THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2013-14

Predicting the Performance of Teams of Bounded Rational Decision-makers Using a Markov Chain Model

Jeffrey Herrmann

Predicting the Performance of Teams of Bounded Rational Decision-makers Using a Markov Chain Model

Jeffrey W. Herrmann
A. James Clark School of Engineering
University of Maryland
College Park, MD 20742
301-405-5433
jwh2@umd.edu

Abstract

In practice, when faced with a complex optimization problem, teams of human decision-makers often separate it into subproblems and then solve each subproblem instead of tackling the complete problem. It would be useful to know the conditions in which separating the problem is the superior approach and how the subproblems should be assigned to members of the teams. This paper describes a mathematical model of a search that represents a bounded rational decision-maker ("agent") solving a generic optimization problem. The agent's search can be modeled as a discrete-time Markov chain, which allows one to calculate the probability distribution of the value of the solution that the agent will find. We compared the distributions generated by the model to the distribution of results from searches of solutions to traveling salesman problems. Using this model, we evaluated the performance of two- and three-agent teams who used different solution approaches to solve generic optimization problems. In the "all-at-once" approach, the agents collaborate to search the entire set of solutions in a sequential manner: the next agent begins where the previous agent stopped. In the "separation" approach, the agents separate the problem into two subproblems: (1) find the best set of solutions, and (2) find the best solution in that set. The results show that teams found better solutions using separation when high-value solutions are less likely. Using the all-at-once approach yielded better results when the values were uniformly distributed. The optimal assignment of subproblems to teams also depended upon the distribution of values in the solution space.

Introduction

In practice, human decision-makers often separate a complex optimization problem into subproblems and then solve each subproblem instead of tackling the complete problem. This approach is a natural strategy given the constraints that human decision-makers have. It is especially convenient in a team setting: different subproblems can be assigned to different members of the team.

If human decision-makers were able to optimize, then separating a problem into subproblems would usually lead to solutions that are inferior to those found by solving the problem all-at-once (only in certain conditions will the optimal solutions to the subproblems form an optimal solution to the complete problem).

It is well-known, however, that real-world decision-makers cannot optimize because of limits on their problem-solving capacity. This concept is known as "bounded rationality" (Simon, 1997a). Bounded rationality reflects the observation that, in most real-world cases, decision-makers have limited information and limited computational capabilities for finding and evaluating alternatives and choosing among them (March and Simon, 1993; Simon, 1997b; Gigerenzer *et al.*, 1999). A decision-maker cannot perfectly evaluate the consequences of the available choices. This prevents complete and perfect optimization.

The study described in this paper was motivated by the following questions: (1) when is separating a better approach than solving the problem all-at-once? (2) what is the best way to assign subproblems to the members of a team?

As part of an ongoing study of the effectiveness of separation by human decision-makers, this paper presents the results of a study that considered a class of generic optimization problems, modeled teams of bounded rational decision-makers, and evaluated the quality of the solutions that these teams found using different solution approaches. The study was intended to provide insights into the relationship between the distribution of the solutions and the relative performance of the solution approaches.

A novel feature of this study is our Markov chain model of the search of a bounded rational decision-maker. This model allows one to calculate, without resorting to computational simulation, the probability distribution of the value of the solution that the decision-maker will find.

The remainder of the paper proceeds by reviewing the concept of bounded rationality, defining separations, and describing the class of optimization problems that we studied. The paper then introduces the Markov chain model, describes the instances analyzed, and presents the results of the study before concluding with some insights gained from this work.

Bounded Rationality

Satisficing and fast and frugal heuristics are two models of bounded rationality (Gigerenzer *et al.*, 1999). Bounded rational decision-makers may search until they find something that meets their requirements (satisficing), or they may use fast and frugal heuristics that search a limited set of objects and information and make choices using rules that are easy to compute (and therefore quick). This study considered only this second type of bounded rationality.

When tackling a complex optimization problem, the decision-maker needs to determine values for various aspects of the solution. That is, he must make a decision. In practice, he does not have complete information about the available alternatives and their impact on overall performance. Thus, he must employ some process for generating and evaluating alternatives.

March and Simon (1993) describe the general characteristics of human problem-solving in organizational decision-making. The first characteristic is that making a complex decision involves making a large number of small decisions. The second characteristic is that problem-solving has a hierarchical structure in which solving any problem goes through phases that, in turn, require solving more detailed subproblems. The general concept of separation is related to these two characteristics. The third characteristic is that problem-solving consists of searching for possible solutions (cf. Nutt, 2005). The fourth characteristic is that problem-solving includes screening processes that evaluate the solutions that are found. The fifth characteristic is that problem-solving has not only random components (such as finding and evaluating solutions) but also a procedural structure that allows it to yield good solutions. The proposed model of a bounded rational designer is motivated by these last three characteristics.

Wang and Chiew (2008) described human problem solving as a higher-layer cognitive process that can be considered as a search process, though it requires other cognitive processes such as abstraction, analysis, synthesis, and decision-making. Their model of a generic problem solving process is a search that iteratively generates and evaluates potential solutions.

Thus, we conclude that search is a valid model of bounded rational decision-making.

