Supporting Information

Mason and Watts 10.1073/pnas.1110069108

SI Text

Experimental Setup. ***Amazon's Mechanical Turk.*** Amazon's Mechanical Turk (AMT) is a Web-based labor market with large volumes of small tasks [called human intelligence tasks (HITs)] offered for small reward. Typical tasks include image labeling, sentiment analysis, or classification of uniform resource locators, and wages are typically on the order of \$0.01–\$0.10 per HIT. AMT is becoming increasingly popular with behavioral science researchers, in part because it allows experiments to be run faster and more inexpensively, and in part because it provides access to a potentially much broader cross-section of the population than is typical of university-based laboratory experiments (1–4). Accordingly, we posted each of our experimental sessions as a HIT and recruited workers to participate in the experiment. After seeing a screenshot of the experiment with explanatory text, workers could choose to accept the HIT; at that point, the work was officially assigned to them and they could begin participating in the study. More details about AMT and its application to behavioral science experiments are given by Mason and Suri (5).

Panel Recruitment. In our experiment, we required all players to participate simultaneously, which is atypical for tasks on AMT. To do this, we follow Suri and Watts (6) in creating a virtual "waiting room." Once arriving participants accepted the HIT, they were directed to a screen informing them that the experiment had not yet filled, along with how many remaining players were required. Once all positions had been filled, participants in the waiting room were informed that the game was about to commence.

Consistent with previous work (6), we found that posting the HIT on AMT was insufficient to fill networks of size $n = 16$ in a reasonable time, resulting in participants abandoning the waiting room and the HIT being terminated. To alleviate this problem, we ran a series of experiments with simple networks comprising $n = 8$ subjects, during which parameters were varied to produce landscapes that were neither too easy (the peak is always found) nor too difficult (the peak is never found). After participating in at least one experiment, the subjects were asked to report their demographics and then were given the opportunity to opt-in to a standing panel of experienced players who were familiar with the rules of the game and were willing to be contacted for future games. The experiments reported in this article were then conducted by recruiting from this standing panel.

Participants. All participants were recruited through AMT. A total of 120 unique players participated in the 232 games reported in this study. Of these, 58 reported their sex as male and 61 reported their sex as female. The median reported age was 28 y, with quartiles of 23 y and 34 y. The modal response for annual household income was "less than \$30,000," and the modal response for the highest level of education attained was "Bachelor's Degree." The majority of participants played in 3 or fewer games, although there were a few who played in as many as 12 (Fig. S1).

Game: Wildcat Wells. As described in the main text, the collaborative problem-solving task that we studied was presented to participants in the form of a game called "Wildcat Wells," in which players were tasked with exploring a desert landscape in search of hidden oilfields. When $n = 16$ participants had accepted the task on AMT, the session would begin. As shown in Fig. S2, each player would see (*a*) the full landscape to be explored; (*b*) his or her own current and previous locations; (*c*) the scores corresponding to those locations; and (*d*) the previous locations and corresponding scores of his or her three assigned collaborators (i.e., network neighbors), whose locations are made visible. On each turn, all players could choose any of the 100×100 grid squares as their next location to "drill." If the player hovered over one of the drill locations, the exact score would appear and the corresponding bar in the player's (or neighbor's) history would be highlighted. Similarly, hovering over a bar in the history would show the points and the corresponding location on the landscape.

Players had to select and submit a location to drill within some amount of time; in the first 2 rounds of the game, they had 60 s to make their decision, and in the remaining 13 rounds, they had 30 s. If they did not submit a location in that time, the round was skipped and they earned no points for that round (therefore, no information about the fitness landscape was obtained; an example of how this was displayed to the user can be seen in the player's round 4 in Fig. S2). The total number of points the players had accumulated was always displayed to them. At the end of the 15 rounds, they were shown a summary of the game as well as the total number of points they earned before advancing to the next game. In all, they played eight games, one for each network topology, which typically took a total of 50–60 min.

All payoffs were determined by a hidden fitness landscape, which comprised two components: a main "peak" and a background "noise" distribution, which are described in the next section. For each game, the participants were paid according to the points that they earned across all eight games in each session at a rate of \$0.00075 per point. Thus, participants were financially motivated to find the main peak and, failing that, to find the minor peaks present in the noise. On average, participants received a total of 819 points per game, which translates to roughly \$0.614 per game, or \$4.91 per session. When the peak was not found, participants earned \$0.449, on average, with a maximum of \$0.645. When the peak was found, participants earned \$0.736, on average, with a maximum of \$1.118. Carried to the extreme, if players never found the peak in all eight games, they could have earned, at most, \$5.16, whereas if they found the peak in all eight games, they could have earned up to \$8.94.

Generating Perlin Noise. Perlin noise is a method for generating pseudorandom noise that was originally developed to improve the realism of computer graphics (7) but can also be used to create "landscapes" of arbitrary ruggedness.

As noted in the main text, Perlin noise (8) is created by summing a sequence of "octaves," where each octave is generated in three stages:

i) For some integer $\omega \in [\omega_{min}, \omega_{max}]$, divide the grid into $2^\omega \times 2^\omega$ cells and assign a random number drawn uniformly from the interval [0,1] to the coordinate at the center of each cell.
ii) Assign values to all other coordinates in the $L \times L$ grid by smoothing the values of cell centers using bicubic interpolation.
iii) Scale all coordinate values by ρ^ω, where ρ is the "persistence" parameter of the noise distribution.

Figs. S3 *A* and *B* and S4 illustrate these steps schematically in one dimension: Fig. S3*A* shows the allocation of random values to cells; Fig. S3*B* shows the interpolation between random values to form a smooth surface; and Fig. S4 shows the summation of several octaves for different values of the persistence.

Generating Networks. The networks to which participants were assigned were constructed as follows. Starting with a regular

random graph comprising $n = 16$ nodes, each with $k = 3$ neighbors, a series of degree-preserving random rewirings was conducted, where only rewirings that decreased a specified loss function $f(\phi)$ were accepted, where ϕ was one of several network statistics of interest, as described below. The rewiring procedure terminated when no more rewirings were possible, and the whole procedure was repeated 100 times to avoid local minima. There are no guarantees that the discovered graph is actually the global optimum, but the resultant graphs were sufficient to capture variance in the relevant features.

The loss function $f(\phi)$ either maximized or minimized the average ϕ across all nodes in the graph, the highest ϕ for any single node, or the lowest ϕ for any single node. The four network features we focused on, enumerated below, were chosen because they were known to be related to information flow in networks. The features of the resultant graphs are shown in Table S1.

Betweenness. Betweenness centrality captures the amount of information that flows between nodes in the network by way of the target node. Specifically, it is the proportion of shortest paths between pairs of nodes that pass through the central node (9). More formally, if we let σ_{st} denote the number of shortest paths from s to t and let $\sigma_{st}(u)$ denote the number of shortest paths from s to t that go through u, then

$$C_b(u) = \sum_{s \neq u \neq t \in V} \frac{\sigma_{st}(u)}{\sigma_{st}}.$$

Closeness. Closeness centrality was originally devised by Beauchamp (10) to indicate how reachable a node was from all other nodes in the network. It is the reciprocal of the average length of the shortest path between the node and all other nodes:

$$C_c(u) = \frac{n-1}{\sum_{v \in V} \mathrm{d}(u,v)}.$$

Here, n is the number of nodes in the network, and $\mathrm{d}(u, v)$ is the length of the shortest path from u to v.

