a solution synopsis of the problem by the TopCoder member “soul-net” on TopCoder’s website. We have replicated it below with their permission.

This problem is pretty straightforward if you know a little about many aspects of programming. You needed some, but not much, of:\n- Probabilities
- Graph theory
- Math
- Integer or algebraic geometry
- Floating point or euclidian geometry

Let's take it one step at a time, first what was obviously needed was to know the exact probability of an arbitrary pass between two of your players or an arbitrary shot. This two things were similar, with the only difference of the final formula.

Let's take it one step at a time, first what was obviously needed was to know the exact probability of an arbitrary pass between two of your players or an arbitrary shot. This two things were similar, with the only difference of the final formula.

To make everything simpler it is better to have a point structure or class defined that has two double precision (double) floating point members: x and y. Also have the + and - operators as applying the operator to both x and y. Since the only non-given variables of the formulas are $ls$ and $dr$, we will calculate them and then apply the corresponding formula with the corresponding given constant.

```java
double length(point from, point to)    set relTo = to - from    return sqrt(relTo.x^2+relTo.y^2)
double probability(bool isShot, point from, point to, point[] rivals)    double ls,dr;    set ls=length(from, to)    set dr = calculateDr(from, to, rivals)    if isShot       return applyShotFormula(ls, dr)    else       return applyPassFormula(ls, dr)
```

Now we need the calculation of $dr$ function:

```java
double calculateDr(point from, point to, point[] rivals)    double dr = 10^200    //note here that making dr really big by default //makes the $dr/(dr+1)$ term practically equal to 1    //if no rival can intercept    for each r in rivals       if canIntercept(from, to, r)          dr = min(dr, perpendicularDistance(from, to, r))    return dr
```

Up to this point everything is easy and intuitive. Let's get to the first problem: how far is a rival that we know we can intercept?

---

$^{21}$ Note that this classification of knowledge domains needed to solve this problem was derived independently by this member.
If we see the 3 points as a triangle, the perpendicular line that passes through the rival point is one height, and since we can easily calculate the base corresponding to that height (because is the distance of the pass) and also the area of the triangle (using cross product), we have it all:

```plaintext
double perpendicularDistance(pint from, point to, point r)    set area = triangleArea(from, to, r)    set base = length(from, to) return area * 2 / base
double triangleArea(point p1, point p2, point p3) return abs(crossProduct(p2-p1,p3-p1))/2
double crossProduct(point p, point q) return p.x*q.y - q.x*p.y
```

About testing for ability of each rival to intercept the trajectory, as was discussed in the forums, it can be done with pure integer arithmetic, avoiding small precision errors. In general, when you need to test a boolean condition it's almost a must to use only integers; a small precision error can easily change true to false or false to true and that could further lead to much bigger precision errors (see the linked thread in the forums with a better explanation and discussion of this point).

Let's see the triangle formed again. For a rival to be in position to intercept the trajectory, the two angles formed by the trajectory line and each of the lines that connect the rival with each of your players have to be less than or equal to 90 degrees. Testing this is equivalent to say that the dot product of the relative vectors is greater than 0 (this follows from algebraic definition of angle, that can be found in the link above). Following this reasoning, the code for this part is:

```plaintext
//for this part we need the points to have integer coordinates or use //different points structures boolean canIntercept(point from, point to, point r) return dotProduct(from-to,r-to) >= 0 and dotProduct(to-from,r-from) >= 0 int dotProduct(point p, point q) return p.x*q.x+p.y*q.y
```

At this point we have left geometry behind and can easily build a lovely graph that has one node for each of our players and one node for the scoring place. Each pair of nodes has a connecting edge that is labeled with the probability of that pass or shot being successful. What we need now is a path in that graph that goes from the starting player to the scoring place such that the product of the labels each traversed edge (the probability of the strategy represented by that path) is maximum.

For some people it's obvious that you can easily adapt any shortest path algorithm to solve this, simply changing minimum for maximum and adding for multiplying. If you are not part of that group, or would like to reuse your prewritten Dijkstra, Floyd-Warshall or Bellman-Ford without modifying anything inside, see the following magical idea.

First step: Build the minus logarithm graph. This graph is the same but each label l is transformed to -log(l) (the base of the logarithm is not important).

Second step: Find the shortest path from player 0 to scoring place in this new graph. Since this starts to sound too crazy, let's explain and continue after that. Shortest path in this new graph is a path in which the sum of the new labels is minimum. This sum, in terms of old labels, can be written as:

```
(-log(l1)) + (-log(l2)) + ... + (-log(ln))
```

where li are the labels in the original graph of each corresponding edge. From that expression we can derive the following:

```
(-log(l1)) + (-log(l2)) + ... + (-log(ln)) =
- ((log(l1) + log(l2) + ... + log(ln)) =
-log(l1 * l2 * ... * ln)
```

Since this last expression was minimized (because we looked for a shortest path), that means that the opposite expression was maximized. And since the logarithm is an strictly monotonically increasing function the inside expression

```
l1 * l2 * ... * ln
```

is maximal. To find it's value, simply take L = the length of the shortest path found, and calculate b-L, where b was the base used for logarithms (any positive base is ok).
To see any shortest path algorithm work, think that since original labels lie on (0;1] interval, the logarithm is always non-positive and the minus logarithm is therefore always non-negative.

After seeing this "magical" approach, maybe you are convinced that the initially mentioned approach of simply changing < for > and + for * works. If you are not, maybe you should. And if you don't think you should, come discuss in the forums.

For an implementation you can see zhuzeyuan's code, its similarly modularized to this presentation (he uses Dijkstra and generates the labels of the graph on demand). To see the integer arithmetic in intercept ability test, see dskloet's code.

Source: http://www.topcoder.com/tc?module=Static&d1=match_editorials&d2=srm313