Herrmann (2010) took a similar approach but considered separations of a specific engineering design optimization problem as models of progressive design processes. His results indicated that well-designed progressive design processes are the best way to generate profitable product designs.

This paper considers a very different problem but addresses the same issue: how does the separation affect the quality of the solution? It also considers the best way to allocate the resources of a team of decision-makers to the subproblems in the separation.

An important aspect of bounded rationality is that the resources and time available for problem-solving are limited. Consequently, the proposed model of a bounded rational decision-maker incorporates limits that will constrain the amount of time available for the search (and the quality of the solution found).

A great variety of models, including agent-based simulation models, have been used to study teams in general (cf. Crowder *et al*., 2012). Teams of bounded rational decision-makers who perform the same task repeatedly were studied by Boettcher and Levis (1982, 1983) and Levis and Boettcher (1983), who modeled the decision-making process of each person as a two-stage process that incorporated situation assessment and response selection. The condition of bounded rationality is modeled as a constraint on the rate at which each person can process information. The decision-making process was later extended to include information fusion, task processing, and command interpretation stages and represented using Petri nets (Levis, 2005).

Gurnani and Lewis (2008) studied collaborative, decentralized design processes in which the models of the individual decision-makers (the designers) were based on the ideas of bounded rationality. In their model, the value chosen by each designer was determined by randomly sampling from a distribution around the (locally) "optimal" solution. This model was meant to represent the mistakes that designers make due to bounded rationality. Their results showed that incorporating bounded rationality led to more desirable solutions in a collaborative, decentralized design process in which the designers had different objectives and no way to coordinate their activities.

Herrmann (2010) presented a method for assessing the quality of a product design process by measuring the profitability of the product that the process generates. Because design decision making is a type of search, the method simulated the choices of a bounded rational designer for each subproblem using search algorithms. The searches, which were limited to a fixed number of iterations, had random components (either randomly selecting a solution or randomly moving to a point near the existing solution) and a procedural structure to keep track of the best solution found. The results showed that decomposing a problem into subproblems yields a better solution than solving the entire problem at once when bounded rational search is employed. Herrmann (2012) described a study in which different approaches for separating the Inventory Slack Routing Problem (a complex vehicle routing problem) were simulated. Again, a random search was used to simulate how a bounded rational human decision-maker would solve each subproblem. The results showed that the structure of the separation and the objectives used in each subproblem significantly affected the quality of the solutions that are generated. Additional details about the Inventory Slack Routing Problem can be found in the report by Montjoy and Herrmann (2012).

Hong and Page (2004) modeled problem-solvers of limited ability as searches that they called "heuristics," and each heuristic searched a finite set of solutions until it could not find a better solution. Thus, the problem-solver is conducting a type of hill-climbing search. Hong and Page studied teams of such problem-solvers and identified conditions under which a diverse set of problem-solvers will likely perform better than a team of high-performing individuals. In their work, the problem-solvers searched a finite set of solutions (the size ranged from 200 to 10,000). A problem-solver was characterized by how many and which points near the current solution it would consider. Essentially, different problem-solvers had different neighborhood definitions.

The research described in the current paper started by modeling decision-makers like the problem-solvers that Hong and Page considered, but the search spaces used are somewhat different. This research did not measure the value of diversity; instead it studied how separating the problem and allocating the team's resources affect the quality of the solutions that teams of problem-solvers generated.

Separations

A separation is a process that solves a sequence of subproblems. A large problem is divided into subproblems, and the solution to one subproblem provides the inputs to one or more subsequent subproblems. Note that the separation does not have to be a simple sequence of subproblems; it may have subproblems that are solved in parallel at places. A given separation specifies a partial order in which the subproblems are solved. A different order of subproblems would be a different separation and would lead to a different solution.

The subproblems' objective functions are surrogates for the original problem's objective function. These surrogates come from substituting simpler performance measures that are correlated with the original one, eliminating components that are not relevant to that subproblem, or from removing variables that will be determined in another subproblem.

Problem Formulation

This study considered teams of bounded rational decision-makers (problem-solvers, or agents) and different approaches for solving a generic class of optimization problems.

An optimization problem requires finding an optimal solution in a space of many solutions. In this study, every point in the solution space has a value in the set $\{1, \ldots, N\}$, where the value N is the optimal value, and a point with value i is better than a point with value j if and only if $i > j$. The number of points in the solution space is not relevant.

The points in the solution space are grouped into "sets." The value of a set is the average value of the points in the set. In engineering design, a set corresponds to a conceptual design, and the points in the set correspond to the detailed designs that are associated with that conceptual design. For instance, a set may be the basic path of a highway that is to be built between two cities, while the points in that set are the specific roadways that follow that general path.

A team of two agents can solve this type of problem using two approaches. In the "all-at-once" approach, the first agent searches the entire space of solutions until he stops at a locally optimal solution, and then the second agent begins searching from that point until he stops at a locally optimal solution (because the second agent begins a new search, a point that is locally optimal for the first agent may not be locally optimal for the second agent; this will discussed in more detail in the next section).

The agents can also separate the problem into two subproblems that are solved sequentially. The first subproblem is to find the best set, and the second subproblem is to find the best point within that set. The first agent searches the sets until he stops at a locally optimal set, and then the second agent searches that set of points until he stops at a locally optimal solution.