Clustering. The clustering coefficient (11) is a measure of the connectedness of a neighbor's contacts. If $\Gamma(u)$ denotes the set of nodes that u connects to ($\Gamma(u) = \{v \mid (u, v) \in E\}$) and $\deg(u) = ||\Gamma(u)||$ is the degree of node u, the clustering coefficient of node u is given by:

$$\mathrm{cc}(u) = \frac{|\{(i,j) | i \in \Gamma(u) \text{ and } j \in \Gamma(u)\}|}{\binom{\deg(u)}{2}}.$$

Constraint. Network constraint, developed by Burt (12), is intended to capture the extent to which a person bridges different groups. Burt (12) defined the network constraint of node u as follows:

$$\mathrm{nc}(u) = \frac{1}{\deg(u)^2} \sum_{v \in \Gamma(u)} \left(p_{uv} + \sum_{w \in \Gamma(u), w \neq v} p_{uw} p_{wv} \right)^2. \qquad \textbf{[S1]}$$

Here, p_{uv} denotes the fraction of direct attention that node u gives to node v. The sum $\sum_{w \in \Gamma(u), w \neq v} p_{uw} p_{wv}$ is the total fraction of indirect attention that u gives to v through some intermediary w. When the sum of the direct plus indirect attention u gives v is high, u is wasting effort on giving redundant attention to v. Therefore, network constraint is "better" for the individual when it is small.

Eq. **S1** defines network constraint for a weighted, directed network. In an undirected, unweighted network, p_{uv} is equal to the inverse of degree; thus, Eq. **S1** reduces to:

$$\mathrm{nc}(u) = \frac{1}{\deg(u)^2} \sum_{v \in \Gamma(u)} \left(1 + \sum_{w \in \Gamma(u), w \neq v} p_{wv} \right)^2.$$

In either case, the measure is minimized when none of u's neighbors are neighbors with each other; in that case, it evaluates to $\frac{1}{\deg(u)}$.

SI Results

Excluded Games. To verify the robustness of the results, we also conducted the analyses after excluding the 61 games in which a player did not take an action in more than half of the rounds. The results were qualitatively and quantitatively similar for all 232 games. As with the complete set of games, networks with higher clustering showed less exploration and networks with shorter path lengths diffused information about the peak faster, leading to better success for the players.

Collectives earned 30 points more than independent searchers ($t = 89.7$, $P \approx 0$) when the peak was found and 8 points more when the peak was not found ($t = 34.7$, $P \approx 0$). The peak was found in 107 of the 171 games, or 62.6% of the time. The relationship between path length and points was again negative and marginally significant ($\beta = -3.89$, $P \approx 0.06$). Efficient networks again earned more points overall than inefficient networks, although there is somewhat more variance. The time for information about the peak to reach a node was exactly equal to the path length separating them 73% of the time. The difference between the efficient and inefficient networks is even clearer with the 61 games excluded, both in the frequency of finding the peak (although the differences are still not significant) and in the amount of copying. It is also still true that increased clustering is associated with neighbors imitating each other and increased imitation is associated with increased copying by the focal node. Similarly, the results still show that players who copied more (before the peak was found) earned more points, on average, and that the probability of finding the peak in the subsequent round was negatively related to the number of players copying each other in the current round.

Exploration of Independent Searchers. The independent searchers did not find the peak significantly more often than the networked collectives. Seventeen (70.8%) of the 24 groups of independent searchers had at least one person find the peak, whereas 138 (59.5%) of 232 groups of collective searchers had at least one person find the peak (Fig. S6). Although the independent searchers had a higher proportion of groups that found the peak, this was not significant with a χ^2 test ($\chi^2 = 0.746$, not significant).

This is somewhat surprising, because the independent searchers were exploring more with respect to the amount of space they were covering. The independent searchers, not being able to observe where the other players were exploring, naturally visited more unique points in the landscape than the networked groups, as shown in Fig. S7*A*. However, this difference disappears when focusing on the round before the peak is found (Fig. S7*B*).

One possible explanation for this is the fact that the independent searchers tended to gravitate toward their own best solutions as the end of the game approached. This can be seen in Fig. S8, which shows the average distance from the player's best solution decreases for both the networked players and the independent searchers, even when the peak is not found.

Efficiency Predicts Success. To determine whether efficiency predicts success, we fit a linear model of the form $pts \sim \alpha + \beta pl$, where pts is the average points earned by players in a particular network and pl is the average path length in that network. The relevant parameter estimates and significance results are shown

in Table S2. The advantage of short path length can be also seen in Fig. S9, which shows the path length and average points for each graph type.

Impact of Network Position on Individual Performance. As stated in the main text, for centralized networks, individuals in central positions performed well relative to individuals in peripheral positions, but individuals in decentralized networks generally outperformed all individuals in centralized networks. Fig. S10 supports the first part of this statement, showing that higher betweenness centrality and closeness centrality both correlate with faster convergence time for individuals, where convergence time refers to the number of rounds after the peak is discovered required for an individual to locate the maximum possible score. By contrast, Fig. S10 also shows the opposite tendency for clustering coefficient and network constraint.

To illustrate this result more clearly, Fig. S11 shows the average number of rounds required for each position to locate the peak (given the peak is found at all), where lighter colors correspond to faster discovery times. From this figure, it is clear that peripheral nodes in centralized networks suffer for the simple reason that when the peak is discovered far from them, the information must propagate for several steps to reach them. By contrast, central nodes always receive the information within a few steps. Moreover, all nodes in decentralized networks receive the information within a few steps.

SI Simulations

Our main finding regarding the superiority of efficient networks raises three related questions with respect to the previous findings of agent-based models (13, 14). First, what is it about the rules governing artificial agents that leads inefficient networks to outperform efficient networks in the simulation but not in our experiments? Second, do humans or agents perform better overall? Third, do either agents perform close to what is theoretically achievable (i.e., optimal)?

To address these questions, we conducted two sets of agent-based simulations, where networks of 16 agents searched a fitness landscape identical to that in the experiments using human subjects. These two agents differed with respect to the behavioral rules, which are described next.

Rule LF. This rule was adapted from Lazer and Friedman (13), using the following key passage: "We assumed that in each round of the simulation, actors' decisions proceed in two stages. In the first stage, each actor evaluates whether anyone to whom he or she is connected has a superior solution. If so, he or she copies the most successful individual to whom he or she is connected. If no one is more successful, then the agent myopically searches for a better strategy than he or she has in the status quo strategy. To capture this myopia, we assumed that agents examine the impact of randomly changing one digit of their status quo solution, and if that potential change offers an improvement, they change their solution" (ref. 13, p. 2).

Mapping this rule to our own context, we assume whenever a network neighbor has a better score than the focal individual, he or she should copy his or her neighbor. When the focal individual has the highest score, however, there is no obvious equivalent of myopic search; thus, we consider a family of models, LF_r, where r constitutes the radius of myopic search. Specifically, we consider four values of r: 3, 6, 12, and 20. We note that $r = 6$ is equivalent to the radius of the main peak and $r = 20$ is roughly half of the grid length ($L = 50$); hence, the $r = 3$ setting is extremely myopic, whereas $r = 20$ is only weakly restrictive. Our choice of LF_r models thus allows us to explore effectively a wide range of sensible interpretations of myopic search.

Rule O. We have derived a rational best-response rule with respect to a simplified fitness landscape with just two payoffs: nonpeak and peak. Clearly, this landscape differs from the landscapes that our human subjects were searching in potentially important ways; indeed, as we will show below, the derived "optimal" rule is far from optimal when applied to the actual landscapes. Nevertheless, the modeling exercise still proved useful by exposing the deficiencies in our assumptions.

