Chapter 14
Talent management in triadic organizational architectures

Marco LiCalzi and Lucia Milone

Abstract We study a model of team problem-solving over a large solution space. Compared to the existing literature, we allow for heterogeneity both in the organizational architectures and in the agents' cognitive abilities; moreover, we introduce a more expressive performance measure. We find a robust ranking of the triadic architectures with respect to their effectiveness and provide a key recommendation for talent management in partial hierarchies.

14.1 Introduction

This paper considers a team of agents of limited problem-solving ability, who must explore a large solution space. Each agent can access only a portion of the space, but may disclose the results of his search to the teammates. By exchanging information, the members of the team may jointly solve the problem even when no agent alone might. More talented agents possess higher cognitive abilities.

LiCalzi and Surucu (2012) proposed this framework to explore the importance of diversity in agents' toolboxes for the team performance. They studied the case of a complete network, where all agents can exchange information with each other *ad libitum*, providing sufficient conditions for team success. However, their work muted the important question of how the organizational architecture affects the quality of the search carried out by a team; see Mihm et al. (2010). This work extends the model in three directions and ask two new questions.

The model is generalized by allowing for (partially) hierarchical organizational architectures, where each agent has only one chance to act and communication flows from one to another agent are unidirectional. We restrict attention to the triadic architectures that form a connected graph. Up to permutations, there are three possible architectures: line, star-in, and star-out. The second generalization is that we allow agents to have different cognitive abilities. The last contribution upgrades the performance measure from a binary indicator (whether the optimal solution is found or not) to a multivalued indicator (which fraction of the solution space is searched).

Within this framework, we obtain sharp qualitative answers to the following two questions. First, which organizational architecture is more effective? We find that the four organizational architectures rank as follow: complete $>$ line $>$ star-in $\sim$

Marco LiCalzi
Dept. of Management, Università Ca' Foscari Venezia, e-mail: licalzi@unive.it

Lucia Milone
Dept. of Economics and Finance, LUISS Roma, e-mail: lmilone@luiss.it

A. Teglio et al. (eds.), *Managing Market Complexity: The Approach of Artificial Economics*, Lecture Notes in Economics and Mathematical Systems 662, DOI 10.1007/978-3-642-31301-1_14,

star-out. For instance, given the cognitive abilities of the agents, a team based on a line architecture tends to outperform a team based on a star-in architecture. Second, given the organizational architecture, how should we place more talented agents to maximize the team performance? For the complete architecture, this is irrelevant; for the line, it is almost indifferent; for the star architectures, the key recommendation is to place the most talented agent in the central position, because performance is positively related to his cognitive ability.

14.2 The model

There is a team of three problem-solving agents of limited ability who work on a demonstrable intellective task. A task is intellective if it involves an activity that can be objectively measured so that people agree on the value of the solution found. And it is demonstrable if, once found by one agent, the correct solution is easily recognizable by his teammates.

We formalize this situation by assuming that the team tries to maximize an objective function V that maps a finite set X of n solutions into real numbers. The function $V : X \to \mathbb{R}$ is one-to-one; in particular, it has a unique maximizer at x^*. The task of the team is locating x^*. This is a demonstrable intellective task that can be carried out *disjunctively*: if one of the agents finds x^*, he can show its value $V(x^*)$ to his teammates and the task is accomplished. Following Marschak and Radner (1972), we assume that all agents in the team share the same objective and there are no frictions such as difficulties of communication among people.

Consider a simple example. The solution space is $X = \{1, 2, \ldots, n\}$, with $V(1) < V(2) < \ldots < V(n)$. Thus, each point k corresponds to its rank with respect to V. Each agent has access to the current candidate solution, which we normalize to be 1. The task of the team is to locate the optimal solution n, when all agents know only the worst available solution.