If the team consists of three agents, then they can collaborate in the all-at-once approach; the third agent can continue the search by starting where the second agent stopped. If they separate the problem, the third agent can help the first agent by continuing the search of the sets or help the second agent by continuing the search of the points in that set.

Modeling Agents Searching

In the generic optimization problem, a problem-solver ("agent") must search a space of solutions (or sets); the agent wants to find the best point (set), that is, the one that has the largest value, but he is limited, however, by the limitations of the search strategy. In this study we assumed that the agent's search had the following characteristics: The agent begins by randomly selecting a point in the solution space. From the current point, the agent checks another randomly selected point and accepts the new point if it is better (has a greater value) than the current point. The agent's search is governed by a parameter K. If the agent checks K consecutive points that are not better than the current point, then the agent stops his search. (Increasing K will increase the effort spent searching and will increase the expected value of the solution found.) The search for a set works in the same way. There are W types of sets and an ordering over the values of the sets such that a set of type a is better than a set of type b if $a > b$.

The quality of the solution found by the agent is a random variable that depends upon K and the distribution of values in the search space. Let $p(x)$ be the probability that a randomly selected point will have the value x, for all $x \in \{1, \ldots, N\}$. Let $v(t)$ be the value of the agent's current point after checking t points (where $t = 0$ corresponds to the initial point).

The stochastic process $v(t)$, $t = 1, 2, \ldots$, is generated by a Markov chain. The state of the Markov chain is the value of the agent's current point and the number of points checked since accepting that point. Let $\left(v(t), n(t)\right)$ be the state at step t, where $v(t)$ is the value of the agent's current point and $n(t)$ equals the number of points checked since accepting that point. The number of states is finite. The states are the elements of $\{1, \ldots, N\} \times \{0, \ldots, K\}$. All states that have $n(t) = K$ are absorbing states because the search stops when this occurs. All of the other states are transient. Thus, the distribution of $v(t)$ converges to the distribution over the absorbing states $(1, K)$, …, (N, K).

The probability transition matrix for this Markov chain can be described as follows:

At $t = 0$, $n(0) = 0$, and $P\{v(0) = x\} = p(x)$.

Let $P(w, j, x, i) = P\{v(t) = x, n(t) = i \mid v(t-1) = w, n(t-1) = j\}$.

$P(w, j, x, 0) = p(x)$ for $w = 1, \ldots, N-1$, $x = w+1, \ldots, N$, $j = 0, \ldots, K-1$

$P(w, j, w, j+1) = p(1) + \cdots + p(w)$ for $w = 1, \ldots, N$, $j = 0, \ldots, K-1$

$P(w, K, w, K) = 1$ for $w = 1, \ldots, N$.

$P(w, j, x, i) = 0$ otherwise.

Consider, for example, the problem with N = 10 and $p(x) = (11-x)/55$ for $x = 1, \ldots, 10$. The values of $p(x)$ are approximately (0.1818, 0.1636, 0.1455, 0.1273, 0.1091, 0.0909, 0.0727, 0.0545, 0.0364, 0.0182).

Let P be the probability transition matrix for this Markov chain.

$$P = \begin{bmatrix} A & B & 0 \\ A & 0 & B \\ 0 & 0 & I \end{bmatrix}, \text{ where}$$

$$A = \begin{bmatrix} 0 & \frac{9}{55} & \frac{8}{55} & \cdots & \frac{2}{55} & \frac{1}{55} \\ 0 & 0 & \frac{8}{55} & \cdots & \frac{2}{55} & \frac{1}{55} \\ 0 & 0 & 0 & \cdots & \frac{2}{55} & \frac{1}{55} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{55} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \text{ and}$$

$$B = \begin{bmatrix} \frac{10}{55} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{19}{55} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{27}{55} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{54}{55} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \text{ and}$$

I is the 10-by-10 identity matrix.

The probability distribution for $v(t)$ converges to (0.0060, 0.0237, 0.0520, 0.0877, 0.1255, 0.1573, 0.1739, 0.1671, 0.1329, 0.0739). Although the smallest values are the ones that are most likely to occur in the solution space, moderate values (like 6, 7, and 8) are the ones that are most likely to be found in the search. The probability of the search returning a solution with a value greater than or equal to 7 equals 0.5477, although the probability that a randomly selected point has a value greater than or equal to 7 equals only 0.1818.

Comparison to Search Results

To generate some evidence that this model is a reasonable model of this type of search, we randomly generated instances of the traveling salesman problem (TSP), constructed and evaluated the corresponding Markov chain models, and used random searches to find solutions to these instances. We then evaluated the similarity of the distributions generated by the Markov chain model and the distributions of the search results using a chi-squared goodness of fit test.

In particular, we generated TSP instances with 12 sites located on a two-dimensional square. The coordinates of the sites were randomly drawn according to a uniform distribution on the interval [0, 100]. The Euclidean distance was used as the distance measure. The 12th site was fixed as the last site in any TSP solution.

The separation involved clustering the 12 sites into three clusters of four sites (the third cluster always included the 12th site). Thus, each cluster includes (4!)(4!)(3!) = 3,456 routes. When creating the Markov chain model, every cluster was evaluated by finding the average length of the routes in that cluster. In the searches, a cluster was evaluated approximately by finding the average length of 20 randomly-selected routes in that cluster.