Specifically, we assume the following notation:

P_t = expected payoff to exploration t steps from end
p = payoff from peak
n = payoff from nonpeak
f = p[finding peak in 1 round of exploration] = R/L
R = width of peak
L = side length of grid
T = number of rounds

From this, we proceed by backward induction.

At round $T - 1$, and assuming the player has not yet found the peak, the expected payoff to exploring a new location is:

$$P_{T-1} = fp + (1-f)(n). \quad \textbf{[S2]}$$

The first part of this equation, fp, is the expected utility of finding the peak on the next round, and the second part of the equation, $(1-f)n$, is the expected utility of not finding the peak on the next round. From this, it follows that the expected utility of exploring on round $T - 2$ is the value of finding it that round (and exploiting it for the remaining rounds) plus the value of not finding it and exploring on the remaining rounds: P_{T-1}. This process leads to the following recursive payoff function:

$$\begin{aligned} P_{T-2} &= 2fp + (1-f)(n + P_1) \\ P_{T-3} &= 3fp + (1-f)(n + P_2) \\ &\vdots \\ P_1 &= nfp + (1-f)(n + P_{t-1}) \end{aligned}$$

Given this recursive payoff function, for any round $t = (T - \tau)$, we can compute explicit payoffs, P^*_τ for t = 1,2,...$T - 1$. From this, we can determine the following best-response rule: If P^*_τ is greater than the highest observed solution, choose a new location uniformly at random; otherwise, copy the highest observed solution.

SI Results

Comparison with *LF* Agents. Figs. S12–S15 summarize our findings:

i) All *LF* agents performed worse than human agents, where greater myopia corresponded to lower likelihood of finding the main peak and, hence, worse performance. Also, all *LF* agents copied far more than human agents.

ii) Inefficient networks slightly outperformed efficient networks for LF_6 and LF_{12} agents (intermediate myopic). However, for highly myopic (LF_3) networks, and also for only weakly myopic (LF_{20}) networks, performances did not appear to depend on efficiency.

Our interpretation of these results is that the main result of Lazer and Friedman (13), namely, the superiority of inefficient networks, depends on a rather nongeneric tradeoff. Specifically, because agents start exploring from different (randomly chosen) locations on the fitness landscape, the slower they converge on a solution, the more of the space they are likely to explore. Because inefficient networks disseminate information about globally suboptimal solutions more slowly than efficient networks, they will spend more time exploring than efficient networks. At the same time, however, the radius of myopic search

determines how much of the space any individual can explore, and thus the likelihood that the main peak will be found at all.

The interaction between these two conflicting forces produces the following result. When search is extremely myopic, neither efficient networks nor inefficient networks are likely to find the peak; hence, no performance difference is registered between them. At the other extreme, when search is only weakly myopic, both efficient and inefficient networks are likely to find the peak; hence, again, there is no performance difference and all networks perform better. Only when search is of intermediate myopia (the LF_6 and LF_{12} cases) does the time delay associated with inefficient networks translate into an advantage in terms of finding the peak.

In recovering the *LF* result, in other words, we have also shown that the claimed benefit of inefficiency depends somewhat sensitively on the choice of parameters. Moreover, this particular choice does not result in the best overall performance for either efficient or inefficient networks, because even intermediate myopic search decreases the likelihood of the main peak being found. Nor do real human agents appear to embody this particular form of myopia, thus accounting for why we see no benefit to inefficiency in our experiments.

To summarize, our conclusion that human agents behave in systematically different ways than is assumed in agent-based models appears sound, although the more detailed comparison also reveals (*a*) that the *LF* model is generically inferior to what human agents do and (*b*) that the *LF* inefficiency result, although replicable in our problem space, arises only for a somewhat narrow range of parameters and that this range is suboptimal even for *LF* agents.

Comparison with *O* Agents. Fig. S16 summarizes the performance of the *O* strategy agents. Curiously, they perform much worse than human agents, for three reasons:

i) They find the main peak much less frequently.
ii) They are less likely to exploit it when they do.
iii) They are also less likely to exploit local maxima.

Obviously, there is something very nonoptimal about this rule when applied to the actual landscapes rather than the simplified landscape assumed to derive the rule.

The problem, we believe, is that the proposed simplification leads *O* agents (in contrast to *LF* agents) to explore far more than they should. The actual landscape, that is, exhibits substantial spatial correlations in the noise that human agents are clearly exploiting. In contrast, the simplified landscape (resembling a "needle in a haystack" landscape) used to derive optimal behavior offers no exploitable correlations; hence, *O* agents (*a*) do not locally explore when they are near peak and (*b*) continue exploring almost until the last round.

It is not clear why this behavior results in *O* agents finding the main peak so much less frequently than human agents; however, it is clear that their excessive exploration prevents them from discovering or exploiting local maxima. Moreover, because a simplified landscape also assumes that the peak is flat, in the sense of comprising a single value, even when *O* agents do stumble on the peak, they often continue searching globally rather than switching to local search, which is what the human agents do.

Unfortunately, although the shortcomings of the assumptions underpinning the *O* strategy are now evident, the solution is not. Clearly, human agents are paying attention to spatial correlations in the fitness landscape, and this is clearly helping them a great deal. It would also clearly be of interest to solve for an optimal strategy under increasingly realistic assumptions about the landscape; however, such an exercise is well beyond the scope of this paper.

Instead, we have simulated a variant of the *O* strategy, which we call *O*+. This strategy addresses one of the obvious shortcomings of the *O* strategy, namely, that *O* agents fail to switch to local search when they discover the peak (i.e., if they land very close to the maximum). To solve this problem, the *O*+ strategy simply adds the following heuristic: If highest observed score is greater than 60, search within the radius of the peak, *R*.

One would not expect this strategy to be optimal either (because it does nothing to address the lower likelihood of finding the peak), and, indeed, *O*+ agents still perform worse than human agents. However, as Fig. S17 shows, it does improve performance in the expected way.

Thus, although our analysis fails to answer the question of whether human agents are performing optimally, it does suggest that it is not easy to outperform human agents. Moreover, we find that for *O*+ agents, efficient networks once again outperform inefficient networks, reinforcing our conclusion that the efficient network result is reasonably robust.

1. Mason WA, Watts DJ (2009) *Financial Incentives and the Performance of Crowds*. Proceedings of ACM SIGKDD Workshop on Human Computation (Paris, France) pp 77–85.
2. Ipeirotis PG, Provost F, Wang J (2010) *Quality Management on Amazon Mechanical Turk*. Proceedings of ACM SIGKDD Workshop on Human Computation (Paris, France) pp 64–67.
3. Paolacci G, Chandler J, Ipeirotis P (2010) Running experiments on Amazon Mechanical Turk. *Judgm Decis Mak* 5:411–419.
4. Horton J, Rand D, Zeckhauser R (2011) The online laboratory: Conducting experiments in a real labor market. *Exp Econ* 14(3):399–425.
5. Mason W, Suri S (2011) Conducting Behavioral Research on Amazon's Mechanical Turk. *Behavior Research Methods*, 10.3758/s13428-011-0124-6.
6. Suri S, Watts DJ (2011) A study of cooperation and contagion in networked public goods experiments. *PLoS One* 6(3):e16836.
7. Perlin K (1985) An image synthesizer. *Comput Graph* 19(3):287–296.
8. Perlin K (2002) Improving noise. *ACM Trans Graph* 21(3):681–682.
9. Freeman LC (1977) A set of measures of centrality based on betweenness. *Sociometry* 40(1):35–41.
10. Beauchamp MA (1965) An improved index of centrality. *Behav Sci* 10(2):161–163.
11. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393:440–442.
12. Burt RS (1992) *Structural Holes: The Social Structure of Competition* (Harvard Univ Press, Cambridge, MA).
13. Lazer D, Friedman A (2007) The network structure of exploration and exploitation. *Adm Sci Q* 52:667–694.
14. Fang C, Lee J, Schilling MA (2010) Balancing exploration and exploitation through structural design: The isolation of subgroups and organizational learning. *Organ Sci* 21:625–642.

