Parallel Search, Incentives and Problem Type: Revisiting the Competition and Innovation Link[1]

Kevin J. Boudreau
HEC Paris

Nicola Lacetera
Case Western Reserve University

Karim R. Lakhani
Harvard Business School

This paper presents econometric evidence of two independent effects of adding more competitors on innovation: 1) a *competition effect* whereby increasing rivalry shapes, and often decreases, incentives to expend effort and invest in innovation; and 2) a *parallel search effect* whereby adding greater numbers of "searchers" benefits innovation by broadening the search for solutions. We further show the importance of these effects depends on the nature of the innovation problem being solved. The analysis uses data from TopCoder's software contest platform, on which elite software developers were assigned different problems to solve within assigned groups of direct competitors. Econometric relationships are identified by exploiting random assignment and a separate instrumental variables procedure.

Key words: Competition, innovation, problem solving, search, complexity

1. Introduction

Success at innovation is central to organizational growth and survival and the betterment of societies. Fundamental to innovation is the ability to successfully solve scientific, technical and design problems that do not necessarily have obvious solutions—or even obvious solution approaches *ex ante* (Nelson 1959; Abernathy and Utterback 1978). Because R&D can sometimes be fraught with uncertainty, one leading view casts innovation and problem solving as a process of "search" over some poorly understood knowledge landscape (Simon and Newell 1962; Levinthal 1997). Not knowing the appropriate solution or solution approach is tantamount to not knowing what skill sets, capabilities or orientations are needed to successfully conclude the search process. Consequently, a "parallel search" approach whereby multiple independent solvers (or teams of solvers) compete to solve the *same* innovation problem has been recommended as a means of bringing a variety of skills and approaches to bear on a problem (Nelson 1959, 1961;

[1] We would like to thank Jack Hughes, Mike Lydon and the staff of TopCoder.com for providing us with the data for this study and answering numerous questions about their programming competitions. Participants in presentations at Harvard Business School, HEC-Paris, Imperial College, London Business School, CWRU and the Wharton Technology Miniconference provided significant suggestions for improvements. All mistakes remain our own.

Abernathy and Rosenbloom 1969). Examples of this principle abound, including introducing many diverse competitors to a market and using multiple internal R&D teams to investigate distinct design approaches (Abernathy and Rosenbloom 1969; Gerchak and Kilgour 1999; Ding and Eliashberg 2002), stimulating "open innovation" around a platform (Chesbrough 2002) and "democratizing innovation," more generally (von Hippel 2005).

At the same time, most economic theory on entry and competition across a wide variety of institutional contexts argues that high numbers of competitors will reduce innovators' incentives to exert effort and invest (e.g., Salop 1979; Aghion and Howitt 1992; Taylor 1995). Its prescription is thus the opposite: restrict the number of competitors. This leaves an important theoretical gap, as a pure problem solving and "parallel search" view does not account for individual innovators' incentives and strategic responses to competition (Dosi et al. 2003). This prompts several questions. *Do parallel search and competition effects coexist? If they do, is it simply the case that they work against one another? Do more subtle interactions and determinants govern how they shape innovation?* This paper presents econometric evidence to show that parallel search effects and competition effects coexist and their relative strength depends on the nature of the innovation problem being solved.

Setting our study in a context explicitly designed with parallel search in mind enables us to look for how competition might also play a role. TopCoder Inc. serves outsourced software development projects to global Fortune 1000 firms. It does so by acting as a two-sided platform (Rochet and Tirole 2006) with a large set of independent elite programmers who enter into contests to compete against one another to solve problems of client firms. We use data on 645 problems, on which 9,627 sets of 15-20 direct competitors worked. We observe for each case problem-solving performance, numbers of competitors, competitor skills and the type of problem being solved.

The fundamental approach of our analysis is to exploit the fact that competition effects on incentives could be discerned by observing how numbers of competitors affected individual competitors' performance. By contrast, parallel search effects could be discerned by observing how numbers of competitors led to changes in the maximum or best innovative performance (above and beyond what would be predicted by changing incentives on their own). Key relationships are estimated by exploiting random assignment of competitors to different groups of direct competitors and problems. Because assignment to groups of direct competitors was not perfectly random, the econometric approach focuses on removing sources of variation in the estimation that could bias estimates. The approach is corroborated using an instrumental variables procedure that exploits variation in the time of day a contest starts as a source of exogenous variation that affects numbers of competitors.

We found that parallel search effects and competition effects coexisted: adding more competitors to a group of direct competitors solving the same problem systematically reduced the problem-solving scores of *individual* competitors and *at the same time* positively affected the *maximum* or best score attained within the group. Corroborating tests and anecdotal information from company interviews confirm these results. We further show that the magnitude of both the parallel search and competition effects depended on whether problems were "complex" or not (Simon 1969) in the sense of implicating a plurality of knowledge sets (Jonassen 2004; Nickerson and Zenger 2004; Macher 2006). We interpret these findings as suggesting that problem complexity does not only change the nature of the innovation search process, but also has implications for strategic interactions and the structure of competition.

The paper contributes to separate streams of empirical research that have provided econometric evidence on parallel search (Cohen and Malerba 2001; Leiponen and Helfat 2007) and competition (Cohen and Levin 1989; Cohen 1995; Nickell 1996; Aghion et al. 2005) effects. Our analysis bridges these traditions by presenting evidence of the mechanisms at work together.

Theoretically, the paper contributes to growing theory development and evidence on the mechanisms of problem solving and search (Simon 1947; Cyert and March 1963; Nelson and Winter 1982; Levinthal 1997; Gavetti and Levinthal 2000; Rivkin 2000; Hong and Page 2001, 2004)—in a context in which agents are *competing* with one another. In doing so, the results drawparticularly on recent work by Terweisch and Xu's (forthcoming). By virtue of the empirical setting, the results also contribute to research on distributed innovation and problem solving around a platform (Bresnahan and Greenstein 1999; Chesbrough 2002; Gawer and Cusumano 2002; von Hippel 2005), providing an indication of which sorts of innovation-related problems are most likely to benefit from "opening up" innovation.

The paper proceeds as follows. Section 2 reviews the literature with the objective of developing empirical hypotheses regarding how competition effects and parallel search effects might coexist. Section 3 details the empirical context and data set used in the analysis. Section 4 explains the approach used in the empirical analysis. Section 5 presents results. Section 6 summarizes our intended contribution and concluding remarks.

2. Literature and Hypothesis Development

In this section, we review two distinct ideas concerning how adding more agents to a group of direct competitors might affect innovation outcomes. One idea concerns how increasing competition affects agents' incentives to invest—a "competition effect." The other concerns how adding agents might add useful diversity to the search process—a "parallel search effect." The goal here is to develop empirical hypotheses concerning (1) how these effects can be distinguished, and (2) how they might relate to one another and interact.

2.1. The Competition Effect

When innovation is organized as a set of directly competing innovators, how many competitors are best? This question has been tackled by a variety of literatures in industrial and organizational economics that cover distinct institutional contexts. A remarkably consistent result across this work is that high numbers of symmetrical competing agents will lower overall innovation performance.

For example, classical work in economics examining the link between innovation and competition has shown that as markets become more crowded, potential entrants have less incentive to join (Dixit and Stiglitz 1977; Salop 1979) as they will capture a smaller share of the market while still bearing the fixed costs of research. Similarly, endogenous growth models of the type of Romer (1990) and Aghion and Howitt (1992) show that innovation declines as competition and threat of imitation increase. A number of theories (and some evidence) suggest that increased rivalry among symmetric competitors can in fact stimulate investment (Nickell 1996; Raith 2003; Aghion et al. 2005). But these same models indicate that the effect should become negative, at least at high levels of competition. (See Aghion and Griffith (2005) for a review of the literature.)

This basic argument of depressed incentives with high competition seems relatively insensitive to different innovation settings. For example, the budding literature on research contests (in which, instead of splitting the market, just one or few agents win) arrives at similar conclusions. Taylor (1995) show that allowing too many competitors reduces the total level of research effort and overall innovation outcome, as contestants find the probability of winning lowered. Fullerton and McAfee (1999)—extending Taylor’s model to consider heterogeneous competitors and a variety of tournament designs—come to the conclusion that limiting entry is preferred. (They conclude that the optimal number of contestants is, in fact, just two.) A similar analytic finding has been reported by Che and Gale (2003) with the assumption that agents invest in the quality of their innovations.

Similar predictions of negative effects of competition on investment have been argued in relation to a range of other institutional contexts including competing internal R&D projects (Rotemberg and Saloner 1994; Gerchak and Kilgour 1999), competing outsourcing partners (Bakos and Brynjolfsson 1993), "competing" employees (Rajan and Zingales 1998) and the adoption of the same technology standard by competing firms (Ellison and Fudenberg 2003; Augereau et al. 2006).

Thus, our first and baseline hypothesis for what happens when the number of competitors increases is as follows:

> *Hypothesis 1 ("COMPETITION EFFECT") Increasing numbers of direct competitors will systematically reduce the innovative performance of individual competitors.*[2]

2.2. The Parallel Search Effect

Notwithstanding the seeming universality of the competition effect result, there remains an equally pervasive intuition that "safety would seem to lie in numbers and variety of attack." This was one of the important conclusions reached by Jewkes et al. (1959, pg. 246) that launched a distinct literature that stressed the *uncertainty*, and sometimes *fundamental uncertainty*, of the innovation process (Nelson 1959, 1961; Abernathy and Rosenbloom 1969).