For the Markov chain model, the route lengths (and cluster averages) were discretized into approximately 50 values (the exact number depended upon the distribution of the route lengths).

The searches include all-at-once searches with one, two, and three agents and three versions of the separation: after one agent searches for a cluster and a second searches the routes in that cluster (this we call “1-1”); after one agent searches for a cluster and two other agents search the routes in that cluster (this we call “1-2”); and after two agents search for a cluster and a third agent searches the routes in that cluster (this we call “2-1”).

In general, chi-squared tests showed that the distributions generated by the Markov chain model and the distributions of the search results were similar. Table 1 and Figure 1 show the results for one 12-site TSP instance. Each search was run 1000 times, and the route lengths were converted into the discrete values used in the Markov chain analysis. For each of the six searches, because the chi-squared statistic is less than the critical value, we do not reject the hypothesis that the search results are distributed according to the distribution that the Markov chain generates. Figure 1 shows the distributions for the Separation 2-1 search for the same instance and the distribution of the values for all of the routes for this instance. (Although the TSP is a minimization problem, in these distributions, a larger value represents better route.)

Table 1. Chi-squared test statistics for one 12-site TSP instance (n = 1000).

Search	Test statistic	Degrees of freedom	Critical value (alpha = 0.10)
All-at-once one agent	26.573	41	52.949
All-at-once two agents	25.724	38	49.513
All-at-once three agents	36.421	35	46.059
Separation 1-1	35.373	42	54.090
Separation 1-2	36.290	38	49.513
Separation 2-1	29.895	40	51.805

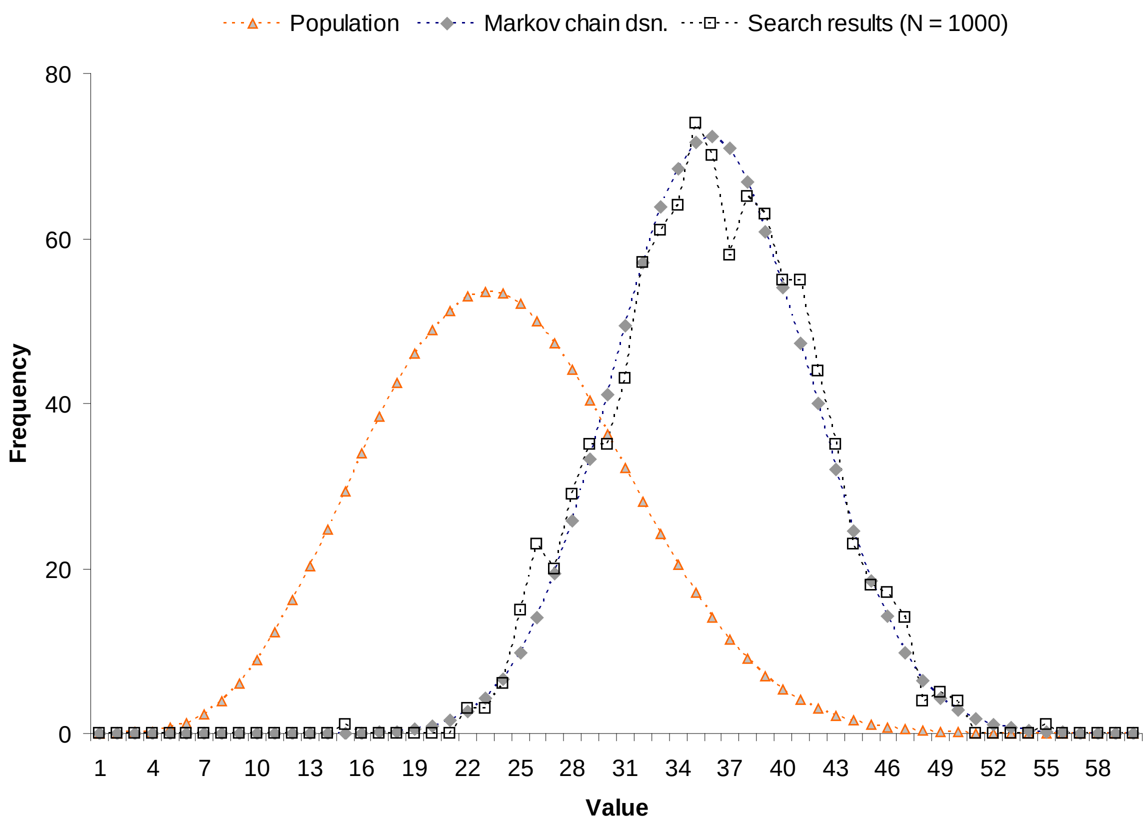

Figure 1. The distributions of values in a 12-site TSP instance.

Instances

To compare the performance of different separations, we generated different distributions for the sets and populations according to the following scheme.

Let $r(s,v)$ be the probability of finding a point with value *v* when searching in a type *s* set. Let $q(s)$ be the probability of finding a type *s* set when searching for a set.

First, we considered a space with 50 types of sets and 99 values. Within this space, there were two different distributions over the values in a set. In the first, which we called "uniform," all values were equally likely in every set. That is, $r_u(s,v) = \frac{1}{99}$ for all *s* and *v*.