We represent the limited problem-solving ability of each agent i by a partition Π_i of X; see Rubinstein (1993) for a related approach. Suppressing momentarily subscripts, consider one agent. The mutually disjoint and exhaustive classes constituting the partition Π are called *blocks*. The agent can find the best solution within the block is working on, but he is impervious to the other blocks until they are disclosed to him by someone else. For instance, assume $n = 12$ and consider the partition of agent 1 given by $\Pi_1 = \{1, 2, 4|3, 5, 6|7, 8, 9|10, 11, 12\}$, formed by four blocks. (A vertical bar separates the blocks.) Given the candidate solution $x = 1$, the agent can only see the points 1, 2, 4 and thus will find the solution $x = 4$.

When two or more agents work together, they can pool their abilities and expand their search spaces. Continuing our example, suppose that there are 3 agents with the following partitions:

$$\Pi_1 = \{1, 2, 4|3, 5, 6|7, 8, 9|10, 11, 12\};$$
$$\Pi_2 = \{1, 3, 5, 7|2, 4, 8|6, 9, 11|10, 12\};$$
$$\Pi_3 = \{1, 2, 7|3, 9, 10, 11|4, 5, 6, 8|12\}.$$

All agents start at $x = 1$. Agent 1 (named Primus) finds 2 and 4; Agent 2 (Secunda) finds 3, 5, and 7; Agent 3 (Tertium) finds 2 and 7. If the agents interacts in a team where everybody can talk to everyone else, the team will find the optimal solution $x^* = 12$. For instance, Tertium may point out 7 to Primus, giving him access to the block $|7,8,9|$. Then Primus finds 8 and 9 for the team. Using 9, Secunda opens up to her block $|6,9,11|$ and can pass 11 back to Primus who discovers $x^* = 12$. Clearly, there are other interactions that may lead the team to the optimal solution. When X is finite, there is a simple characterization of the set of solutions that are jointly explored by a team under a complete architecture; see LiCalzi and Surucu (2012).

The final element of the model is that the partitions representing agents' problem-solving abilities are randomly (and independently) chosen according to some distribution. This is important to model the idea that, although it may be possible to have a qualitative ranking over agents' abilities, we cannot rule out that their individual performance over a specific problem may be quite different. For instance, when an agent's partition is made of very small blocks, in general he is unlikely to find the optimal solution; however, if the initial condition $x = 1$ and the optimal solution $x = n$ happens by chance to belong in the same block, even a lousy agent may solve the problem.

This paper relies on the urn model described in Section 5 of LiCalzi and Surucu (2012), generalized to allow for heterogeneity in agents' cognitive abilities. See Collevecchio and LiCalzi (2012). Given a space X with n solutions, we assume that the partition of each agent i contains at most m_i blocks. For each agent i, we randomly assign each point from X to one of his blocks m_i with equal probability. Each random draw is stochastically independent, both within and across agents. On average, fewer blocks imply that a block contains more points and thus an agent with a lower index m_i is more likely to explore a larger subset of X. Hence, the fewer the blocks, the higher the problem-solving ability of an agent (on average). We call the ratio $k_i = n/m_i$ the *cognitive ability* of an agent.

14.3 Comparison of organizational architectures

The complete architecture assumed in LiCalzi and Surucu (2012) lets agents freely communicate with each other. This paper studies three triadic organizational architectures, based on a different information structure modeled by a directed graph. An agent located at a node i of the graph can pass information to another agent located at j if there is an edge leading from i to j. Figure 14.1 displays the information structure for the complete network and for the three architectures considered in this paper.

The *line* architecture is the prototypical hierarchy where earlier agents have no access to the information generated by later agents: Primus feeds information to Secunda; then she forwards information to Tertium, who picks the solution on behalf of the team. The *star-in* architecture has Secunda and Tertium passing their information to Primus, who picks the solution for the team. This architecture is used in Rivkin

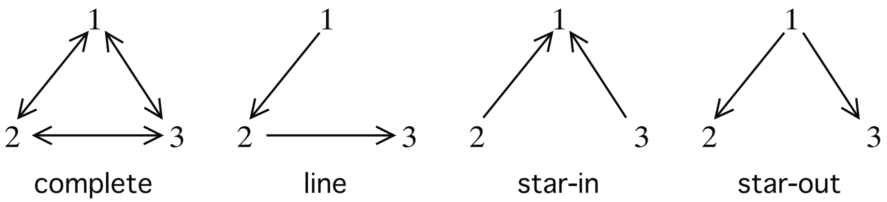


Fig. 14.1: Triadic organizational architectures.