In contrast to just "risky" or stochastic outcomes, fundamental uncertainty implies an inability to formulate objective prior beliefs (Knight 1921). But as a practical matter, analytical models developed in this tradition have been formulated in terms of drawing "balls from an urn" with some *known* distribution of possible outcomes (e.g., Marshall 1961).[3] Therefore, notwithstanding deep conceptual differences in this tradition, the models themselves do not substantially differ

[2] Innovation incentives should be "crowded out" at least at high levels of competition. As mentioned earlier, the competition effect may in fact stimulate investment at lower levels of competition.

[3] This drawing-balls-from-an-urn approach has been adopted by and furthered in other research streams such as that on product modularity (Baldwin and Clark 2000) and product management (Dahan and Mendelson 2001; Terwiesch and Loch 2004; Terwiesch and Xu Forthcoming).

from earlier-mentioned models of incentives and competition (which themselves often consider probabilistic outcomes).

A more important difference in the mechanics of these stochastic parallel search models is their focus of analysis. Rather than focus on *expected* outcomes of innovation that result from rational investment choices, stochastic parallel search models consider variance, and model how the *maximum* innovation performance will be affected based on *nth* order statistic calculations. Following this logic, insofar as adding competitors contributes greater variation in outcomes, the maximum score should systematically increase with numbers of competitors. In cases in which innovation outcomes are stochastic and risky across innovators, for whatever reason, we should therefore expect the maximum innovation performance in a group to be affected by both the competition effect and the "upside" created by more parallel search attempts with *random outcomes*.

> *Hypothesis 2 ("PARALLEL SEARCH EFFECT") The maximum or best innovation performance within a group of direct competitors will respond more positively to an increase in numbers of direct competitors than will individual performance.*

2.3. Bounded Rationality and the Search for Solutions

So, are the economics of parallel search really *just* about generating "more variation" in innovation outcomes by adding more competitors? If this were the case, there would be no reason to look further than economic models of competition and incentives for answers. Such analyses might simply pay closer attention to how stochastic outcomes affect maximum performance levels.

Research on behavioral approaches to problemistic search (e.g. March and Simon 1958; Cyert and March 1963) suggests a richer set of relationships. Key to this work is the underlying assumption of *bounded rationality*, or limits in the ability of agents to know, calculate, formulate and predict outcomes--and thereby take fully informed, rational decisions (Simon 1955). These notions are allied to those of earlier-mentioned fundamental uncertainty (in the environment), which effectively imposes bounded rationality (on the decision-maker).

A first important contribution of this literature is to clarify *when* agents' behaviors should be expected to be boundedly-rational. Like the earlier-mentioned research on parallel search, this tradition ultimately points to the environment and, particularly, the *nature of the problem being solved* as the key determinant of boundedly-rational search behavior. Problems that are *complex* or involve a large number of interdependent knowledge sets to solve a problem (Simon 1969) should lead to boundedly-rational behavior in problem-solvers (innovators).[4] Increasing numbers of knowledge sets are inherent in solving a more complex problem; fewer knowledge sets are inherent in solving a less complex or simpler problem (Simon 1969; Jonassen 2004; Nickerson and Zenger 2004; Macher 2006). Recent formal and simulation-based abstractions of problem-solving, as *N* separate decisions with *K* interactions, or "N-K models" (Levinthal 1997; Gavetti and Levinthal 2000; Rivkin 2000; Rivkin and Siggelkow 2003), provide a formal interpretation of the limits of rational calculation of agents. Rivkin (2000) argues that attempts to solve problems necessarily become boundedly-rational at the point at which *N* and *K* become sufficiently large that a problem becomes "NP complete," a theoretical distinction of insolubility borrowed from computer science.[5]

Apart from pointing to the nature of the problem as a determinant of boundedly-rational search, this literature also provides some indication of the process of search itself. The N-K formulation analogizes search as an attempt to find a "peak" solution across a poorly understood "landscape" (Levinthal 1997). Increased complexity leads to increased "ruggedness," or greater discontinuities and more numerous peaks that might be "climbed" (a simple problem with few interactions has a single peak). Typically, the models assume agents are unable to map the link

[4] Simon's (1973) closely related theorizing on the *ill-structuredness* of problems places particular emphasis on how the linkages between necessary knowledge sets in a problem might be poorly understood.

[5] Rivkin's article focuses particularly on the problem of *imitating* a complex solution (organizational strategy).

between alternative solution approaches and performance outcomes (i.e., they have no prior knowledge of the landscape). Phenomenological characterizations of innovation lend credence to this analogy by stressing the search for alternative design paths or approaches (Abernathy and Utterback 1978; Christensen 1997), design architectures (Clark 1985; Henderson and Clark 1990) or new technological combinations (Fleming 2001)—akin to alternative "hills to climb." Within a given approach, innovation might proceed more incrementally along a given technological trajectory (Dosi 1982; Nelson and Winter 1982; Sahal 1985)—akin to "climbing a particular hill."

2.4. The Mediating Effect of Problem Type

If bounded-rationality emerges when a problem is complex, the solution tends to implicate multiple knowledge sets and the process of innovation is largely about searching for the best approach rather than just exerting high effort, *how should this matter*?[6] In principle, bounded rationality might imply any number of things are not wholly known by competing, problem-solving agents. The emphasis of the N-K literature on agents' inability to map solution approaches to performance perhaps most obviously points to unknowable technical performance resulting from different approaches and, ultimately, the value generated from innovation. It might not even be known whether there exists *any* feasible approach at all or what alternative approaches might be.

Regarding the effect of adding greater numbers of competitors, Terwiesch and Xu (forthcoming) provide a useful analytical starting point. They analyze a case they call "trial and error" innovations that links closely to the bounded rational search processes described in Section 2.3 in the sense that the appropriate solution approach is unknown. This is the very same intuition offered by Jewkes et al. The authors make the analytical leap of recognizing that insofar as it is difficult to rank order alternative solution approaches *ex ante*, so too should it be difficult to rank order the capabilities and orientations of competing agents *ex ante*. They model the case of

[6] See Lenox et al. (2006) for an N-K model that includes competition, but without endogenous innovation.

complete uncertainty wherein firms take decisions with the expectation that the value of their "expertise" (likelihood of winning) is the very same as that of competitors. In a sense, agents might be thought of as being distributed on a search landscape at different initial positions without prior knowledge of whether they are positioned next to a tall or a short peak.

Problem complexity thus effectively adds another source of "surprise" to innovation outcomes related to how valuable one's own capabilities and problem solving approach turns out to be in relation to those of other competitors. Adding another source of uncertainty to outcomes should increase the benefits of parallel search.

> *Hypothesis 3 ("COMPLEXITY AND PARALLEL SEARCH") Greater complexity of the problem being solved will lead to a more positive response of the maximum innovation performance to increasing numbers of competitors (i.e. a stronger parallel search effect).*

We argue that this added uncertainty regarding agents' capabilities might also have the potential to shape the competition effect. Uncertainty in the appropriateness of skills and approaches virtually "levels the playing field"; competitors may become closer in terms of their individual chances of finding the best solution. However, the usual intuition that "closer" competitors should engender tighter competition and closer substitutes is clearly incorrect in this case. Although competitors might be closer in *ex ante* likelihood of success when solving a complex problem, only a subset of competitors (that have chosen the same best approach, same hill to climb) are relevant competitors *ex post*. Therefore, the response to an added competitor should be less sensitive.

> *Hypothesis 4 ("COMPLEXITY AND COMPETITION EFFECT") Greater complexity in the problem being solved leads to a less negative individual response to increasing numbers of competitors (i.e. a weaker competition effect).*

The remainder of the paper is devoted to empirically testing these hypotheses.

3. Empirical Context & Data

3.1. Software Development Contests at TopCoder

An ideal data set is one in which both individual and group-level problem-solving outcomes are observed. Further, variation in competitors and problem complexity should allow key relationships to be econometrically identified. We used data on software developer competitions at TopCoder as a context that conforms well to these requirements. Between 2001 and 2007 TopCoder conducted 350 contests that included 1,050 problems and 22,544 programmers.

TC's creates outsourced software for Fortune 1000 firms by encouraging coders from around the world to compete in ongoing programming contests. Clients benefit by having their internal software programming problems resolved and getting access to potential recruits; participants benefit from the opportunity to win prize money and signal their talent in a global competition. TC's widely publicized (on its Web site and in interviews and marketing materials) proposition to its clients is that it can harness the value of a large number of number of programmers working in parallel and let competition determine the best solution. Thus, TC emphasizes a parallel search notion of competition, and its contests are intended to act as selection mechanisms for best solutions. TC works with clients to identify software module requirements it then converts into contests for members. The contests typically run for several weeks and target specific programming tasks like design, development, assembly and testing. Winners are awarded pre-announced cash prizes for their contributions.

Essential to TC's success is a stable of programmers willing to participate in the firm-sponsored module development contests, and the ability to convince client firms that this community of programmers possesses the skill and ability to generate code that delivers results. Ongoing programmer recruitment and skill assessment at TC is done through weekly online "algorithm programming contests" in which participants compete against each other for 75 minutes to solve three software development problems. TopCoder then automatically tests solutions to provide a score and ranking in terms of the design criteria laid out in the original problem statement and robustness tests. It is these regular algorithm contests that are the focus of our empirical analysis.