In the second, which we called "triangular," only some values were possible, but the possible values were equally likely in every set. In particular, for $s = 1,\ldots,50$, $r_t(s,v) = \frac{1}{50}$ for $v = s,\ldots,s+49$ and 0 otherwise.

There were two distributions over the sets. In the first ("uniform sets"), all of the sets were equally likely: $q_u(s) = \frac{1}{50}$ for all *s*. In the second ("skewed sets"), the better sets are less likely: $q_k(s) = (51-s)/1275$ for all *s*.

From these distributions, we created 11 cases (A to K). Each case was an interpolation between the distributions over the values in a set or the distributions over the sets.

Case	Distributions over the values in a set	Distribution over the sets
A	$r(s,v) = \frac{9}{10} r_u(s,v) + \frac{1}{10} r_t(s,v)$	$q(s) = q_u(s)$
B	$r(s,v) = \frac{6}{10} r_u(s,v) + \frac{4}{10} r_t(s,v)$	$q(s) = q_u(s)$
C	$r(s,v) = \frac{3}{10} r_u(s,v) + \frac{7}{10} r_t(s,v)$	$q(s) = q_u(s)$
D	$r(s,v) = r_t(s,v)$	$q(s) = q_u(s)$
E	$r(s,v) = r_t(s,v)$	$q(s) = \frac{3}{4} q_u(s) + \frac{1}{4} q_k(s)$
F	$r(s,v) = r_t(s,v)$	$q(s) = \frac{1}{2} q_u(s) + \frac{1}{2} q_k(s)$
G	$r(s,v) = r_t(s,v)$	$q(s) = \frac{1}{4} q_u(s) + \frac{3}{4} q_k(s)$
H	$r(s,v) = r_t(s,v)$	$q(s) = q_k(s)$
I	$r(s,v) = \frac{3}{10} r_u(s,v) + \frac{7}{10} r_t(s,v)$	$q(s) = q_k(s)$
J	$r(s,v) = \frac{6}{10} r_u(s,v) + \frac{4}{10} r_t(s,v)$	$q(s) = q_k(s)$
K	$r(s,v) = \frac{9}{10} r_u(s,v) + \frac{1}{10} r_t(s,v)$	$q(s) = q_k(s)$

Next, we considered a space with 15 types of sets and 120 values. Within this space, there were three different distributions over the values in a set. In the first, which we called "uniform," all values were equally likely in every set. That is, $r_u(s,v) = \frac{1}{120}$ for all *s* and *v*. In the second, which we called "triangular," only some values were possible, but the possible values were equally likely in every set. In particular, for $s = 1,\ldots,15$, $r_t(s,v) = \frac{1}{50}$ for $v = 5(s-1)+1,\ldots,5(s-1)+50$ and 0 otherwise. In the third, which we called "split," only some values were possible, but the possible values were equally likely in every set. In particular, $r_s(s,v) = 1/(16-s)$ for $v = \frac{1}{2}(32-s)(s-1)+1,\ldots,\frac{1}{2}(31-s)s$ and 0 otherwise.

In this space, there was only one distribution over the sets. The better sets are less likely: $q_k(s) = (16-s)/120$ for all *s*.

From these distributions, we created 11 more cases (L to V). Each case was an interpolation between the distributions over the values in a set.

Case	Distributions over the values in a set	Distribution over the sets
L	$r(s,v) = \frac{9}{10} r_u(s,v) + \frac{1}{10} r_t(s,v)$	$q(s) = q_k(s)$
M	$r(s,v) = \frac{6}{10} r_u(s,v) + \frac{4}{10} r_t(s,v)$	$q(s) = q_k(s)$
N	$r(s,v) = \frac{3}{10} r_u(s,v) + \frac{7}{10} r_t(s,v)$	$q(s) = q_k(s)$
O	$r(s,v) = r_t(s,v)$	$q(s) = q_k(s)$
P	$r(s,v) = \frac{3}{4} r_t(s,v) + \frac{1}{4} r_s(s,v)$	$q(s) = q_k(s)$
Q	$r(s,v) = \frac{1}{2} r_t(s,v) + \frac{1}{2} r_s(s,v)$	$q(s) = q_k(s)$
R	$r(s,v) = \frac{1}{4} r_t(s,v) + \frac{3}{4} r_s(s,v)$	$q(s) = q_k(s)$
S	$r(s,v) = r_s(s,v)$	$q(s) = q_k(s)$
T	$r(s,v) = \frac{3}{10} r_u(s,v) + \frac{7}{10} r_s(s,v)$	$q(s) = q_k(s)$
U	$r(s,v) = \frac{6}{10} r_u(s,v) + \frac{4}{10} r_s(s,v)$	$q(s) = q_k(s)$
V	$r(s,v) = \frac{9}{10} r_u(s,v) + \frac{1}{10} r_s(s,v)$	$q(s) = q_k(s)$

Computational Experiments

The purpose of the computational experiments was to compare the performance of the teams' solution approaches. We tested $K = 2$ and looked at teams of two and three agents.