and Siggelkow (2003) to typify the simplest possible notion of hierarchy in their agent-based study of the effects of organizational design on search. The *star-out* architecture has Primus feeding information to Secunda and Tertium, who both suggest their own solutions and then the best one is implemented. Typical situations that necessitate such architectures involves issues of confidentiality (lower ranks should not know the final decision) or speed of decision (cutting down on the rounds of communication allows for tighter deadlines).

Continuing with the example above, under a line architecture with initial condition $x = 1$, Primus feeds his block $|1,2,4|$ to Secunda. Using this, she can access $|1,2,3,4,5,7,8|$ and pass them to Tertium. He views $|1,2,3,4,5,6,7,8,9,10,11|$ and discovers the solution $x_l^* = 11$; the team has reviewed 11 elements out of 12, examining a fraction $f_l = 11/12 \approx 91.67\%$ of the solution space. By a similar argument, under the star-in architecture (with Primus on the receiving end), the team ends up examining $|1,2,3,4,5,6,7,8,9|$ and finds $x_i^* = 9$ after reviewing a fraction $f_i = 9/12 = 75\%$ of the solution space. Finally, under the star-out architecture, the team gains access to $|1,2,3,4,5,6,7,8|$, discovers $x_o^* = 8$ and explores a fraction $f_o = 8/12 \approx 67\%$ of the solution space. Recall that, under the complete architecture, the team discovers $x_c^* = 12$ and examines $f_c = 100\%$ of solution space.

In the example, we have $x_c^* > x_l^* > x_i^* > x_o^*$ as well as $f_c > f_l > f_i > f_o$. In general, however, the two rankings may not coincide even if the fraction f is positively correlated with the ability to discover better solutions. We measure the team performance by the fraction f of the solution space jointly examined by the team. Since f is a numerical variable normalized in $[0,1]$, this allows direct comparisons over solutions space with different cardinalities. (We assume that the starting point for the team is one of the n available solutions, so $f \geq 1/n$.) Moreover, under an obvious assumption of exchangeability, the span of the search carried out by the team exactly correlates with the probability of finding the optimal solution. Hence, the higher f, the higher the probability that the team is successful.

Roughly speaking, our first question is whether we can rank the triadic organizational architectures with respect to the quality of the team performance as measured by f. Clearly, the complete architecture dominates because it allows a larger set of communication links than any other architecture; hence, we focus on ranking the line, the star-in, and the star-out architectures. Our second question is concerned with the effects of how agents with different abilities are positioned within a given architecture.

14.4 Results for the exemplar

Since agents may have different cognitive abilities and their actual problem-solving abilities are randomized, reaching general conclusions requires a detailed experimental design. Our approach is to derive plausible claims from the analysis of an exemplar of the model, and then validate them by means of Monte Carlo simulations over an extensive range of parameters that define its variants. This latter work can be also interpreted as a robustness analysis for the representative model associated with the exemplar. This section describes the experimental design and state our claims. The following section reports on the design of the Monte Carlo simulations and on the validation of the claims.

14.4.1 The exemplar

We study three organizational architectures with three agents (line, star-in, star-out) represented by the directed graphs in Figure 14.1 over a representative model. We assume a solution space of $n = 100$ nodes. Each agent i has at most m_i blocks in his partition. For each agent, the pair (n, m_i) of parameters defines a random distribution over the partition that represents i's problem-solving ability.

A configuration of talents for the agents corresponds to a vector (m_1, m_2, m_3). For each architecture, we assume $m_1 + m_2 + m_3 = 216$ and study the team performance for all configurations that satisfy this constraint. E.g., this includes the symmetric case $m_1 = m_2 = m_3 = 72$ where all agents have identical cognitive ability $k_i = n/m_i = 3$, as well as an extreme case like $m_1 = 1$, $m_2 = 100$, $m_3 = 115$ where the first agent's cognitive ability is much higher than the others'.