"Algorithm" Problems: The presence of multitudes of test cases indicates that the problems do not have just "Right" or "Wrong" answers, but instead are of the class of NP Complete problems (like the traveling salesman), for which there is no "right" answer and solutions can approach a theoretical maximum. Although participants have a sense of problem difficulty from the potential point value of the problem, they do not have any prior indication of its inherent nature (or, rather, its solution) or the number of knowledge domains that might be implicated in its solution. However, TC employees, when designing the problems for the algorithmic challenges, do internally classify problems into the sixteen knowledge categories listed in Table 1. Thus, a problem that is internally categorized as graph theory will have been deemed to draw on knowledge in that domain for its solution. Roughly half of the TC problems, however, are associated with multiple categories, and their solutions will thus necessarily implicate multiple knowledge sets.

Consistent with earlier theory on complex problems (Section 2.3), TC executives and problem designers noted that problems associated with multiple categories could not simply be thought of as "adding" two sorts of problem together; because they often involved "tricks" and "greater creativity," participants could not rely on rote solution approaches.[7] Importantly, this notion of problems that drew on multiple knowledge domains, lead to fresh sort of problem was distinct from the pure "difficulty" of a problem, which was assessed separately via a subjective score from 1 to 3 by problem designers. (As seen in Tables 2 and 4, the difficulty and complexity of problems were mostly uncorrelated.) Thus, in our analysis we distinguish single category

[7] In a relevant anecdote, TC executives noted that prior problem solutions and even attempt to use "stock" code when solving single-category problems that these techniques of gaining an edge were far less useful in the case of multi-category problems.

problems as "simple" and those involving multiple knowledge categories as "complex," consistent with theorizing about the knowledge domains and complexity in Section 2.3.

Table 1 TopCoder's Breakdown of Problem Categories

The algorithm contest consists of two distinct phases, programming and solution testing. In the programming phase, participants write and submit code for each of the problems. Each problem is assigned a set amount of points that is visible at the start of the contest. The higher the number of points, the more difficult the problem is. As soon as a participant opens the problem, that is, gets the full problem statement (see Appendix B for sample problem statements), 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 the participants open all three problems at the same time, all three problems will have the total number of points declining.

In the testing phase, final scores for each participant are determined by automatically compiling the software code for each problem and subjecting the code to a barrage of automated test cases ranging from hundreds to thousands to determine the accuracy of the solution over a range of potential conditions. Performance over all the test cases is then summed and the time that was taken to submit the answer converted into an objective final public score and ranking of each participant's algorithm code-writing skills.

Assignment to "Rooms" of Direct Competitors: Competitors are assigned to distinct subgroups of direct competitors in each contest. The typically hundreds of entrants in any given contest are divided into groups called "rooms" of not more than 20 competitors. 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 and public recognition for winning. The top two competitors in each room receive prizes, regardless of differences in scores across rooms. Competitors within individual 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, 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. As there are 20 or fewer competitors in a room, this information is easily navigable. This "heads up" display also reveals who has submitted solutions to enable the progression of the contest to be observed in real-time. Participants are able to observe the submission of solutions by competitors, giving them an idea if they are ahead or behind in the competition. An open chat channel is also available. TC reports that this is most often used by competitors to "trash talk" one another during the competition.

Interviews with TC executives and employees revealed that in this set-up a great deal of competitive rivalry drives much of the interaction among the participants; the weekly contests provide an opportunity for them to demonstrate their skills to each other and improve their rankings. The near-instantaneous scoring of performance and public ranking causes participants to try their best at the various problems. The public nature of the ratings, and the fact that many algorithm contests are sponsored by firms like Google and Microsoft, also causes programmers to exert effort and showcase their skills. There are also the cash prizes. The total cash prize awarded in any particular contest has fluctuated from zero to $21,900, and has averaged $2,700/contest over the course of the six years; 65% (227) of the contests have not involved a cash award. As prizes are divided among different subsets of direct competitors, there might typically be on the order of one to two dozen winners among several hundred entrants.

Weekly algorithm contests are held at a different time and day of the week to accommodate TC's global programming members. Contest dates and times are advertised well in advance to all registered members of TC through a personalized email and on the company's Web site. On the day of a contest, members are given a three-hour window to register their intention to compete. Five minutes before the start of the contest, registration is closed and participants are assigned to the online virtual rooms, or subgroups, of direct competitors.

Competitors are divided into two divisions, I and II, based on prior skills rating in algorithm contests. Division I consists of participants who rank above a pre-determined rating score, Division II of newcomers (i.e., those who do not yet have skill ratings) and those who rank below the Division I threshold score. Room assignment in a division attempts to randomize participants based on the following rules. Coders are first sorted by prior algorithm contest rating. This rating is then transformed into a score by dividing by 1024 and squaring the result. A "search value" is then computed by dividing the score by the sum of all scores for all participants, and then adding to the result the search value of the previous participant. A random floating-point number between 0 and 1 is then chosen. The coder whose search value is closest to this is assigned to the current room. This process repeats itself until a room is deemed full (up to a maximum of 20 competitors), at which point new rooms are opened. The structure of the contests described in this section is further clarified in the following figure.

Figure 1 Illustration of the Structure of TopCoder Competitions

3.2. Data and Sample

Our analysis focuses on the elite Division I algorithm coding contests in which all competitors have skill ratings and are serious competitors.[8] We dropped observations for rooms in which there were fewer than 15 competitors, leaving just rooms with 15 to 20 competitors (average: 16.8). Although there are rooms with as few as 10 competitors, rooms with fewer than 15 competitors represented less than 1% of the total sample, and were eliminated to assure that they did not unduly influence the analysis. (Doing so did not affect results.)

[8] Histogram plots of ratings suggest that the two divisions represent qualitatively different distributions of competitors.

In the sample, 645 problems were solved by a total of 9,627 rooms of competitors, implying, on average, just fewer than 15 rooms simultaneously competing on each problem on average. The principle unit of analysis in the econometric analysis to follow will be a given problem solved by a given room; however, several regressions also study the 162,207 attempts by individuals to solve individual problems. Because the 645 problems implicated different numbers of problem and knowledge categories, problems had different levels of "complexity" (distinct from problem difficulty) as discussed in Section 2. The set of problems in the sample is summarized in Table 2.

Table 2 Sample Problems (and Rooms Distributed across Problems)
by Complexity and Difficulty

Having access to the full database of the TopCoder platform, we were able to draw on measures that closely link to the earlier theoretical development. This will be further elaborated in explaining the econometric modeling approach, below. Table 3 lists and defines the variables used in the analysis. Table 4 provides means, standard deviations and correlations of our key variables. As summarized in these tables, we have measures of problem-solving performance (*AVGSCORE* and *MAXSCORE*), contest-level variables (*MONEYPRIZE*, *TIMEOFDAY*), individual room-level variables (*COMPETITORS, AVGRATING*), and problem-level variables (*COMPLEXITY, DIFFICULTY*, *MAXPOINTS*).

Table 3 Variable Definitions

Table 4 Descriptive Statistics and Correlations

4. Empirical Approach

The central objective of the econometric analysis is to measure how varying the numbers of competitors affected problem-solving performance in the groups of direct competitors in the different rooms of TopCoder contests. We further measure how the complexity of problems being solved by competitors mediated this relationship. Thus, we will test the hypotheses developed in

Section 2. Two distinct approaches to econometric estimation will be used in the analysis to follow. These are explained in turn.

4.1. Exploiting Random Assignment to Problems and Competitor Groups

In our first, and preferred, approach we estimate the causal effect of varying the numbers of competitors on problem-solving performance by exploiting features of the empirical context that are akin to a randomized experiment. Here we have "rooms" of up to 20 competitors who competed against one another to solve problems. The assignment of competitors to rooms is effectively the equivalent of having competitors receiving different "treatments" of competition. The problem is only unveiled during the competition, so competitors are also effectively treated to unforeseen different problem types. There is thus something of an experimental character to this context that might be exploited to estimate the causal effect of varying competition.

However, the context is not a purely randomized experiment; several sources of variation need to be controlled to eliminate potential omitted variable bias. For example, the roster of software developers who have signed up to TopCoder, representing the pool of prospective entrants to a given competition, grows steadily over time. Average competitor quality could plausibly also change over time, as could problem design or grading. If, at the same time, numbers of competitors assigned to rooms systematically varied across competitions, spurious correlation could lead to biased estimates of coefficients of interest.

Although we will explore specifications with round-level and problem-level covariates, our preferred specification simply annihilates all these potential sources of between-round and between-problem variation with individual problem-level fixed effects, η.[9] Thus, the preferred model exploits just the variation across different rooms competing on the same problem to identify how differences in these rooms cause systematic differences in problem-solving

[9] Including fixed effects for individual problems effectively controls for individual weekly contests, as there are three problems per individual contest.

performance. It is this *across-room variation for a given problem* that best approximates an experimental set-up with random assignment.

But interviews with TopCoder executives suggested that variation in numbers of competitors per room might not have been entirely random. Two potential sources of non-random assignment were identified. First, TopCoder's room assignment algorithm attempts to somewhat randomly assign competitors to rooms, while attempting to group competitors somewhat by skill level. If there is any sort of systematic relationship between the average skill (i.e., *AVGRATING*) in a room and the extent to which it gets filled, spurious correlation would result. For example, if top-skill rooms tended to be filled first and fullest (i.e., closer to 20 contestants), this would lead to a positive bias in the estimated effect on problem-solving outcomes of adding competitors.