For each case, we considered the all-at-once search and the separation. For the all-at-once search, we constructed the probability distribution over the values $p(x)=\sum_{s=1}^{W} q(s)r(s,x)$ for $x = 1, \ldots, V$ and then constructed the probability transition matrix and calculated the distribution for every time step until the probability of being in a transient state was sufficiently small (no greater than 10^{-6}). This yielded the probability distribution over the values that the agent finds. If a another agent continues the search, then this distribution was used as the distribution over the initial states (those with $n(0) = 0$).

For the separation, we first constructed the probability transition matrix for the search over the types of sets and using that calculated the probability distribution over the types that the agent finds. The distribution $q(s)$ was used as the initial distribution and to construct the probability transition matrix. Let $q'(s)$ be posterior distribution over the types of sets. Then, for each type of set, we constructed the probability transition matrix for the search over the values in that set and calculated the probability distribution over the values that the agent finds if he searches that set. These were combined using $q'(s)$ to generate a posterior distribution over the values.

Thus, for each case and each solution approach (all-at-once and separation) we constructed a posterior distribution over the values. For the all-at-once approach, we recorded this distribution after one agent, two agents, and three agents. For the separation approach, we recorded this distribution for three versions: after one agent searches for a set and a second searches the points in that set (this we call "1-1"); after one agent searches for a set and two other agents search the points in that set (this we call "1-2"); and after two agents search for a set and a third agent searches the points in that set (this we call "2-1").

Results

Here we report the expected value and the probability that the agent(s) will find a value greater than or equal to a given threshold. The expected value of the entire population is also given for comparison.

Table 2. Expected value found in searches with $W = 50$ and $N = 99$. $K = 2$.

Case	Population	One agent	Two agents	Three agents	Separation 1-1	Separation 1-2	Separation 2-1
A	50.00	79.41	85.96	89.08	80.54	86.73	80.89
B	50.00	77.16	83.58	86.82	81.20	86.40	82.54
C	50.00	74.58	80.67	83.89	81.05	85.36	83.34
D	50.00	71.63	77.12	80.09	80.27	83.60	83.60
E	47.96	69.63	75.23	78.31	78.50	81.83	82.19
F	45.92	67.36	73.01	76.14	76.32	79.65	80.30
G	43.88	64.81	70.38	73.50	73.64	76.97	77.74
H	41.83	61.92	67.25	70.24	70.36	73.69	74.25
I	44.28	68.86	75.62	79.40	74.41	79.43	76.82
J	46.73	74.41	81.47	85.15	77.45	83.35	78.71
K	49.18	78.83	85.58	88.81	79.58	86.00	79.88

Table 3. Expected value found in searches with $W = 15$ and $N = 120$. $K = 2$.

Case	Population	One agent	Two agents	Three agents	Separation 1-1	Separation 1-2	Separation 2-1
L	59.33	95.44	103.65	107.57	96.49	104.23	96.97
M	55.83	89.92	98.66	103.18	93.91	100.95	95.87
N	52.33	82.95	91.50	96.26	89.75	95.61	93.49
O	48.83	74.17	81.18	85.15	83.84	87.16	89.93
P	51.75	81.63	90.03	94.75	92.71	95.83	99.95
Q	54.67	87.78	96.56	101.22	96.17	97.66	103.86
R	57.58	92.84	101.38	105.62	97.26	97.94	105.15
S	60.50	97.00	104.95	108.66	97.43	97.85	105.48
T	60.50	97.00	104.95	108.66	101.35	103.76	107.06
U	60.50	97.00	104.95	108.66	102.92	107.17	106.67
V	60.50	97.00	104.95	108.66	99.85	106.99	101.09

The results show that relative performance of the all-at-once and separation approaches varies across the cases considered. In general, of course, the performance of the all-at-once approach improves as more agents collaborate. The improvement (the increase in expected value) varies relatively little across the cases considered here. In cases C, D, E, F, G, and H, high-value solutions are less likely in the entire solution space (note the relatively low expected values for the population in these cases), and the all-at-once approach is inferior, as shown in Table 2 and Figure 2. The 2-1 and 1-2 separation approaches perform about the same in these cases and better than the 1-1 separation approach. In cases A, B, I, and J, however, the distribution across the values in the entire solution space is nearly uniform, and the all-at-once approach is superior. The 1-2 separation approach is superior to the 1-1 and 2-1 separation approaches because the distribution of values is nearly the same in all sets; spending more effort searching the points in the set found yields better results than finding a better set.

In cases O, P, and Q, high-value solutions are less likely in the entire solution space (note the relatively low expected values for the population in these cases), and the all-at-once approach is inferior, as shown in Table 3 and Figure 3. Because high-quality set types are less likely, the 2-1 separation approach is superior to the 1-2 separation approach; spending more effort finding a better set yields better results than searching the points in a set. For cases, L, M, N, S, T, U, and V, however, the distribution across the values in the entire solution space is nearly uniform, and the all-at-once approach is superior. In cases L, M, N, U, and V, the 1-2 separation approach is superior to the 1-1 and 2-1 separation approaches because the distribution of values is nearly the same in all sets; spending more effort searching the points in the set found yields better results than finding a better set.

Because the distribution of values in the entire solution space is uniform for cases S, T, U, and V, the performance of the all-at-once approaches was the same in all of these cases. The performance of the separation approaches was inferior because it was hard for the agents to find the best types of sets and changed as the distribution of values in the sets approached the uniform distribution.