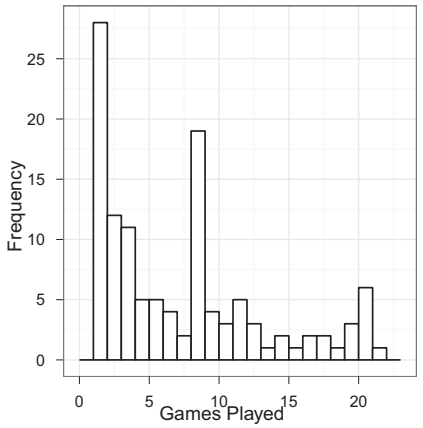


Fig. S1. Distribution of the number of games played. Most participants only played 1 game, although a few participants played as many as 12 different games.

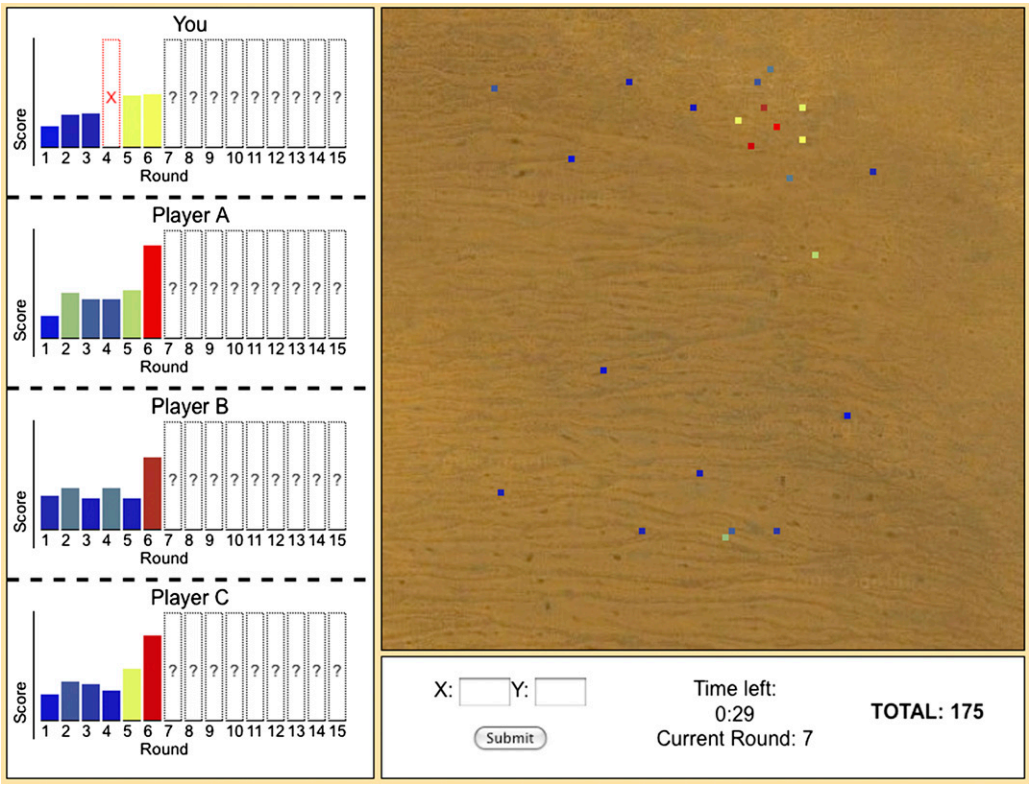


Fig. S2. Screenshot of the game Wildcat Wells, as seen by a single player.

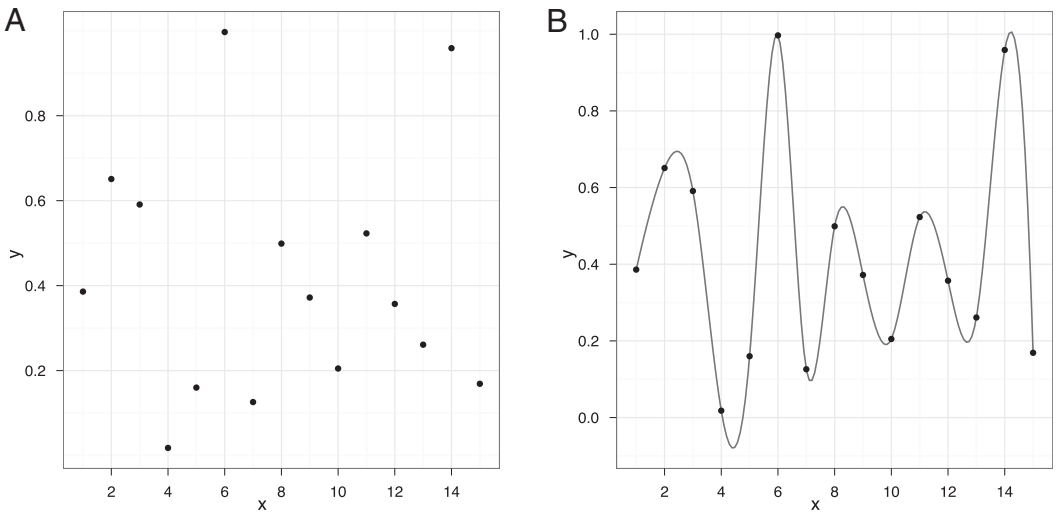


Fig. S3. Construction of one octave for generating Perlin noise in one dimension; a number of random points are chosen (*A*), and a cubic spline is interpolated through the points (*B*).