Our choice of parameters for the exemplar is not arbitrary: we get some guidance from an asymptotic result proved in Collevecchio and LiCalzi (2012) for the complete network architecture in a different context from ours; see their Theorems 3 and 5. Let $m^* = \max\{m_1, m_2, m_3\}$ and $m_* = \min\{m_1, m_2, m_3\}$. Stated informally, they show for large n the condition $m^* < n/(8 \ln n)$ implies that (almost surely) a team arranged in a complete network architecture is always successful and thus $f \approx 1$; on the other hand, if $m_* > 3n/\ln n$ then (almost surely) the team cannot be always successful and thus $f < 1$. For $n = 100$, this suggests that $m^* \geq 3$ (and $m_* \geq 65$, respectively) approximates a necessary (sufficient) condition for having $f < 1$ in a complete network. The choice of parameters in our exemplar allows for a correction factor in order to compensate for the lower effectiveness of the other triadic architectures with respect to the complete network.

An exhaustive search over the exemplar allows us to evaluate how performance is affected by the agents' cognitive abilities and by their position in the architecture. Each simulation (independently) samples the agents' partitions and lets the agents search the solution space, recording the team performance as measured by the fraction f of the solution space surveyed by the team. For each architecture and each configuration, we run 1000 simulations and record the average performance $\bar{f}$. Hence, we have distinct datapoints for each architecture and each configuration.

14.4.2 Ranking architectures

Our first question concerns the performance of different triadic architectures when agents' abilities are not known. We begin with a visual description of the data. For each architecture, we represent each configuration as a point in the simplex $\{(m_1,m_2,m_3)\in\{1,2,\ldots,214\}: m_1+m_2+m_3=216\}$. Each point is colored into one of 10 shades of grey for each decile, increasing from white ($\bar{f}\leq 10\%$) to black ($\bar{f}>90\%$). Note that darker regions correspond to better performances.

Figure 14.2 summarizes the data for the three architectures. The vertices of the simplex represent Agent $i=1,2,3$ as we move counterclockwise from the bottom left corner. The three bisectors represent respectively the general directions in which m_i is decreasing and thus the cognitive ability of Agent i is increasing: f.i., the 30°-line coming out from the bottom-left corner represent the configurations where Primus is increasingly better while the two other agents are equally (and decreasingly) good.

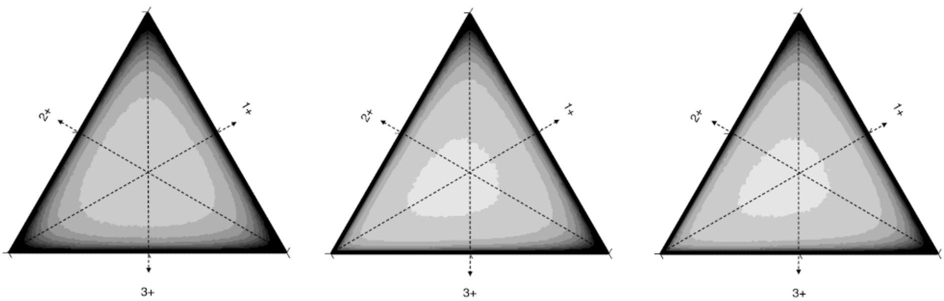
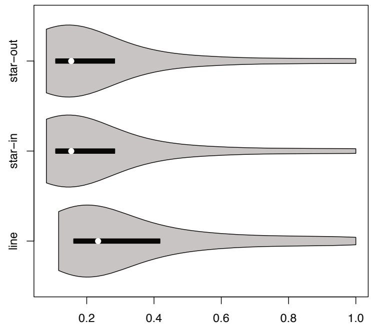


Fig. 14.2: Performance of the three architectures: line (left), star-in (center), star-out (right).