Overall, these results indicate that the separation approaches performed better than the all-at-once approaches when the expected value of the population was less (cases H and O); in these cases, the better values were less likely, which decreased the expected value and reduced the likelihood that the all-at-once approaches would find solutions with good values.

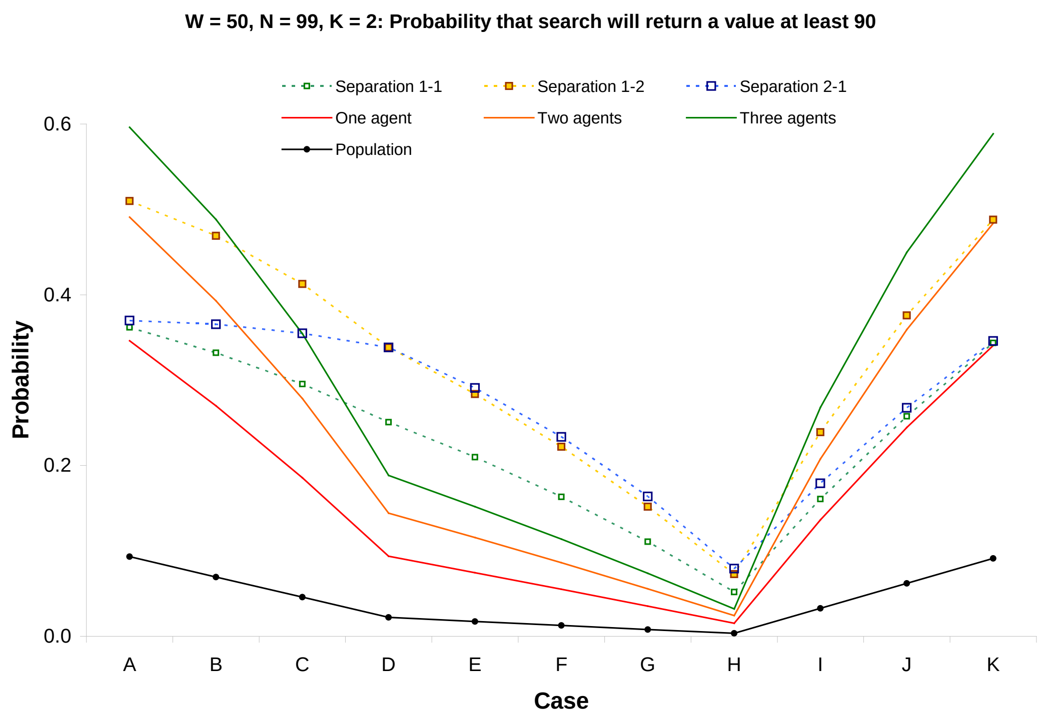

Figure 2. Probability that a search will return a value of at least 90 for cases A to K (*W* = 50, *N* = 99, *K* = 2).

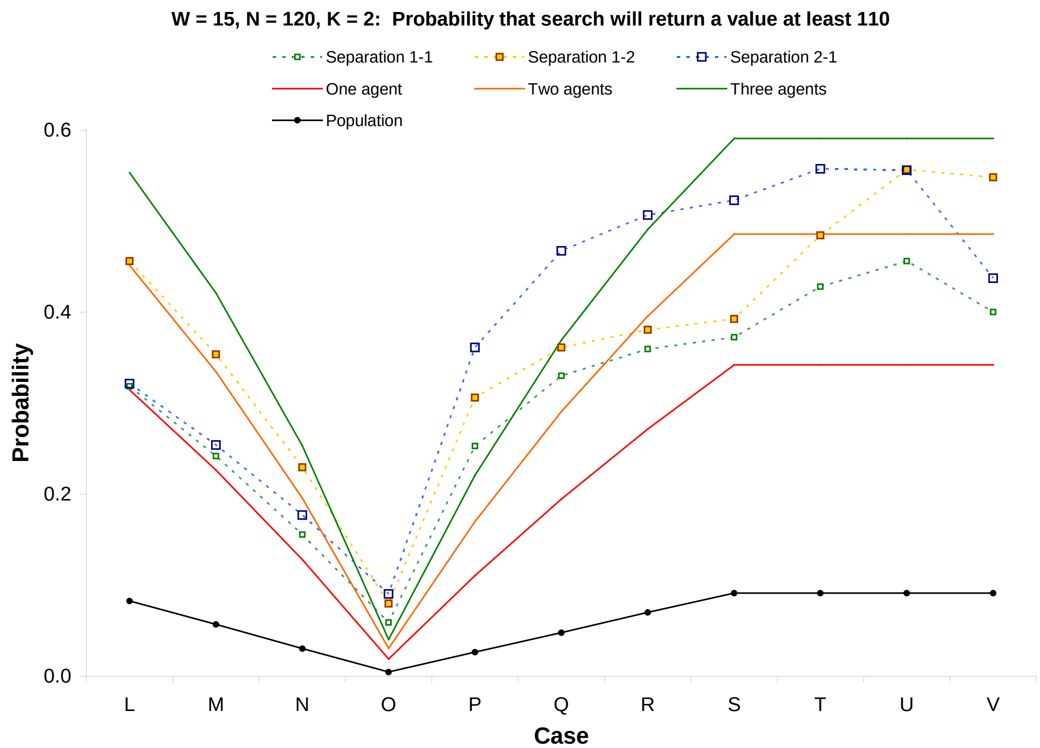

Figure 3. Probability that a search will return a value of at least 110 for cases L to V (*W* = 15, *N* = 120, *K* = 2).