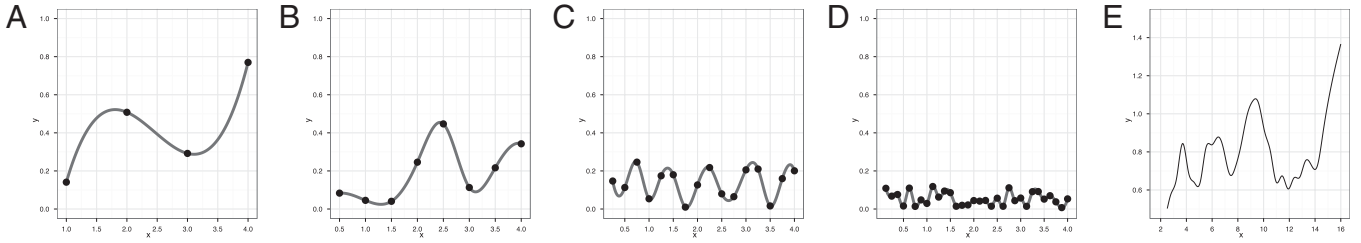
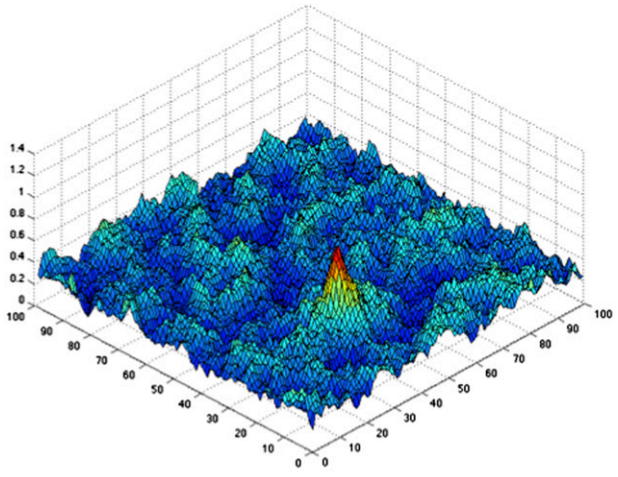
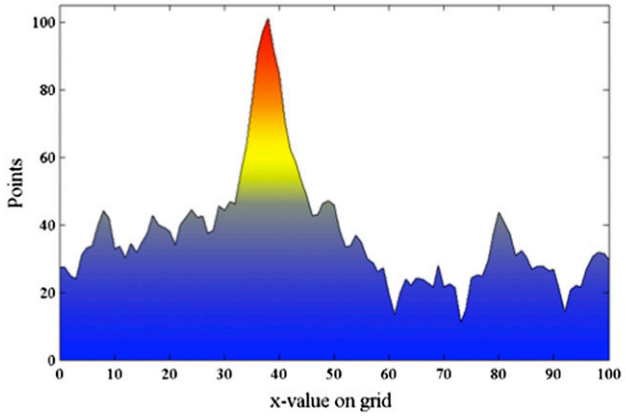


Fig. S4. Several smoothed curves with increasing frequency and decreasing amplitude (*A*–*D*) are summed to create the final Perlin noise distribution (*E*).



(a) A single instance of the fitness landscape



(b) Cross section of the same landscape

Fig. S5. Example of the fitness landscape being searched by the players.

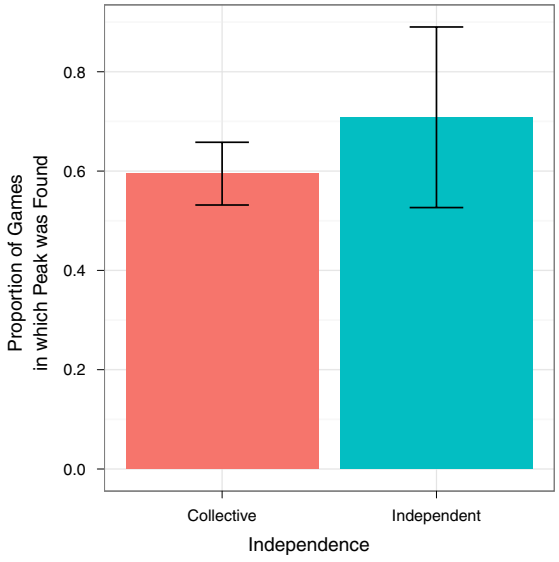


Fig. S6. Proportion of games in which the peak was found by at least one searcher for independent searchers and collectives. Error bars show binomial proportion confidence interval.

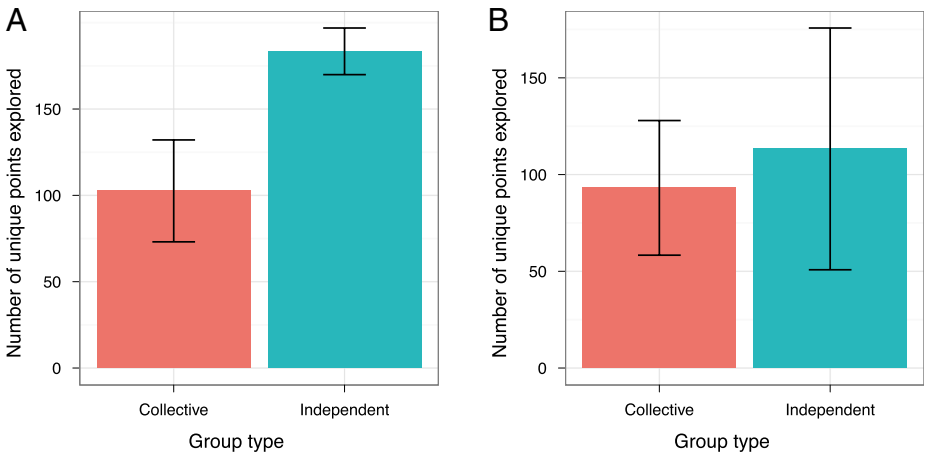


Fig. S7. Average number of unique coordinates visited by a group of players for the networked groups and independent searchers, for all rounds (*A*) and for only the rounds before the first player finds the peak (*B*).

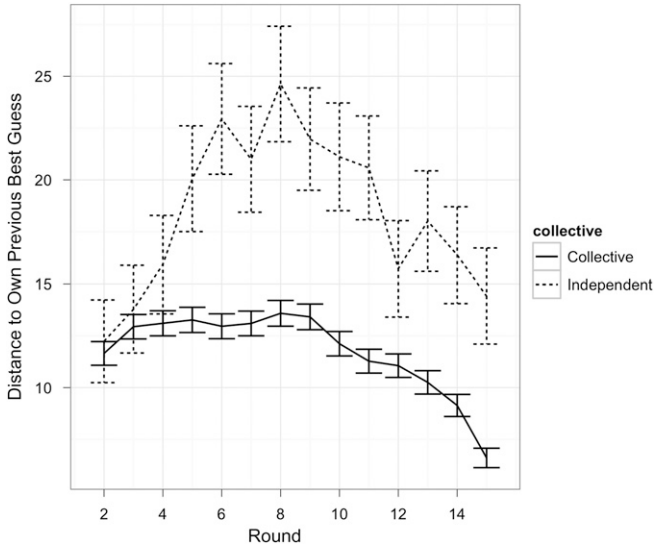


Fig. S8. Average distance from the each participant's highest scoring location across all rounds for networked groups and independent searchers.