A common feature of the three simplices in Figure 14.2 is the following. As we move away from the center towards the edges, the performance measure increases: making the cognitive ability of an agent sufficiently large allows the team to explore the entire search space. More importantly, visual inspection suggests that, *ceteris paribus*, a configuration of talents is more effective under the line architecture than under either star architecture. In particular, out of 23005 possible configurations, we find that the line architecture performs no worse than either star architectures in 22989 cases (99.93%), while the star-in architecture wins/ties/loses over the star-out in 11140/690/11175 cases (48.42%, 3%, 48.58%).

To probe this further, we offer a second representation of the data in Figure 14.3. Based on the `vioplot` package in **R**, the picture on the left juxtaposes the box plot and the kernel density plot for each architecture. The box plot depicts the interquartile range in black and the median as a white point, while the grey area surrounding it visualizes the kernel density. Next to this, we give the main descriptive statistics for the (average) performance of each architectures over all the configurations in the simplex.



	min	lower quartile	median	upper quartile	max	mean	std. dev.
star-out	.0799	.1075	.1540	.2830	1	.2478	.2226
star-in	.0801	.1075	.1540	.2834	1	.2478	.2226
line	.1165	.1612	.2339	.4178	1	.3328	.2372

Fig. 14.3: Boxplot, kernel densities and descriptive statistics for the three architectures.

The evidence points to a higher performance of the line architecture with respect to either star architecture when agents are randomly assigned to their positions within an architecture, while there seems to be no apparent difference between the two star architectures. We summarize these observations in two claims.

Claim 1a. The line architecture performs better than either star architecture.

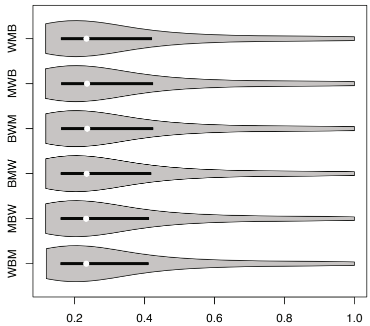
Claim 1b. The star architectures perform similarly.

Consistent with such claims, we find that for our samples the null hypothesis that the line architecture performs no better than either star architectures is rejected by a two-sample Wilcoxon test (also known as Mann-Whitney test) for any practical level of confidence. (Here and in the following, by a *practical* level of confidence we mean a p-value lower than 10^{-9}.) Similarly, the null hypothesis that the two star architectures perform differently is rejected by a Mann-Whitney test for any practical level of confidence. Under our two claims, the ranking of the triadic organizational architectures with respect to team performance is: complete $>$ line $>$ star-in $\sim$ star-out.

14.4.3 Placement within architectures

Our second question concerns the placement of differently talented individuals within an organizational architecture. For the complete network, where communication is unrestricted and everybody gets access to all available information, this is clearly irrelevant. The line architecture, instead, progressively discloses more information as we move up in the hierarchy. Thus, one may legitimately expect that placement matters: if a more talented person acts first, the following agent has access to more information. The countervailing effect is that a less talented agent in second place has an inferior ability to exploit this information. As the 3-side symmetry in the simplex on top of Figure 14.2 suggests, the net effect in our model makes placement almost irrelevant for the team performance. What is gained by moving talented people upfront is lost by missing their abilities downstream.

We label the six possible placements of agents in a line architecture as follows. Given a configuration (m_1, m_2, m_3), the agent with the lowest value of m_i has the highest cognitive ability and we denote him by B for "Best"; similarly, the agent with second and third lower values of m_i are respectively denoted M for "Medium" and W for "Worst". Note that the labels refers to m_i: depending on the realization of the partition that is randomly drawn, an agent with a higher cognitive ability may end up being less effective. The string WBM indicates that W sits at the source, B in the middle, and M at the sink of the line architecture. Figure 14.4 reports the descriptive statistics using the same format as Figure 14.3. To avoid ambiguities when discussing placement, we always remove the 319 configurations (out of 23005) located on the three bisectors where two or more agents share the same value of m_i.