A second potential source of non-random variation in numbers of competitors could be related to competitors "signing up without showing up." If no-shows are random, this is, in fact, helpful variation that can be exploited to identify relationships of interest. If, however, dropouts tend to be weaker, less-serious competitors, fewer competitors could correspond to higher average scores, creating a negative bias.

The convenient feature of these plausible sources of endogeneity bias is that both relate to the skill levels of competitors in a given room. Therefore, adding an explicit control for the average rating of competitors in a room, *AVGRATING*, should address this bias while increasing the precision of the estimates.

Thus, the preferred model to measure the effect of *COMPETITORS* on problem-solving performance, *Y*, as mediated by our measure of complexity, *COMPLEXITY*, is as follows:

$$Y_{ij} = G(\, COMPETITORS_i \mid COMPLEXITY_j \,) + \eta_j + \delta AVGRATING_i + \varepsilon_{ij} \qquad (1)$$

In this expression, *i* indexes each room of direct competitors and *j* indexes each problem. The term ε_{ij} is a stochastic, zero-mean error term assumed to be normally distributed. The key relationship of interest to be identified is *G*(). The empirical analysis to follow will show that *G()*

is adequately modeled as a simple linear relationship. In the analysis, the interaction with *COMPLEXITY* is captured with both an interaction term and through separate regressions on stratified data.

Problem-solving performance, *Y*, is modeled in two ways. First we model the *average* score achieved by competitors within a room, *AVGSCORE*. Doing so is a means of capturing how competitors respond to *COMPETITORS*, on average. Looking at these systematic differences in mean performance outcomes most directly relates to measuring the competition effect (Section 2.1). Performance is also modeled as the maximum score achieved in a group of direct competitors in a room, *MAXSCORE*, which most directly relates to the parallel search effect (Section 2.2).[10]

4.2. Instrumental Variable Approach

As the earlier approach requires *a priori* assumptions concerning the sources of endogeneity bias, we use a secondary, instrumental variables approach to assure the robustness of results from the earlier approach. We use variation in the time of the day at which contests begin as a source of exogenous variation in *COMPETITORS*. Executive interviews revealed that starting times varied from week to week based on a range of factors. Start time might be 10:13 am one week and 10:47 am the next. Visual inspection of the data showed start times spread from early morning until the early afternoon and a tight cluster of start times after 19:00 until roughly 22:00, eastern standard time. The visual inspection suggested an increasing number of competitors in rooms with later start times, but a clear drop off in the contests beginning later in the day. Therefore the effect of start time on *COMPETITORS* was modeled as a linear model of *TIMEOFDAY* and a dummy variable to account for start times after 3 pm, *AFTERNOON*.

[10] Note that this approach of using problem fixed effects rules out the measurement of direct effects of problem type on innovation performance. However, our earlier hypotheses place the focus on how problem type mediates the relationship between competition and innovation outcomes—interaction effects rather than direct effects.

To account for the possibility that time of day did not relate just to *numbers* of competitors, but perhaps also to the *composition* of competitors (e.g., Russian or Asian competitors might be less likely to enter contests late in the day), the analysis was performed on individual competitor-level data to control for the individual skills of competitors with individual competitor fixed effects, γ.

This approach has the advantage of directly controlling for competitor characteristics when measuring the effect of variation in *COMPETITORS*. But because this estimate is based on variation between rounds, it does not take advantage of the attractive pseudo-experimental variation across rooms. For this reason, the model in this case also includes a vector of contest and problem controls, *X*, for the contest and problem. (If the fixed effects of the earlier approach were to be used, they would annihilate all the identifying variation across contests.) In assessing the individual responses, the approach also can only be used to assess the competition effect and compare the estimate to the model of *AVGSCORE* obtained in the earlier approach.

Thus, the instrumental variables specification measures the effect of *COMPETITORS* on problem solving performance of each individual competitor, *SCORE*, as mediated by *COMPLEXITY*, with an expression that modifies expression (1) in the following manner:

$$SCORE_{ijkl} = G(\, COMPETITORS_i \mid COMPLEXITY_j\,) + \beta \cdot X_{ijk} + \gamma_l + \delta AVGRATING_i + \varepsilon_{ijkl} \quad (2)$$

In this expression, *i* indexes each room of direct competitors, *j* indexes each problem, *k* indexes the contest and *l* indexes the individual competitor.

5. Analysis & Results

5.1. Number of Competitors and Problem Solving Performance

In this section, we examine the relationship between competition and problem solving in order to establish a baseline relationship between them that provides evidence of both a competition effect and parallel search effect. We then examine the mediating effect of the nature of the problem, which makes the section relevant to *Hypotheses 1* and *2*. Table 5 presents results.

We begin by estimating how the average score achieved in a room, *AVGSCORE*, changed in response to changing numbers of competitors, *COMPETITORS*. Properly estimated, the relationship should reflect how changes in *COMPETITORS* systematically affected individual level performance, on average. Focusing on competitors' average responses to competition has direct relevance to *Hypothesis 1*, which predicts that higher numbers of competitors will result in lower individual performance. We present several specifications so as to explicitly document ways in which the model deals with endogeneity bias as well as the robustness of the approach.

Model (5-1) presents the regression of *AVGSCORE* on problem covariates (*DIFFICULTY*,[11] *MAXPOINTS*, and dummy variables for the type of problem) and round covariate (*MONEYPRIZE*).[12] Coefficients on covariates appear sensible. Given that more difficult problems are worth many more points, it is not surprising to see large positive coefficients on indicator variables for increasing levels of *DIFFICULTY*. The coefficient on *MAXPOINTS* is negative. This might at first seem surprising, but given strong collinearity with *DIFFICULTY*, and the strong explanatory power of this variable, it appears that *MAXPOINTS* is simply picking up a second order effect. (Re-regressing the model without *MAXPOINTS* confirms this.) This multicollinearity should not bias estimates of other coefficients, in any case. The coefficient on *MONEYPRIZE* is positive, as should be expected. The estimated coefficient on the explanatory variable of greatest interest, *COMPETITORS*, is negative, with a magnitude of -11.5, and is highly statistically significant (all estimates are based on robust standard errors).

Table 5 Baseline Estimates of the Relationship between Competition and Problem-Solving

Model (5-2) introduces fixed effects for individual problems. Annihilating variation across problems (and across week-to-week contests at the same time) leads contest and problem

[11] Because the dummy variable for *DIFFICULTY* = 1 drops out, the other dummy variables should be interpreted as differences with this level.

[12] The F-test for overall model fit is significant at $p = 1\%$ for all models.

covariates to drop out. The estimated coefficient on *COMPETITORS* remains negative and highly significant, though just slightly smaller in magnitude at -8.9.

Having included problem-level fixed effects, coefficients are now estimated only on the basis of room-to-room variation *for a given problem*. As discussed earlier, any remaining endogeneity bias should relate to the skills of the players in a given room (Section 4.1). Model (5-3) therefore introduces *AVGRATING* as an explicit control. Doing so again leads to a negative and highly significant coefficient on *COMPETITORS*, but of the slightly smaller magnitude of -5.0. To assure that this linear control sufficiently eliminates any bias associated with the skills of competitors in a room, model (5-4) allows the effect of *AVGRATING* to take a very flexible shape, with a series of dummy variables that divide its effect into 20 levels that correspond to the 5th, 10th, 15th, etc. percentile levels of this variable. The coefficient on *COMPETITORS* is statistically unchanged, affirming model (5-3).

To confirm the appropriateness of the linear model, various parametric and non-parametric specifications were assessed. Figure 2 graphically compares the linear model (5-3) with a very flexible specification that estimates the effect of increasing numbers of competitors with independent dummy variatles to to illustrate the good fit of the linear model.

Figure 2 Linear and Non-Linear Models of Average Responses to Changing Numbers of Competitors

We now re-estimate average performance responses to *COMPETITORS* using the instrumental variables approach (Section 4.2) to ensure that unexpected sources of bias are not playing a role. This approach effectively measures how competitor scores systematically changed from week to week in response to the different numbers of competitors they faced in different contests. Problem and round covariates are re-included in the model.[13] Model (5-5) presents the estimated “first stage” relationship between *COMPETITORS* and *TIMEOFDAY* and the indicator variable

[13] Problem covariates are averaged across the three problems in each round.

AFTERNOON. The results suggest that holding a contest in the afternoon, on average, led to one more competitor in each room. Apart from this effect, for each hour advance of the clock from when a round started, there were about one-tenth more competitors per room, on average.

Model (5-6) reports the full IV model. The IV regression, using an entirely different source of variation, yields essentially the same results as the earlier approach. Coefficients on control variables *DIFFICULTY*, *MAXPOINTS* and *MONEYPRIZE* take identical signs and similar values as those in model (5-1). The coefficient on COMPETITION is -6.97, which is slightly more negative than the earlier room-level estimate of -5.0 in model (5-3), but well within a standard error of 2.59.

The analysis thus provides strong support for the econometric approach and confirms that for every additional competitor, the average effect on competitors was to reduce performance by roughly 5 points, consistent with *Hypothesis 1* and the presence of a competition effect.[14]

We now turn to the question of whether the maximum score attained in group of direct competitors, *MAXSCORE*, was also affected by variation in *COMPETITORS*. *Hypothesis 2* predicts that *MAXSCORE* will respond more positively to adding competitors than would be predicted by the competitor effect on its own.