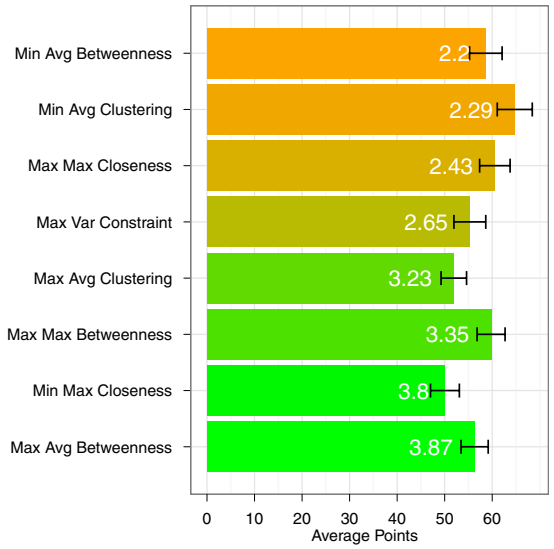
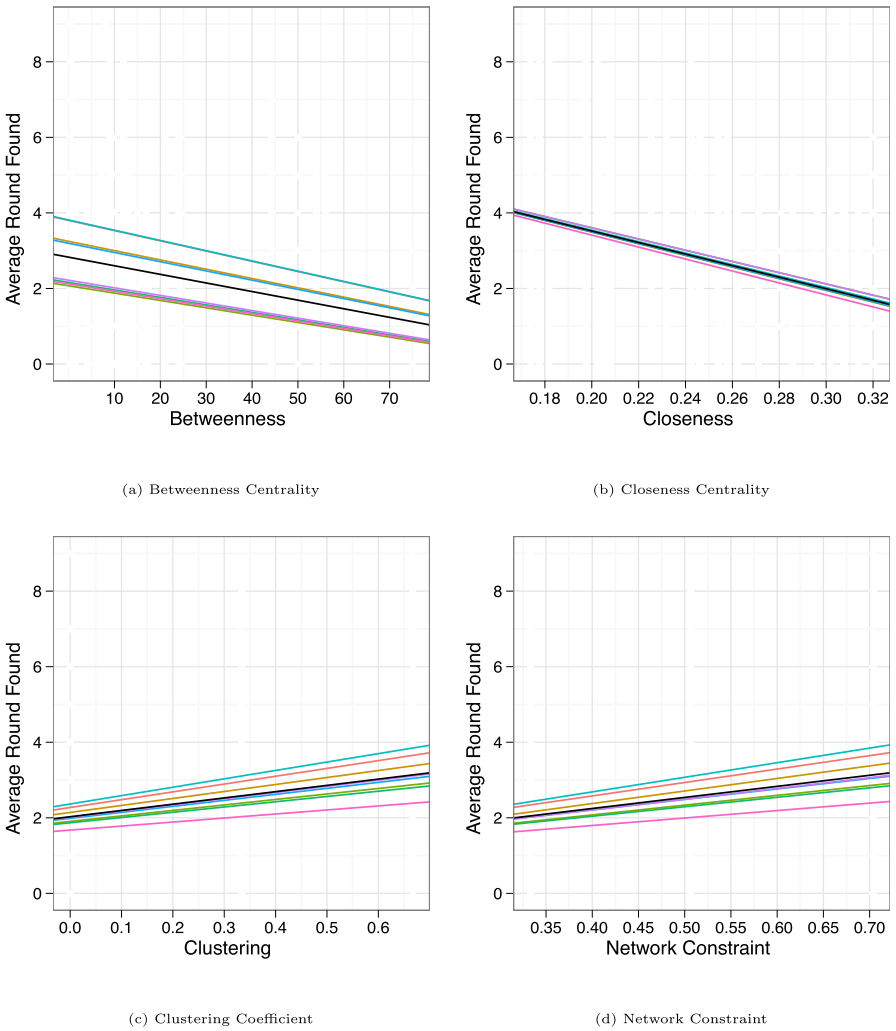


Fig. S9. Points earned by graph across all games and rounds. The average path (Avg) length is indicated by the color, where orange is shorter and green is longer, and is displayed on the bar in white. Max, maximum; Min, minimum; Var, variance.



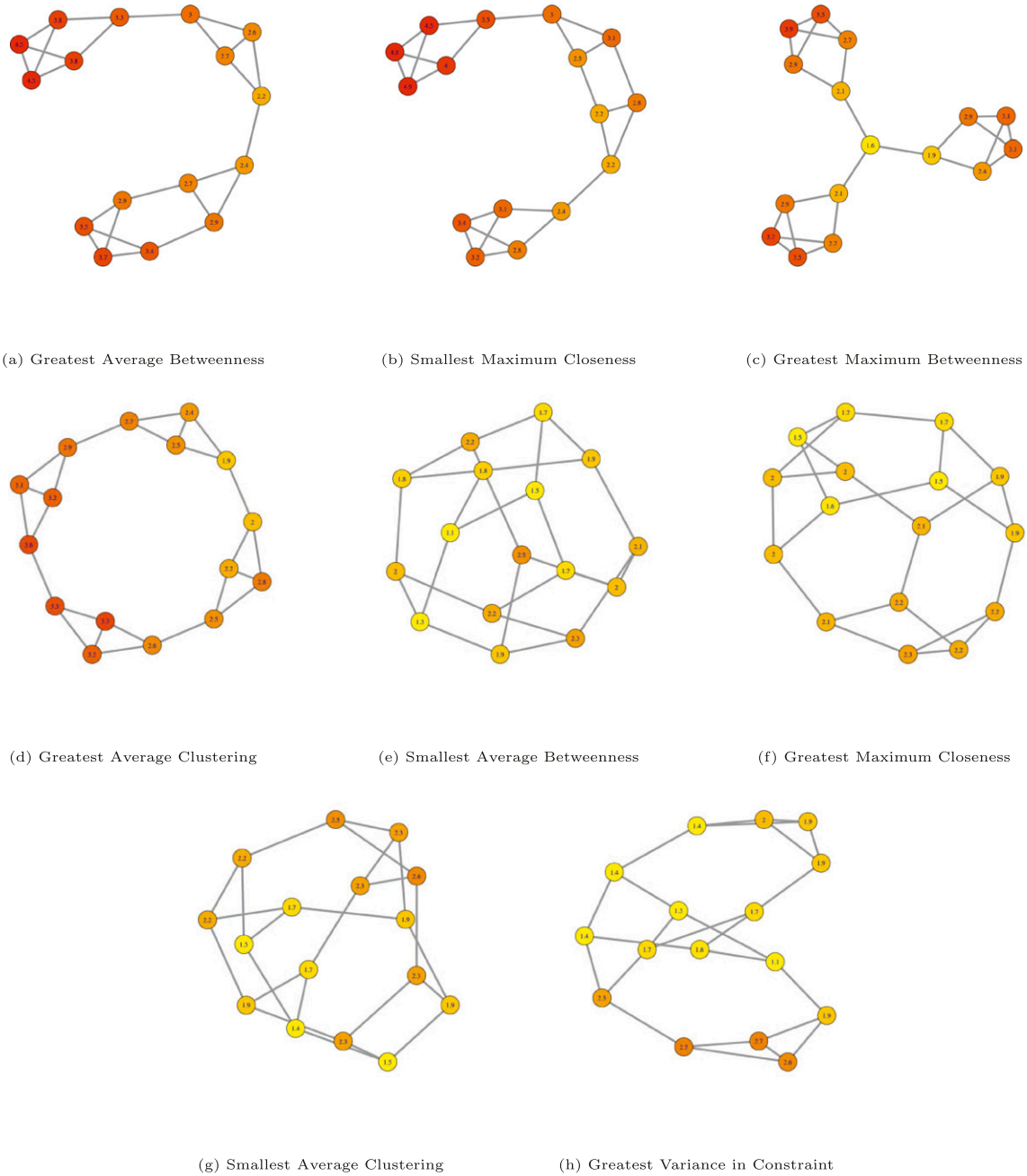
(a) Betweenness Centrality

(b) Closeness Centrality

(c) Clustering Coefficient

(d) Network Constraint

Fig. S10. Fit for hierarchical linear model of convergence time as a function of betweenness centrality (*A*), closeness centrality (*B*), clustering coefficient (*C*), and network constraint (*D*). Clearly, higher betweenness and closeness centrality correlates with faster convergence time, whereas the opposite applies to clustering coefficient and network constraint. Each of the colored lines represents the best fitting line for each of the eight network structures.