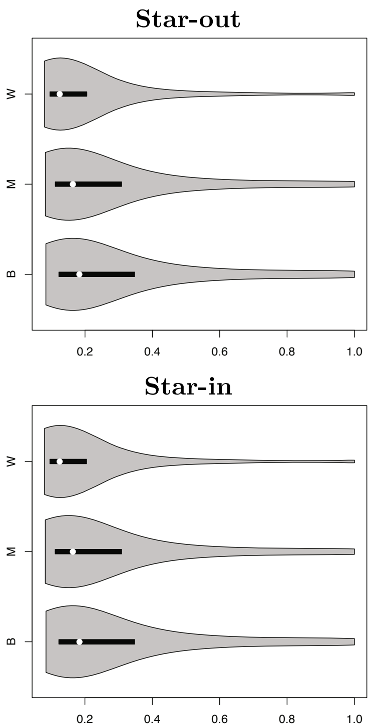
	min	lower quartile	median	upper quartile	max	mean	std. dev.
WMB	0.1174	0.1618	0.2342	0.4203	1	0.3334	0.2370
MWB	0.1165	0.1624	0.2357	0.4238	1	0.3367	0.2403
BWM	0.1178	0.1623	0.2363	0.4243	1	0.3366	0.2402
BMW	0.1177	0.1614	0.2347	0.4187	1	0.3335	0.2370
MBW	0.1173	0.1611	0.2332	0.4118	1	0.3296	0.2334
WBM	0.1192	0.1608	0.2336	0.4108	1	0.3295	0.2333

Fig. 14.4: Boxplot, kernel densities and descriptive statistics for different line placements.

The six distributions look very similar. A Kruskal-Wallis test over the six set of data points yields a p-value of .8816 and hence fails to reject the null hypothesis that they share the same location parameter. We are thus led to the following claim.

Claim 2a. Team performance in a line is not affected by the placement of agents.

Moving on to the star architectures, the shape of the colored regions in the simplices is asymmetric with respect to the Agent 1 sitting respectively at the sink (for the star-in architecture) or at the source (for the star-out architecture). Following the directions of increasing cognitive ability, the performance of the network degrades much faster along the diagonal springing from the bottom-left corner than from the other two. This suggests that the team performance is positively affected by the cognitive ability of the agent placed in the central position of a star architecture. This is supported by the descriptive statistics in Figure 14.5, where B denotes the case where the agent with highest cognitive ability sits at the center and the analogous convention applies for M and W.



	min	lower quartile	median	upper quartile	max	mean	std. dev.
			Star-out				
W	.0799	.0959	.1250	.2055	1	.1965	.1843
M	.0824	.1119	.1641	.3090	1	.2629	.2314
B	.0839	.1225	.1833	.3473	1	.2851	.2381
			Star-in				
W	.0801	.0961	.1250	.2051	1	.1965	.1843
M	.0824	.1118	.1642	.3090	1	.2630	.2314
B	.0845	.1226	.1839	.3476	1	.2850	.2382

Fig. 14.5: Boxplot, kernel densities and descriptive statistics for different star placements.

For either star architecture, we perform two distinct Mann-Whitney tests: one between B and M, and the other between M and W. In all cases, the test rejects the null hypotheses that the first placement is not better than the second for any practical level of confidence. We are thus led to the claim that the placements in either star architecture can be ranked with respect to the cognitive ability of the central agent as follows: B > M > W.

Claim 2b. Team performance in a star positively relates to the cognitive ability of the agent in the central position.

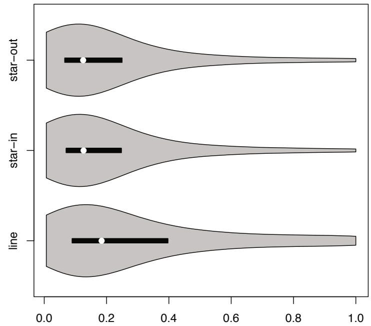
14.5 Validation and robustness

This section validates the claims advanced in the previous one, providing a robustness analysis by means of Monte Carlo simulations. With respect to the exemplar, we assume that the four parameters n, m_1, m_2, m_3 are random variables, independently distributed. More specifically, we assume that n is drawn according to a uniform distribution on $[50, 150]$ and each m_i according to a uniform distribution on $[2, 142]$ for $i = 1, 2, 3$. The expected values for this distribution match those chosen for our exemplar. Note that, while $E(n) = 100$ and $E(\sum_i m_i) = 72$, the stochastic independence between m_i's implies that the event $\{\sum_i m_i = 72\}$ occurs with low probability. This specification adds a lot of noise around our exemplar. We separately examine the validity of our claims.