Model (5-7) effectively repeats the earlier preferred model (5-3), simply changing the dependent variable to *MAXSCORE*. The coefficient found on MAXSCORE is -.24 and statistically indistinguishable from zero. The result that competition has no effect on the best score in a group, on average, is itself remarkable. The result is also consistent with *Hypothesis 2*: the response of the maximum performance is, indeed, more positive than what would be predicted by the competition effect on its own (i.e., -.24 is larger than -5). The zero net effect suggests that

[14] Corroborating this finding of changing *behavior* in response to changing numbers of competitors, TopCoder executives indicated competitors “try less” and sometimes “give up” when they face high competition. We performed ancillary analyses that confirmed this point: competitors spent less time working on solutions, on average, when there were more competitors.

despite a systematically negative effect on individual competitors of increasing competition, the upside variation or parallel search effects weighed against this negative effect.

Important to note is that, in principle, the explanation for the less negative (zero) slope could have been that "top" competitors were simply more likely to have been more stimulated (or less negatively affected) by competition (cf. Aghion et al. 2005). However, re-running model (1-1) on just the subsample of those "top" competitors with high pre-competition ratings (various levels were assessed) finds that these top competitors respond even more negatively than average firms do. This is consistent with top competitors acting more strategically. Thus, the possibility of a more positive response by top competitors can be ruled out.

It should also be noted that the zero coefficient might, in principle, also simply reflect a null and meaningless result. But results presented in the following section clarify that this zero average effect is, in fact, an average of significantly positive and significantly negative effects in subsets of the data. Therefore, the result is meaningful and consistent with *Hypothesis 2*.

5.2. The Nature of the Problem Being Solved by Competitors

The preceding analysis provided evidence suggesting that both competition and parallel search effects are at work in the data. Here we explore how the complexity of the problem, *COMPLEXITY*, might mediate these effects. The following analysis therefore relates to *Hypotheses 3* and *4*, which essentially predict that complexity leads to less negative effects of competition on both the average and maximum scores. Results are presented in Table 6.

Table 6 Estimates of the Effect of Problem Type on the Competition-Problem-Solving Relationship

We begin our analysis of the mediating effect of problem type on the link between adding competitors and problem-solving outcomes by including an interaction term, *COMPETITORS ×*

COMPLEXITY.[15] The coefficient on this interaction term should indicate the average effect of implicating one more knowledge set in a problem on the link between competition and problem-solving. Consistent with *Hypothesis 3*, the coefficient on the interaction term in a model of *AVGSCORE*, model (6-1), takes a positive and significant sign, which indicates that complex problems have a less negative response to competition. As in Figure 3, plotting these results shows that the estimated effect of adding another competitor goes to zero at *COMPLEXITY* = 3. Remarkably, this implies that for the more complex problems, competition had no net effect on individual behaviors, on average. (The coefficient estimates would imply that the effect, in fact, turns positive at *COMPLEXITY* = 4, 5. Although this is theoretically possible, as noted in Section 2.1, we do not emphasize this point given that 98% of the sample data are for *COMPLEXITY* between 1 and 3.)

Based on the coefficient on *COMPETITORS* and the interaction term, the slope in the case of simple problems (i.e., *COMPLEXITY* = 1) should be -8.60, that is, the sum of direct and interaction effects, or -13.06 + 4.46. This is a more negative effect than the earlier overall estimate of -5 in model (5-3), as should be expected. To provide an indication of the robustness of the estimate, we re-estimate the effect on stratified subsamples of simple (*COMPLEXITY* = 1) and complex (COMPLEXITY > 1) problems. This effectively allows all model coefficients to flexibly vary across sample subsets, rather than just the effect of competition, as in the interaction model. As reported in model (6-3), the coefficient on *COMPETITORS* for simple problems is -8.56, which is extremely close to the earlier estimate based on the interaction model. The coefficient in the case of complex problems is -2.64, again suggesting a less negative effect of competition for these sorts of problems. Further stratifying complex problems into subsets of *COMPLEXITY* = 2 and *COMPLEXITY* = 3, presented graphically in Figure 3, shows that the

[15] The interaction term does not require *COMPLEXITY* to also be added, as the fixed effect for problems already soaks up any problem-specific variation.

linear interaction model is a good approximation of the decreasingly negative effect.[16] (Stratified analyses at *COMPLEXITY* = 4, 5 are not included given few degrees of freedom, as seen in Table 2.)

We then turn to the same sort of analysis of the mediating effect of problem complexity, but in relation to *MAXSCORE*. Consistent with Hypothesis 4, the coefficient on the interaction term in a model of *MAXSCORE*, model (6-2), also takes a positive and significant sign and one that is larger than the interaction term on *AVGSCORE* in model (6-1). This is important, as the difference between these two interaction terms approximates how the parallel path effect is mediated by problem complexity above and beyond the mediating effect on individual competitors' incentives.[17]

Again, we assess the robustness of the result by re-estimating the effect on stratified subsamples of simple (*COMPLEXITY* = 1) and complex (COMPLEXITY > 1) problems. As reported in model (6-4), the coefficient in simple problems on *COMPETITORS* is -6.27, statistically indistinguishable from the -4.46 that is implied in the interaction model at *COMPLEXITY* = 1. Adding another competitor to a complex problem, in fact, has an absolutely positive effect, with a coefficient of 3.64, as presented in model (6-6). (The lower statistical significance of the coefficient is not surprising given that there is considerably more noise in the maximum score, as is readily seen in the descriptive statistics in Table 4.) This substantial, positive impact is consistent not just with the hypotheses developed in Section (2.2), but also with the broader spirit of work that argues for the virtues of parallel paths (Nelson 1959; Cohen and Malerba 2001; Leiponen and Helfat 2007; Terwiesch and Xu Forthcoming). It is also consistent with TopCoder's enthusiasm for utilizing competition as a means of surfacing best solutions

[16] We performed ancillary analyses that further corroborated the view that complexity creates greater uncertainty with regard to who will win: we find that winners of complex problems were lower ranked, on average, (in terms of their pre-competition skill ratings) than were winners of simple problems.

[17] This assumes that the effect of *COMPETITORS* on *AVGSCORE* (i.e. the average competition effect on individual behavior) approximates how competition affected winning competitors. In fact, this is likely a conservative assumption given that, as noted earlier, top competitors tended to respond more negatively to competition.

(Section 3). Further stratifying complex problems into subsets of *COMPLEXITY* = 2 and *COMPLEXITY* = 3 provides evidence consistent with the linear approximation of the interaction term, although less obviously so. Figure 3 shows that stratified estimates of the effect of adding a competitor on MAXSCORE approximate the linear model for *COMPLEXITY* = 1, 2. However, the estimate for *COMPLEXITY* = 3 is -.16, well below the linear model. This appears to be at least partly the result of much greater noise in the MAXSCORE variable; the standard error on this estimate is 4.48. Therefore, the estimate is statistically in the range of the estimate in the linear interaction model. To better assess this possibility, we add the data for higher levels of complexity, *COMPLEXITY* ≥ 3 rather than just equal to 3. The point estimate in this case is 3.87. Although the estimate has a similarly large standard error, the point estimate is consistent with an effect on *MAXSCORE* that increases with *COMPLEXITY*, as 3.87 exceeds estimates of the effect of adding competitors at *COMPLEXITY* = 2 or 1.

The dominant curves presented in Figure 3 provide an overall view of the responses of *MAXSCORE*, model (6-2), and AVGSCORE, model (6-1), to added competitors, as mediated by problem complexity. This figure summarizes points consistent with the four hypotheses: the effect of adding another competitor on *AVGSCORE* is negative (*Hypothesis 1*); the effect of adding another competitor on *MAXSCORE* is less negative than this (*Hypothesis 2*); problem complexity leads to a less negative response of *AVGSCORE* to competition (Hypothesis 3); problem complexity leads to an even less negative response of *MAXSCORE* (*Hypothesis 4*).

Figure 3 Estimated Effect of Adding an Additional Competitor to Average Scores and the Winning Score in a Group of Direct Competitors in a Room[18]

6. Conclusions

In this paper, we empirically studied the relationship between competition and innovation using data on elite software programmer competitions. The strength of the analysis rests largely on our

[18] See the discussion in Section 5.2 to interpret this graph, particularly the stratified sample estimates related to *MAXSCORE*. Results are plotted for just *COMPLEXITY* ≤ 3, as this constitutes 98% of sample data, as shown in Table 2.

ability to observe key microeconomic variables and exploit random assignment in this context. The analysis shows that increasing numbers of competitors had a negative effect on individual competitors' performance. However, the effect on the best individual score, or maximum performance, within a group of direct competitors was much less negatively (more positively) affected by adding competitors. The effect on both average performance and maximum performance was found to be substantially less negative (more positive) when the problem being solved was complex, that is, when solution approaches drew from multiple knowledge domains. The effect of adding more competitors on the maximum score was, in fact, absolutely positive for complex problems.

This study contributes empirical evidence relevant to the long-studied question of how competition and innovation are linked. The baseline finding of a reduced average response of innovation to competition affirms one of the most basic arguments in this literature, relating innovation to strategic incentives. The contribution here is simply the strength of our econometric approach in identifying the causal relationship.