(a) Greatest Average Betweenness

(b) Smallest Maximum Closeness

(c) Greatest Maximum Betweenness

(d) Greatest Average Clustering

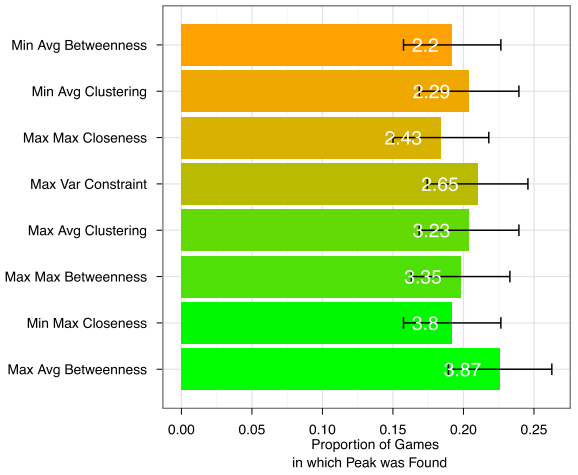
(e) Smallest Average Betweenness

(f) Greatest Maximum Closeness

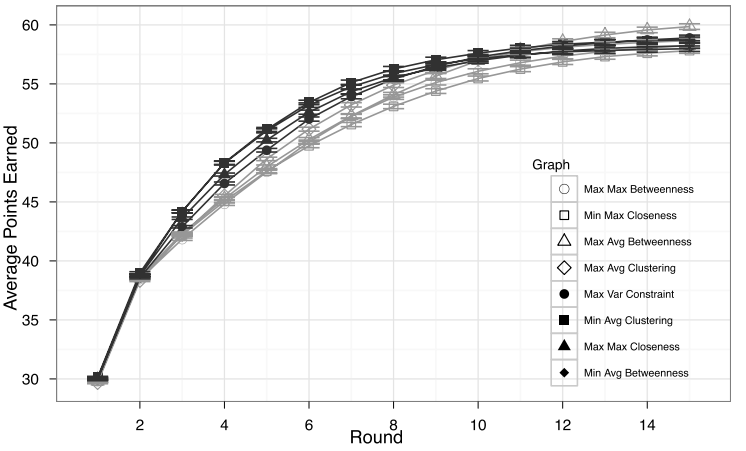
(g) Smallest Average Clustering

(h) Greatest Variance in Constraint

Fig. S11. Average number of rounds to find the peak after it has been discovered for each position within each graph. Brighter colors indicate faster discovery.

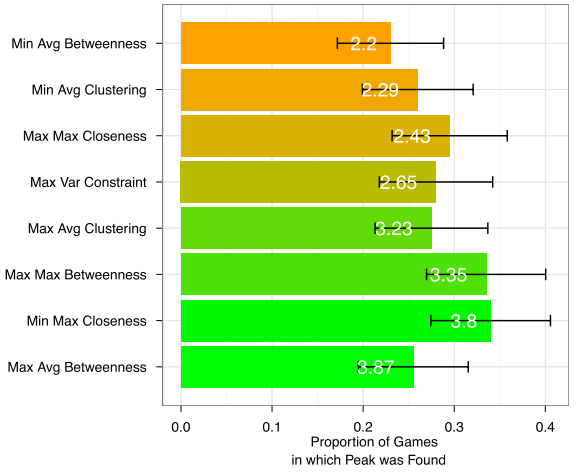


(a) Probability of Finding Peak

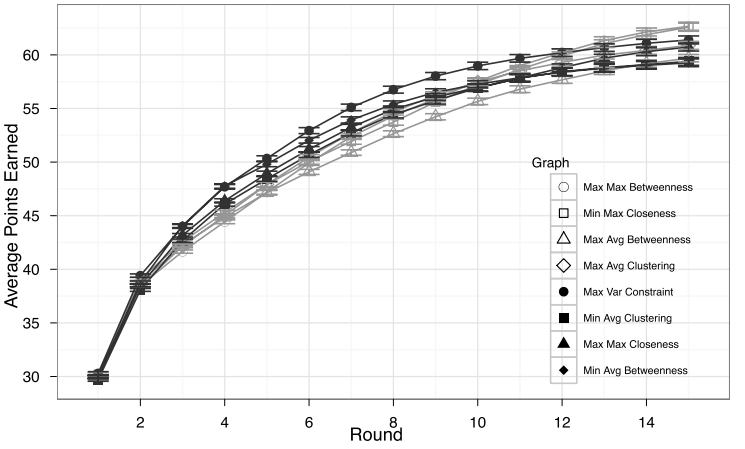


(b) Points Earned by Round

Fig. S12. Probability of finding the peak for each graph type and points earned over rounds for the LF_3 agents. The average path length is indicated by the color, where orange is shorter and green is longer, as well as the number located on the bar. Avg, average; Max, maximum; Min, minimum; Var, variance.

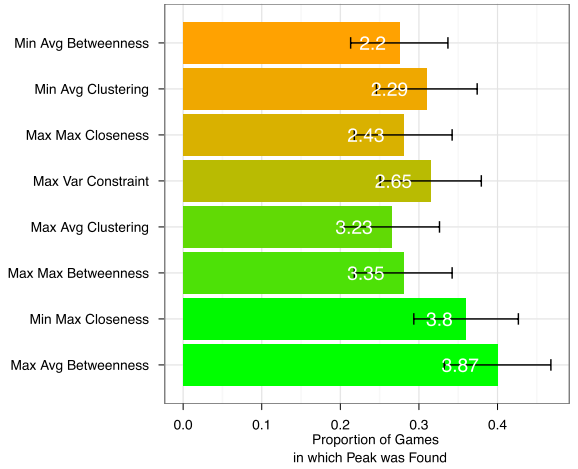


(a) Probability of Finding Peak

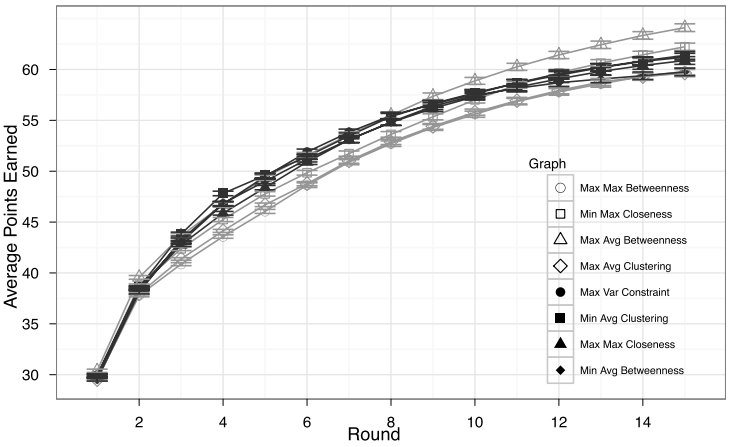


(b) Points Earned by Round

Fig. S13. Probability of finding the peak for each graph type and points earned over rounds for the LF_6 agents. The average path length is indicated by the color, where orange is shorter and green is longer, as well as the number located on the bar. Avg, average; Max, maximum; Min, minimum; Var, variance.