Our first set of claims concerns the effectiveness of the different triadic architectures, *regardless* of the placement of the agents. Intuitively, this corresponds to the case where a principal who ignores the agents' cognitive abilities can select the architecture but not the arrangement of agents within it. We formalize this situation by testing the performance of each architecture under the assumption that any distinguishable placement of the agents is equally likely. The line architecture has six of these, while each of the stars has three.

We run 100,000 Monte Carlo simulations, testing behavior for random configurations of the four parameters across different architectures. Each simulation randomly picks values for n and m_i's and accordingly draws a partition for each agent. Keeping all of these choices fixed, we select a placement for the agents using a uniform distribution over each architecture. Then we compute the performance measure f for each architecture, so that each round k generates a triple $\mathbf{f}_k = (f_l, f_i, f_o)$. The only differences in the generation of the elements in the triple are the architecture and the (random) placement of the agents. That is, the noise is only due to the lack of information about the cognitive abilities of agents in different position. The descriptive statistics are given in Figure 14.6 in the usual format.

Claims 1a and 1b respectively posit $f_l > \max\{f_i, f_o\}$ and $f_i \approx f_o$. The descriptive statistics seem consistent with both claims, although one may point out that the distribution for star-out seems to be slightly more spread out than the distribution for star-in. Concerning Claim 1a, the Mann-Whitney test rejects the null hypotheses that f_l is not better than $\max\{f_i, f_o\}$ for any practical level of confidence. Concerning Claim 1b, the Mann-Whitney test rejects the null hypotheses that f_i and f_o are similar for any practical level of confidence, suggesting that the difference in location parameters between star-in and star-out may be close to .002. We conclude that



	min	lower quartile	median	upper quartile	max	mean	std. dev.
star-out	.0067	.0656	.1250	.2500	1	.2031	.2135
star-in	.0067	.0700	.1263	.2479	1	.2033	.2091
line	.0067	.0893	.1842	.3973	1	.2866	.2684

Fig. 14.6: Boxplot, kernel densities and descriptive statistics for the three architectures.

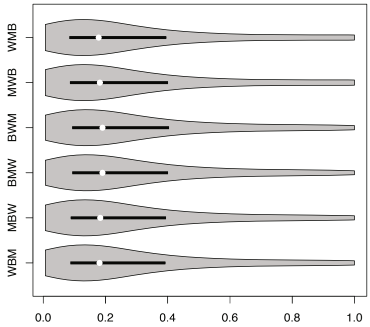
Claim 1a is confirmed, while Claim 1b is not unqualifiedly true: however, differences are tiny and seem to bear little economic significance.

Our second claim concerns the impact of different placements on each triadic architectures. Intuitively, this is the case where, given the architecture, the principal knows the agents' cognitive abilities and wants to pick the arrangement that generates the best performance. Here, we need to separately test the three architectures. For each architecture we run 100,000 Monte Carlo simulations using the same setup described above except that in any round we compute the performance measure f for each distinguishable arrangement within each architecture, so that a round generates a 6-tuple for the line architecture and a triple for each star. Given the architecture, the only difference in the generation of the elements in an n-tuple lies in the placement of the agents.

We begin with the line architecture, whose descriptive statistics are given in Figure 14.7. Claim 2a posits that any of the six distinguishable arrangements of agents in the line architecture yields (approximately) the same performance. A Kruskal-Wallis test over the six set of data points, however, rejects the null hypothesis that they share the same location parameter. On the other hand, if we pair series who have agents with the same relative ability (BWM/BMW, MWB/MBW, WMB/WBM), all three Mann-Whitney tests find statistically significant evidence (p-value: .546, .569, .063) that such pairs are similarly distributed. It is possible that in general the cognitive ability of the leading agent in a line may matter, but at this stage we can only conclude that there is some minor effect differentiating the placements.