A more novel empirical contribution is to demonstrate the differential effect of increasing competition on average versus maximum performance within the same analysis. By distinguishing these effects, this study shows that effects described here as the "competition effect" and the "parallel search effect" coexist and should both be considered when determining optimal ways to organize competing innovation.

Our most important finding was that both the average and maximum response of innovation performance's to increasing numbers of competitors became less negative (more positive) with the increasing complexity of the problem being solved. We interpret this finding to indicate that complex problems change the nature of the innovation search process and the link with competition. The response to problem complexity is consistent with theorizing on boundedly-rational problemistic search (Nelson 1959; Simon and Newell 1962; Levinthal 1997) and, at the same time, consistent with conceptions of innovation as relate to neoclassical investment functions and strategic interactions (e.g. Arrow 1962; Romer 1990; Aghion et al. 2005).

Perhaps most fascinating of all in these results is that the competition effect is substantially mitigated in complex problems, the sorts of problems in which parallel path effects and bounded rationality are also bound to be strongest. This finding suggests that insofar as individual competitors anticipate the nature of the problem being solved, they adjust their responses to added competitors. Therefore, the nature of the problem implicitly changes the innovative process and, ultimately, the structure of competition.

References

Abernathy, W.J., R.S. Rosenbloom. 1969. Parallel Strategies in Development Projects. *Management Science* **15**(10) B486-B505.

Abernathy, W.J., J.M. Utterback. 1978. Patterns of Industrial Innovation. *Technology Review* **79**(June/July) 41-47.

Aghion, P., N. Bloom, R. Blundell, R. Griffith, P. Howitt. 2005. Competition and Innovation: An Inverted-U Relationship. *Quarterly Journal of Economics* **120**(2) 701-728.

Aghion, P., R. Griffith. 2005. *Competition and Growth: Reconciling Theory and Evidence*. MIT Press, Cambridge, MA.

Aghion, P., P. Howitt. 1992. A Model of Growth through Creative Destruction. *Econometrical* **60**(2) 323-351.

Arrow, K.J. 1962. K.J. Arrow, Economic welfare and the allocation of resources for invention. In: R.R. Nelson, Editor, The Rate and Direction of Inventive Activity, Princeton University Press, New Jersey (1962), pp. 609–625.

Augereau, A., S. Greenstein, M. Rysman. 2006. Coordination versus Differentiation in a Standards War: The Adoption of 56K Modems. *Rand Journal of Economics* **37**(4) 887-910.

Bakos, J.Y., E. Brynjolfsson. 1993. Information Technology, Incentives, and the Optimal Number of Suppliers. *Journal of Management Information Systems* **10**(2) 37-53.

Baldwin, C.Y., K.B. Clark. 2000. *Design Rules: Volume 1. The Power of Modularity*. MIT Press, Cambridge, MA.

Bresnahan, T.F., S. Greenstein. 1999. Technological Competition and the Structure of the Computer Industry. *The Journal of Industrial Economics* **47**(1) 1-40.

Che, Y.-K., I. Gale. 2003. Optimal Design of Research Contests. *American Economic Review* **93**(3) 646-671.

Chesbrough, H. 2002. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Press, Boston.

Christensen, C. 1997. Patterns in the Evolution of Product Competition. *European Management Journal* **15**(2) 117-127.

Clark, K.B. 1985. The Interaction of Design Hierarchies and Market Concepts in Technological Evolution. *Research Policy* **14**(5) 235-251.

Cohen, W. 1995. Empirical studies of innovative activity, in: P. Stoneman (Ed.), Handbook of the Economics of Innovation and Technological Change, Blackwell.

Cohen, W., R. Levin. 1989. Empirical Studies of Innovation and Market Structure. R. Sehmalensee, R.D. Willig, eds. *Handbook of Industrial Organization*. Elsevier Science Publishers, 1060-1107.

Cohen, W.M., F. Malerba. 2001. Is the Tendency to Variation a Chief Cause of Progress? *Industrial and Corporate Change* **10** 587-608.

Cyert, R.M., J.G. March. 1963. *Behavioral Theory of the Firm* Englewood Cliffs, N.J.

Dahan, E., H. Mendelson. 2001. An Extreme-Value Model of Concept Testing. *Management Science* **47**(1) 102-116.

Ding, M., J. Eliashberg. 2002. Structuring the New Product Development Pipeline. *Management Science* **48**(3) 343-363.

Dixit, A.K., J.E. Stiglitz. 1977. Monopolistic Competition and Optimum Product Diversity. *American Economic Review* **67**(3) 297-308.

Dosi, G. 1982. Technological Paradigms and Technological Trajectories A Suggested Interpretation of the Determinants and Directions of Technical Change. *Research Policy* **11**(3) 147-162.

Dosi, G., D.A. Levinthal, L. Marengo. 2003. Bridging Contested Terrain: Linking Incentive-Based and Learning Perspectives on Organizational Evolution. *Industrial and Corporate Change* **12**(2) 413-436.

Ellison, G., D. Fudenberg. 2003. Knife-Edge or Plateau: When Do Market Models Tip? *The Quarterly Journal of Economics* **118**(4) 1249-1278.

Fleming, L. 2001. Recombinant Uncertainty in Technological Search. *Management Science* **47**(1, Design and Development) 117-132.

Fullerton, R.L., R.P. McAfee. 1999. Auctioning Entry into Tournaments. *American Economic Review* **107**(3) 573-605.

Gavetti, G., D. Levinthal. 2000. Looking Forward and Looking Backward: Cognitive and Experiential Search. *Administrative Science Quarterly* **45**(1) 113-137.

Gawer, A., M.A. Cusumano. 2002. *Platform Leadership: How Intel, Microsoft and Cisco Drive Industry Innovation*. Harvard Business School Press, Boston.

Gerchak, Y., D.M. Kilgour. 1999. Optimal Parallel Funding of Research and Development. *IIE Transactions* **31** 145-152.

Henderson, R.M., K. Clark. 1990. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly* **35**(1) 9-30.

Hong, L., S.E. Page. 2001. Problem Solving by Heterogeneous Agents. *Journal of Economic Theory* **97**(1) 123-163.

Hong, L., S.E. Page. 2004. Groups of Diverse Problem Solvers Can Outperform Groups of High-Ability Problem Solvers. *PNAS* **101**(46) 16385-16389.

Jewkes, J., D. Sawers, R. Stillerman. 1959. *The Sources of Invention*. St. Martin's Press, New York.

Jonassen, D.H. 2004. *Learning to Solve Problems: An Instructional Design Guide*. Jossey-Bass, San Francisco.

Knight, F.H. 1921. *Risk, Uncertainty and Profit*. Houghton Mifflin Company, Boston.

Leiponen, A., C.E. Helfat. 2007. Innovation Objectives, Knowledge Sources and the Benefits of Breadth. Working Paper (October).

Lenox, M., S. Rockart, and A. Lewin (2006) ““Interdependency, Competition and the Distribution of Firm and Industry Profits.” Management Science 52(5): 757-772.

Levinthal, D.A. 1997. Adaptation on Rugged Landscapes. *Management Science* **43**(7) 934-950.

Macher, J., T. 2006. Technological Development and the Boundaries of the Firm: A Knowledge-Based Examination in Semiconductor Manufacturing. *Management Science* **52**(6) 826-843.

March, J.G., H. Simon. 1958. *Organizations*. Wiley.

Marshall, A. 1961. *Principles of Economics*, 9th ed. The Macmillan Company, New York.

Nelson, R.R. 1959. The Economics of Parallel R and D Efforts: A Sequential-Decision Analysis. *RAND Research Memorandum* **RM2482**.

Nelson, R.R. 1961. Uncertainty, Learning, and the Economics of Parallel Research and Development Eff *The Review of Economics and Statistics* **43**(4) 351-364.

Nelson, R.R., S.G. Winter. 1982. *An Evolutionary Theory of Economic Change*. Belknap Harvard, Cambridge, MA.

Nickell, S. 1996. Competition and Corporate Performance. *Journal of Political Economy* **2**(4) 724-746.

Nickerson, J.A., T.A. Zenger. 2004. A Knowledge-Based Theory of the Firm - The Problem Solving Perspective. *Organization Science* **15**(6) 617-632.

Raith, M. 2003. Competition, Risk and Managerial Incentives. *American Economic Review* **93**(1425-1436).

Rajan, R., L. Zingales. 1998. Power in a Theory of the Firm. *The Quarterly Journal of Economics* **113**(2) 1998.

Rivkin, J.W. 2000. Imitation of Complex Strategies. *Management Science* **46**(6) 824-844.

Rivkin, J.W., N. Siggelkow. 2003. Balancing Search and Stability: Interdependencies among Elements of Organizational Design. *Management Science* **49**(3) 290-311.

Rochet, J.C., J. Tirole. 2006. Two-Sided Markets: A Progress Report. *Rand Journal of Economics* **35**(3) 645-667.

Romer, P.M. 1990. Endogenous Technical Change. *Journal of Political Economy* **98**(5) S71-s102.

Rotemberg, J.J., G. Saloner. 1994. Benefits of Narrow Business Strategies. *American Economic Review* **84**(5) 1330-1349.

Sahal, D. 1985. Technological Guideposts and Innovation Avenues. *Research Policy* **14**(2) 61-82.

Salop, S.C. 1979. Monopolistic Competition with Outside Goods. *Bell Journal of Economics* **10**(1) 141-156.