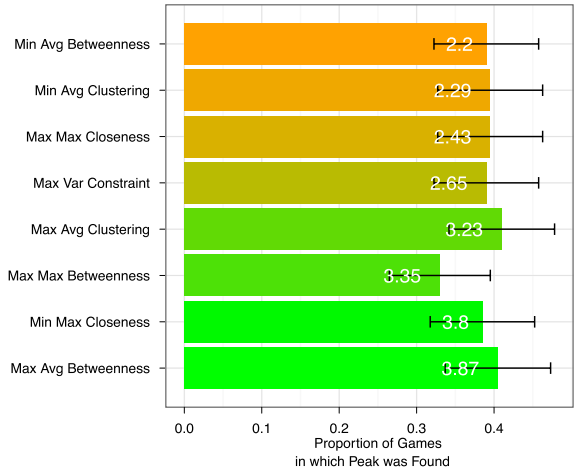


(a) Probability of Finding Peak

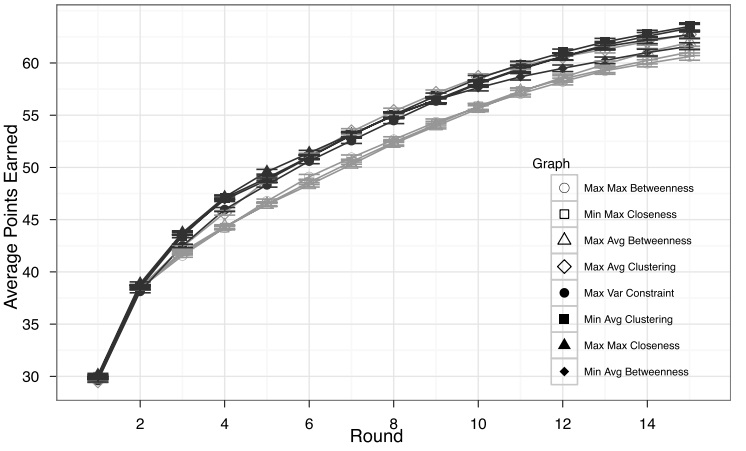


(b) Points Earned by Round

Fig. S14. Probability of finding the peak for each graph type and points earned over rounds for the LF_{12} agents. The average path length is indicated by the color, where orange is shorter and green is longer, as well as the number located on the bar. Avg, average; Max, maximum; Min, minimum; Var, variance.

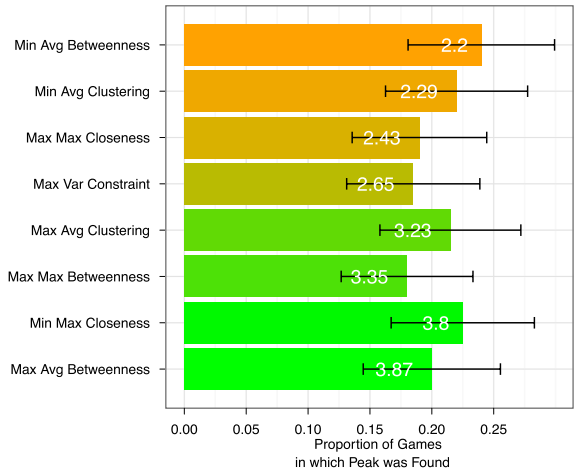


(a) Probability of Finding Peak

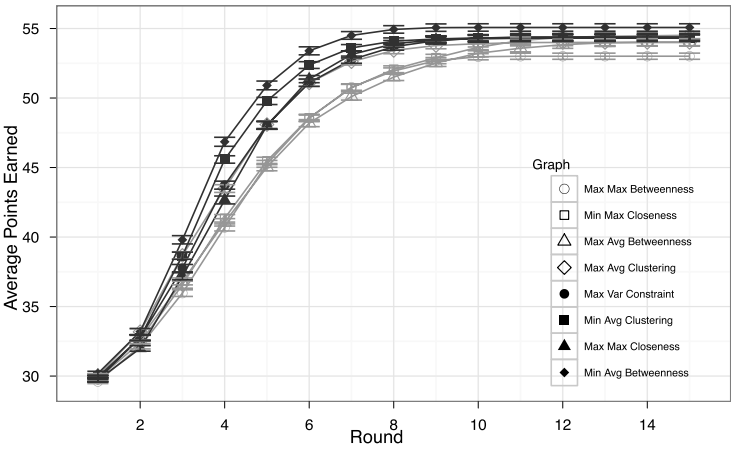


(b) Points Earned by Round

Fig. S15. Probability of finding the peak for each graph type and points earned over rounds for the LF_{20} agents. The average path length is indicated by the color, where orange is shorter and green is longer, as well as the number located on the bar. Avg, average; Max, maximum; Min, minimum; Var, variance.

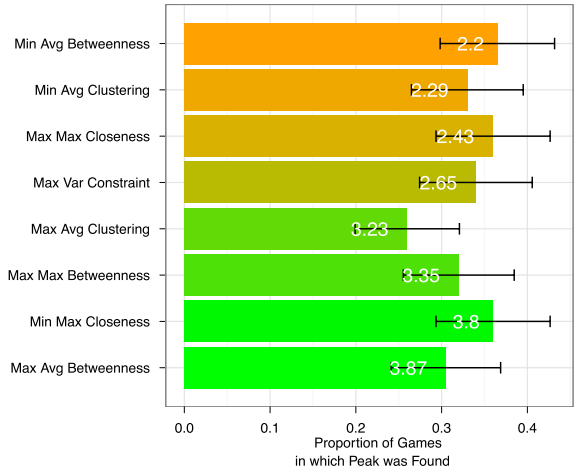


(a) Probability of Finding Peak

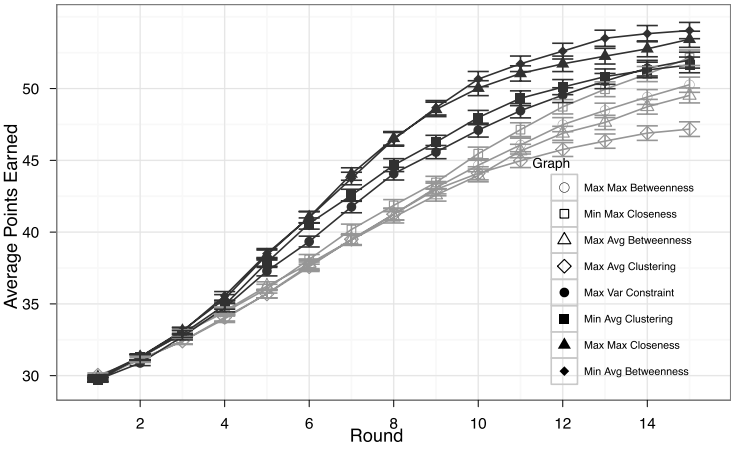


(b) Points Earned by Round

Fig. S16. Probability of finding the peak for each graph type and points earned over rounds for the *O* agents. The average path length is indicated by the color, where orange is shorter and green is longer, as well as the number located on the bar. Avg, average; Max, maximum; Min, minimum; Var, variance.



(a) Probability of Finding Peak



(b) Points Earned by Round

Fig. S17. Probability of finding the peak for each graph type and points earned over rounds for the *O+* agents. The average path length is indicated by the color, where orange is shorter and green is longer, as well as the number located on the bar. Avg, average; Max, maximum; Min, minimum; Var, variance.

Table S1. Structural properties of communication networks used in game

Topology	Radius	Diameter	Closeness	Betweenness	Clustering	Constraint
Min avg betweenness	3	3	0.45	0.09	0	0.33
Max max closeness	3	5	0.41	0.1	0.06	0.36
Min avg clustering	3	4	0.44	0.09	0	0.33
Max var constraint	3	6	0.39	0.12	0.25	0.47
Max avg clustering	6	6	0.31	0.16	0.5	0.6
Max avg betweenness	5	9	0.27	0.2	0.44	0.57
Min max closeness	5	9	0.27	0.2	0.37	0.53
Max max betweenness	3	6	0.31	0.17	0.37	0.54

avg, average; max, maximum, min, minimum; var, variance.

Table S2. Parameter estimates for linear model predicting points by path length of graph

	Estimate	SE	*t*	*P*
α	70.95	5.46	12.99	$\approx$0
β	−4.63	1.79	−2.58	0.01