Summary and Conclusions

This paper presented a Markov chain model for analyzing the search of a bounded rational decision-maker and discussed the results of a study that analyzed the relative performance of teams who are solving generic optimization problems. The results indicate that there are some situations in which searching the complete space of solutions is a better approach and other situations in which separating the problem is a better approach. The optimal assignment of subproblems to members of the team also varied. The distribution of the values in the solution space was a key factor.

This study, like the one described by Hong and Page (2004), considered a class of generic optimization problems. Doing this allowed us to vary the characteristics of the search space directly without the distractions of specific decision variables, constraints, and objective functions.

In this study, every agent who was searching the same space was modeled with the same probability distribution over the value of the next point tried; this approach could be modified easily, however, to model agents with different skills who sample the search space in different ways.

Because human decision-makers have limitations and cannot optimize, using well-designed separations can be the best way to find quality solutions and make decisions in situations in which trying to set everything all-at-once will lead to problems that are too large to solve well because the distribution of high quality points in a large solution space makes it difficult for the bounded rational decision-maker to approach an optimal solution.

These results reinforce the conclusions of Herrmann (2010) about the usefulness of separating complex optimization problems for bounded rational decision-makers. They also demonstrate the usefulness of this analysis approach for comparing separations that have different subproblems and different allocations of team resources.

References

Boettcher, Kevin L., and A.H. Levis, "Modeling the Interacting Decisionmaker with Bounded Rationality," *IEEE Transactions on Systems, Man and Cybernetics*, Volume 12, Number 3, pages 334-344, 1982.

Boettcher, Kevin L., and A.H. Levis, "Modeling and analysis of teams of interacting decisionmakers with bounded rationality," *Automatica*, Volume 19, Issue 6, pages 703-709, 1983.

Crowder, R.M., M.A. Robinson, H.P.N. Hughes, Yee-Wai Sim, "The Development of an Agent-Based Modeling Framework for Simulating Engineering Team Work," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Volume 42, Number 6, pages 1425-1439, 2012.

Gigerenzer, Gerd, Peter M. Todd, and the ABC Research Group, 1999, *Simple Heuristics That Make Us Smart*, Oxford University Press, New York.

Gurnani, Ashwin, and Kemper Lewis, 2008, "Collaborative, Decentralized Engineering Design at the Edge of Rationality," *Journal of Mechanical Design*, 130(12), 121101.

Herrmann, Jeffrey W., "Progressive Design Processes and Bounded Rational Designers," *Journal of Mechanical Design*, August 2010, Volume 132, Issue 8, 081005 (8 pages). doi:10.1115/1.4001902

Herrmann, Jeffrey W., "Separating the Inventory Slack Routing Problem," Technical Report 2012-8, Institute for Systems Research, University of Maryland, College Park, July, 2012. Available online at http://hdl.handle.net/1903/12865.

Herrmann, Jeffrey W., "Separating the Searches of Bounded Rational Decision-Makers," Technical Report 2013-13, Institute for Systems Research, University of Maryl and, College Park, June, 2013. Available online at http://hdl.handle.net/1903/13967.

Hong, Lu, and Scott E. Page, "Groups of diverse problem solvers can outperform groups of high-ability problem solvers," *Proceedings of the National Academy of Sciences of the United States of America*, Volume 101, Number 46, pages 16385-16389, 2004.

Levis, Alexander H., "Executable models of decision making organizations," in *Organizational Simulation*, W.B. Rouse and K.R. Boff, eds., John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

Levis, A.H., and Kevin L. Boettcher, "Decisionmaking organizations with acyclical information structures," *IEEE Transactions on Systems, Man and Cybernetics,* vol.SMC-13, Number 3, pages 384-391, 1983.

March, James, and Herbert Simon, 1993, *Organizations*, second edition, Blackwell, Cambridge, Massachusetts.

Montjoy, Adam, and Jeffrey W. Herrmann, "Optimizing Urgent Material Delivery by Maximizing Inventory Slack," Technical Report 2012-6, Institute for Systems Research, University of Maryland, College Park, June 12, 2012. Available online at `http://hdl.handle.net/1903/12499`.

Nutt, Paul C., 2005, "Search During Decision-Making," *European Journal of Operational Research*, 160, pp. 851-876.

Simon, Herbert A., 1981, *The Sciences of the Artificial*, second edition, MIT Press, Cambridge, Massachusetts.

Simon, Herbert A., 1997a, *Models of Bounded Rationality*, Volume 3, The MIT Press, Cambridge, MA.

Simon, Herbert A., 1997b, *Administrative Behavior*, fourth edition, The Free Press, New York.

Wang, Y., and Chiew, V., 2008, "On the Cognitive Process of Human Problem Solving," *Cognitive Systems Research*, doi:10.1016/j.cogsys.2008.08.003.