Simon, H.A. 1947. *Administrative Behavior: A Study of Decision-making Processes in Administrative Organizations*. Macmillan, Chicago, IL.

Simon, H.A. 1955. A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics* **69** 99-118.

Simon, H.A. 1969. *The Sciences of the Artificial*. Massachusetts Institute of Technology, Cambridge.

Simon, H.A., A. Newell. 1962. Computer Simulation of Human Thinking and Problem Solving. *Monographs of the Society for Research in Child Behavior* **27** 137-150.

Taylor, C.R. 1995. Digging for Golden Carrots: An Analysis of Research Tournaments. *American Economic Review* **85**(4) 872-890.

Terwiesch, C., C.H. Loch. 2004. Collaborative Prototyping and the Pricing of Custom-Designed Products. *Management Science* **50**(2) 145-158.

Terwiesch, C., Y. Xu. Forthcoming. Innovation Contests, Open Innovation, and Multiagent Problem Solving. *Management Science* 1-15.

von Hippel, E. 2005. *Democratizing Innovation*. MIT Press, Cambridge, MA.

Figures

Figure 1 Illustration of the Structure of TopCoder Competitions

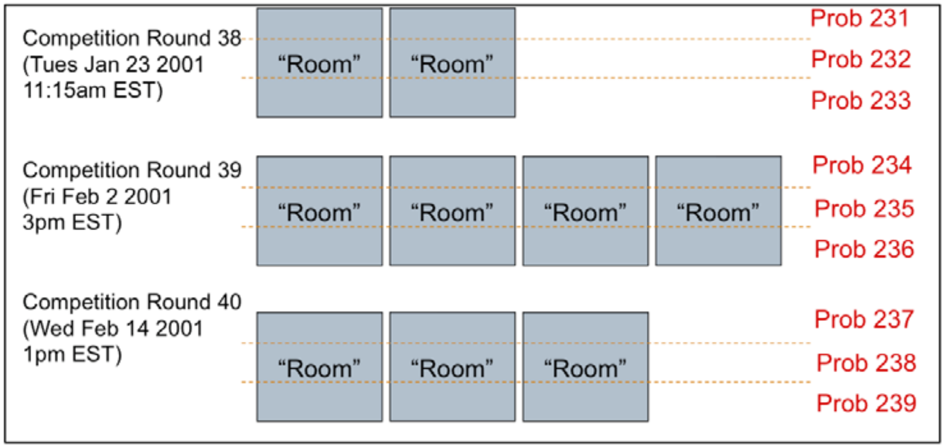


Figure 2 Comparison of Preferred Linear Specification,with Flexible Non-Parametric Specification

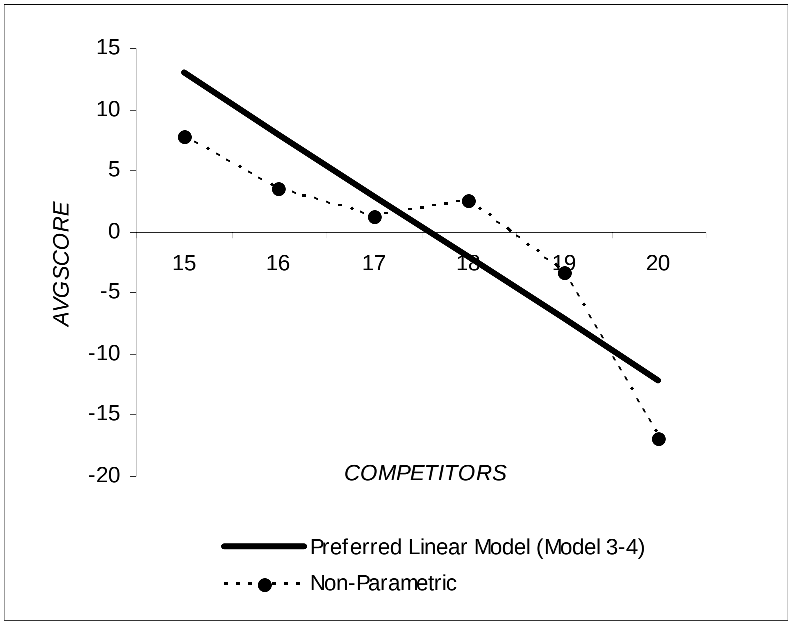
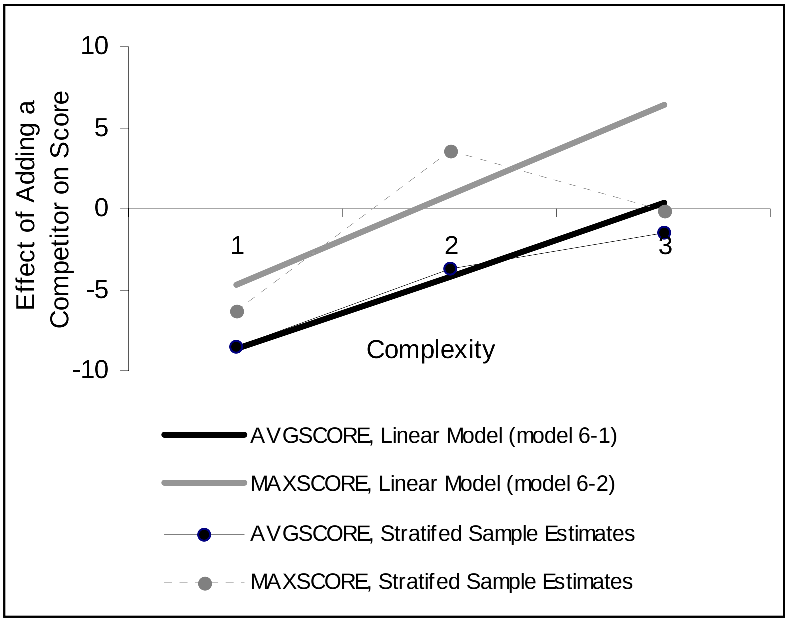


Figure 3 Estimated Effect of Adding an Additional Competitor to Average Scores and the Winning Score in a Group of Direct Competitors in a Room



Tables

Table 1 TopCoder’s Breakdown of Problems Categories

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

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 Sample Problems (and Rooms Distributed across Problems) by Complexity and Difficulty

	Sample Problems				"Rooms" of Direct Competitors			
COMPLEXITY	*DIFFICULTY*				*DIFFICULTY*			
(No. Knowledge Categories)	1	2	3	Total	1	2	3	Total
1	87	92	95	274	1,308	1,233	1,343	3,884
2	99	94	77	270	1,451	1,496	1,205	4,152
3	36	24	34	84	408	390	500	1,298
4	3	4	9	16	42	68	161	271
5	0	1	0	1	0	22	0	22
Total	215	215	215	645	3,209	3,209	3,209	9,627

Table 3 Variable Definitions

Variable	Definition
AVGSCORE	Total number of points scored by a set of direct competitors in a given room, divided by the number of competitors in the room
MAXSCORE	Best or winning score in a group of direct competitors in a room
COMPETITORS	Number of competitors in a given room of direct competitors
COMPLEXITY	Count of different TopCoder knowledge categories that a given problem has been associated with
MONEYPRIZE	Dollar prize awarded per room
DIFFICULTY	TopCoder's assessment of the difficulty of solving a problem as a score of 1, 2 or 3, where 3 is most difficult
MAXPOINTS	Maximum possible points that are possible to score on a given problem
AVGRATING	TopCoder's internal rating of competitor's abilities, based on historical performance, averaged across all direct competitors in a given room
TIMEOFDAY	The precise clock time (hours, minutes) at which a given contest started, expresed in terms of (decimalized) 24 hour clock
AFTERNOON	Indicator variable switched on when TIMEOFDAY > 15 (i.e. 3pm)

Table 4 Descriptive Statistics and Correlations

Variable	Mean	Std. Dev.	Min	Max	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1) *AVGSCORE*	279.4	131.1	0	840							
(2) *MAXSCORE*	313.7	212.9	0	995	.39						
(3) *COMPETITORS*	18.4	1.1	15	20	-.11	-.02					
(4) *COMPLEXITY*	1.8	.8	1	5	-.09	-.06	.03				
(5) *MONEYPRIZE*	38.8	68.1	0	234	.01	.01	.09	.04			
(6) *DIFFICULTY*	2.0	.8	1	3	.00	.20	.00	.05	-.01		
(7) *MAXPOINTS*	585.6	304.7	200	1200	-.01	.15	.00	.05	-.01	.98	
(8) *AVGRATING*	1757.7	244.0	1036	3122	.28	.17	-.06	.01	.04	.18	.19

Table 5 Baseline Estimates of the Relationship between Competition and Problem-Solving

	Room-level, AVGSCORE				Competitor-level, SCORE		Room-level, MAXSCORE
	(5-1)	(5-2)	(5-3)	(5-4)	(5-5)	(5-6)	(5-7)
Explanatory Variables	Covariates	Problem FE	FE, Room Controls I	FE, Room Controls II	First-Stage	IV, FE	FE, Room Controls
COMPETITORS	-11.50***	-8.94***	-5.00***	-5.06***		-6.97***	-.25
	(1.14)	(.84)	(.74)	(.73)		(2.59)	(1.56)
MONEYPRIZE	0.04*				.001***	.04***	
	(.02)				(.00)	(.01)	
DIFFICULTY = 2	40.78***				.03	52.05***	
	(8.73)				(.02)	(2.26)	
DIFFICULTY = 3	109.67***				.06	162.01***	
	(22.54)				(.05)	(6.01)	
MAXPOINTS	-.14***				.00	-.34***	
	(.03)				(.00)	(.01)	
AVGRATING			.17***		-.0003***	-.006	.15***
			(.00)		(.00)	(.00)	(.01)
AVGRATING Dummies				Yes			
TIMEOFDAY					.08***		
					(.00)		
AFTERNOON					.95***		
					(.02)		
Problem Type Dumm	Yes				Yes	Yes	
Round FE							
Problem FE		Yes	Yes	Yes	Yes	Yes	Yes
R-squared	.04	.64	.73	.73	.03		0.55

Notes. *, **, and *** indicates statistical significance at the 10%, 5% and 1% levels, respectively. Robust standard errors in parentheses; In room-level regressions, *N* = 9,627; In individual competitor-level regressions, N = 162,207; these observation counts relate to the sample of 645 problems.