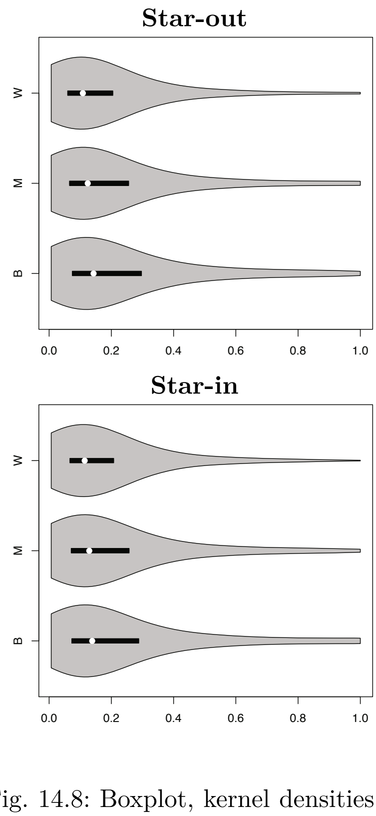
We now move to consider the star architectures. The descriptive statistics are in Figure 14.8.

Claim 2b posits that the cognitive ability of the agent located in the middle positively affects the team performance. This is backed up by the descriptive statistics. Moreover, for either star architecture, two distinct Mann-Whitney tests (one between B and M, the other between M and W) reject the null hypotheses that the first place-



	min	lower quartile	median	upper quartile	max	mean	std. dev.
WMB	0.0067	0.0849	0.1781	0.3947	1	0.2871	0.2780
MWB	0.0067	0.0866	0.1818	0.4000	1	0.2903	0.2792
BWM	0.0067	0.0938	0.1905	0.4035	1	0.2892	0.2640
BMW	0.0067	0.0941	0.1905	0.4000	1	0.2867	0.2597
MBW	0.0067	0.0892	0.1833	0.3932	1	0.2837	0.2644
WBM	0.0067	0.0882	0.181	0.3918	1	0.2831	0.2658

Fig. 14.7: Boxplot, kernel densities and descriptive statistics for different line placements.



	min	lower quartile	median	upper quartile	max	mean	std. dev.
			Star-out				
W	0.0067	0.0595	0.1084	0.2048	1	0.1697	0.1783
M	0.0067	0.0654	0.1241	0.2557	1	0.2092	0.2248
B	0.0067	0.0745	0.1429	0.2969	1	0.2304	0.2297
			Star-in				
W	0.0067	0.0667	0.1143	0.2075	1	0.1696	0.1618
M	0.0067	0.0709	0.129	0.2571	1	0.2084	0.2123
B	0.0067	0.0729	0.1385	0.2881	1	0.2312	0.2408

Fig. 14.8: Boxplot, kernel densities and descriptive statistics for different star placements.

ment is not better than the second for any practical level of confidence. Thus, we conclude that Claim 2b is confirmed.

To summarize, we found full support for two statements: the line architecture performs better than either of the star architectures and the cognitive ability of the central agent in either star architecture is positively related to team performance.

References

1. Collevecchio A, LiCalzi M (2012), The probability of nontrivial common knowledge. Working Paper 6/2011, revised March 2012, Dept. of Management, Università Ca' Foscari Venezia
2. LiCalzi M, Surucu O (2012), The power of diversity over large solution spaces. Management Science, forthcoming
3. Marschak J, Radner R (1972), Economic Theory of Teams. Yale University Press, New Haven, CT
4. Mihm J, Loch C.H., Wilkinson D., Huberman B.A. (2010), Hierarchical structure and search in complex organizations. Management Science 56, 831–848
5. Rivkin J.W., Siggelkow N. (2003), Balancing search and stability: Interdependencies among elements of organizational design. Management Science 49, 290–311
6. Rubinstein A (1993), On price recognition and computational complexity in a monopolistic model, Journal of Political Economy 101, 473–484