Table 6 Estimates of the Effect of Problem Type on the Competition-Problem-Solving Relationship

	ALL PROBLEMS		SIMPLE PROBLEMS (COMPLEXITY=1)		COMPLEX PROBLEMS (COMPLEXITY>1)	
	AVGSCORE	MAXSCORE	AVGSCORE	MAXSCORE	AVGSCORE	MAXSCORE
Explanatory Variables	(6-1)	(6-2)	(6-3)	(6-4)	(6-5)	(6-6)
COMPETITORS	-13.06***	-10.26**	-8.56***	-6.27**	-2.64***	3.64*
	(1.78)	(4.00)	(1.17)	(2.50)	(.94)	(1.98)
COMPETITION x COMPLEXITY	4.46***	5.54***				
	(.88)	(2.08)				
AVGRATING	.17***	.15***	.18***	.14***	.17***	.16***
	(.00)	(.01)	(.01)	(.01)	(.00)	(.01)
Problem FE	Yes	Yes	Yes	Yes	Yes	Yes
R-squared	0.73	0.55	0.74	0.57	0.72	0.54

Notes. *, **, and *** indicates statistical significance at the 10%, 5% and 1% levels, respectively. Simple problems are those that implicate a single category of problem type, i.e. COMPLEX = 1; Complex problems are those that implicate more than one category of problem type, i.e. COMPLEX > 1; Robust standard errors in parentheses; In room-level regressions, N = 9,627; these observation counts relate to the sample of 645 problems.

Sample TopCoder Problem

Problem Statement

You are working for the FBI and are trying to locate a particular criminal organization. Within the organization you know which members communicate with each other. The problem is that the members may go by aliases. Given both the information in the database about the organization, and the field data about a suspicious organization, you will determine whether they represent the same group. The two sets of data represent the same group if and only if they only differ by the names of the participants.

For example:

```
Database         Field Data
FRANK ----- BOB      WILLARD ----- GEORGE
  |               |
  |               |
  |               |
GEORGE              GREG
```

The Database and Field Data represent the same organization even though the participants are using different names.

```
Renaming Scheme:
FRANK -> WILLARD        FRANK -> WILLARD
BOB -> GEORGE      or   BOB -> GREG
GEORGE -> GREG          GEORGE -> GEORGE
```

If the database data and the field data represent the same organization return how many of the members are going by aliases, otherwise return -1. If there is more than one naming scheme possible in mapping the database information to the field data, use the one that gives the greatest value for the number of aliases. So in the previous example we would use the first renaming scheme thus giving 3 instead of 2.

The database data will be given in a String[] database. Each element of database will be in the form "NAME1 NAME2" meaning that NAME1 and NAME2 communicate with each other. The field data will be given in a String[] fieldData that is formatted in the same way as database. In the above example, the input could have been formatted as:

database = {"FRANK BOB","FRANK GEORGE"}
fieldData = {"WILLARD GREG","GEORGE WILLARD"}

In any particular organization, no two people will have the same name or alias. In other words, no two different people in the database will have the same name in database. In addition, no two different people in the field data will have the same name in fieldData.

Problem Statement

A certain vending machine delves out its goods from a rotating cylinder, which can rotate around in both clockwise and counter-clockwise directions. The cylinder has a number of shelves on it, and each shelf is divided into a number of columns. On the front of the machine, there is a panel of doors that extends the entire height of the column. There is one door for each shelf, which is the width of one column. When a purchase is made, the user uses two buttons to rotate the cylinder so their purchase is located at a door. They make their purchase by sliding the appropriate door open, and removing the item (there can only be one item per column on a particular shelf). The cylinder can rotate in a complete circle, and so there are always two ways to get from a particular column to another column.

Because the vending machine company wants to sell the most expensive items possible, and the machine can only show one column at a time, the machine will always try to put forth the most expensive column available. The price of a column is calculated by adding up all the prices of the remaining items in that column. The most expensive column is defined to be the one with the maximum price. If 5 minutes have elapsed since the last purchase was made, the machine rotates the cylinder to the most expensive column. If, however, another purchase has been made before the 5 minutes are up, the rotation does not occur, and the 5 minute timer is reset.

Recently, some machines' rotating motors have been failing early, and the company wants to see if it is because the machines rotate to show their expensive column too often. To determine this, they have hired you to simulate purchases and see how long the motor is running.

You will be given the prices of all the items in the vending machine in a String[]. Each element of prices will be a single-space separated list of integers, which are the prices (in cents) of the items. The Nth integer in the Mth element of prices represents the price of the Nth column in the Mth shelf in the cylinder. You will also be given a String[] purchases. Each element in purchases will be in the format:

"<shelf>,<column>:<time>"

<shelf> is a 0-based integer which identifies the shelf that the item was purchased from. <column> is a 0-based integer which identifies the column the item was purchased from. <time> is an integer which represents the time, in minutes, since the machine was turned on.

In the simulation, the motor needs to run for 1 second in order to rotate to an adjacent column. When the machine is turned on, column 0 is facing out, and it immediately rotates to the most expensive column, even if the first purchase is at time 0. The machine also rotates to the most expensive column at the end of the simulation, after the last purchase. Note that when an item is purchased, its price is no longer used in calculating the price of the column it is in. When the machine rotates to the most expensive column, or when a user rotates the cylinder, the rotation is in the direction which takes the least amount of time. For example, in a 4-column cylinder, if column 0 is displayed, and the cylinder is rotated to column 3, it can be rotated backwards, which takes 1 second, versus rotating forwards which takes 3 seconds.

If a user tries to purchase an item that was already purchased, this is an incorrect simulation, and your method should return -1. Otherwise, your method should return how long the motor was running, in seconds.

Problem Statement

DNA testing is one of the most popular methods of establishing paternity. In such a test, you compare samples of DNA from the man, the child, and the child's mother. For the purposes of this problem, assume that each sample is represented as a String of uppercase letters ('A'-'Z'). If half of the characters in the child's sample match the characters in the corresponding positions in the man's sample, and the remaining characters in the child's sample match the characters in the corresponding positions in the mother's sample, then the man is most likely the father. On the other hand, if it is impossible to partition the child's sample such that half of the characters match the man's and the other half match the mother's, then the man is definitely ruled out as the father.

For example, suppose the child's sample is "ABCD" and the mother's sample is "AXCY" (all quotes for clarity only). The 'A' and 'C' in the child's sample must have come from the mother, so the 'B' and 'D' must have come from the father. If the man's sample is "SBTD", then he is most likely the father, but if the man's sample is "QRCD", then he is definitely not the father. Note in the latter case that the man was definitely ruled out even though half of his sample (the 'C' and 'D') did in fact match the child's.

Your method will take samples from the child and the mother, as well as samples from several men, and return the indices of the men who cannot be ruled out as the father, in increasing order.

Problem Statement

Fabian is in charge of a law firm working on an important case. For a case coming up, he needs a specific folder which is stored in one of the filing cabinets arranged in a line against the wall of the records room. He has assigned a number of workers to find the folder from the filing cabinets. He doesn't want the workers to get in each other's way, nor does he want folders from different filing cabinets getting mixed up, so he has decided to partition the cabinets, and assign a specific section to each worker. Each worker will have at least 1 cabinet to search through.

More specifically, Fabian wants to divide the line of filing cabinets into N sections (where N is the number of workers) so that every cabinet that the ith worker looks through is earlier in the line than every cabinet that the jth worker has to look through, for i < j.

His initial thought was to make all the sections equal, giving each worker the same number of filing cabinets to look through, but then he realized that the filing cabinets differed in the number of folders they contained. He now has decided to partition the filing cabinets so as to minimize the maximum number of folders that a worker would have to look through. For example, suppose there were three workers and nine filing cabinets with the following number of folders:

1. 10 20 30 40 50 60 70 80 90

He would divide up the filing cabinets into the following sections:

1. 10 20 30 40 50 | 60 70 | 80 90

The worker assigned to the first section would have to look through 150 folders. The worker assigned to the second section would have to search through 130 folders, and the last worker would filter through 170 folders. In this partitioning, the maximum number of folders that a worker looks through is 170. No other partitioning has less than 170 folders in the largest partition.

Write a class FairWorkload with a method getMostWork which takes a int[] folders (the number of folders for each filing cabinet) and an int workers (the number of workers). The method should return an int which is the maximum amount of folders that a worker would have to look through in an optimal partitioning of the filing cabinets. For the above example, the method would