Herbert Simon introduced the notion of satisficing to explain how boundedly rational agents might approach difficult sequential decision problems. His satisficing decision makers were offered as an alternative to optimizers, who have impressive computational capacities which allow them to maximize. There is no reason, however, why satisficers can not do their task optimally. In this paper, we present a simplified sequential search problem for a satisficing decision maker, and show how to compute its optimal satisficing search policies. Our findings demonstrate that satisficing, when done properly, can be a quite effective search policy.

Key words: satisficing; sequential search; dynamic programming

1. Introduction

The assumptions upon which rational choice theories—including neo-classical economic theory, game theory, and the like—are built are often quite strong, supposing that we take them to apply to actual human decision makers. Simon (1955, 1993) recognized this and proposed an alternative to the maximization objective upon which these theories tend to rest. He suggested that actual agents might be more apt to try to find alternatives that are “good enough” rather than to try to find those that maximize their payoffs. He referred to the former objective as satisficing. Superficially, it seems that these objectives are at odds and that they should produce quite different decision outcomes. In the current paper, we show that these suspicions are not warranted: Satisficing, if done optimally, can produce outcomes on par with those obtained by maximizing.

Consider the commonplace problem of trying to find a house to buy. Presumably, for the house one chooses, one would like to minimize its cost and maximize its quality. Further, one might wish to maximize the quality of one’s neighbors and of the surrounding schools, and to minimize the distance one will travel to work. And so on and so on. The complexity of optimally selecting a home quickly blows up when one simultaneously considers all of these factors.

As traditionally conceived, a satisficing decision maker (DM) need not, for example, solve any difficult non-linear stochastic programming problems. She simply decides what she will find acceptable in a home and takes the first one that meets these standards. For example, she might decide that she cannot spend more than $100,000, that she will not live more than 30 minutes from work, and that she must have a tree in her front yard. She does not worry about making complicated trade-offs among the attributes of the homes, and just tries to find one that meets her aspirations.

Simon (1955) suggested that DMs such as our house-hunter might, for a number of reasons, use such simplified (0-1) payoff functions in order to make decisions. Most important for us here, he suggested that it is often difficult to map a vector of apples (e.g., price) and oranges (e.g., driving time) to a scalar payoff because making appropriate trade-offs is difficult. To get around this, when setting out to shop, a DM can simply decide on the minimum number of apples and the minimum number of oranges she will find acceptable.
It is easy to see why a DM might make decisions in a satisficing manner when alternatives are encountered sequentially. What is not clear is how well a (0-1) payoff function captures individual preferences. Presumably, three apples and three oranges at $x$ is preferred to two oranges and two apples at $x$, all things being equal. When faced with the two alternative bundles simultaneously, the DM would choose the former. This leads to the following two, rather weak, assumptions:

**Assumption 1.** Decision makers who must choose among multi-attribute alternatives presented sequentially are likely to choose the first one they find acceptable.

This is equivalent to saying that DMs are likely to satisfice. However:

**Assumption 2.** All other things equal, more of a good thing is better.

For illustration, let us suppose that a DM is faced with a problem involving just two attributes. For each, she sets an aspiration level, and an alternative that does not exceed both of her aspiration levels will be rejected. Those alternatives that are not immediately rejected are acceptable. An acceptable alternative *satisfices.* Fig. 1 shows this graphically. The alternatives that satisfy are those that fall within the shaded region. The arrows represent the gradient of the DM’s true (non-(0-1)) payoff function at various combinations of the two attribute values. According to Simon (1955), the DM might sensibly set her aspiration level for an attribute at a point where the returns on its values are rapidly diminishing. But, importantly, and in accord with Assumption 2, even within the satisficing set, the gradient is positive. Some acceptable alternatives would be preferred to others. Should the DM thus set higher aspiration levels?

The satisficing DM faces a dilemma in setting her aspiration levels. If she sets them too high, the satisficing regime will shrink, and, *ceteris paribus,* her odds of finding an alternative that satisfices will decrease. On the other hand, if she sets her aspiration levels too low, she will easily find an alternative, but it may be quite poor. Since, by definition, a satisficing DM will take the first alternative she encounters whose attributes meet all of her aspiration levels, her problem lies in determining how best to set her aspiration levels. In this paper, we present a procedure for solving this problem, that is, for finding aspiration levels that allow DMs to *optimally satisfice* or to *satisfice with the objective of maximizing their expected payoffs.*

The reader might worry that our objective involves a perversion of the whole notion of satisficing. The concern might be that satisficing is precisely not optimizing, so satisficing optimally is an oxymoron. However, Simon himself (Simon and Kadane 1975) worked out *optimal* search policies for a class of problems in which a DM has a (0-1) payoff function and wishes to find an alternative that satisfices as quickly as possible—that is, “[o]ptimal algorithms...for satisficing problem-solving search” (p. 235). In essence, the problem Simon and Kadane examined involved searching for treasure in different sites. For each site, the DM knows the unconditional probability that a treasure is buried there, and her objective is simply to find *any* treasure, as they are all equally valuable. Simon and Kadane derived search policies that would allow DMs to minimize the expected number of sites through which they had to search before finding a treasure. The policies dictate the order in which the sites should be searched. In contrast, we assume that the order in which the DM observes alternatives is determined *exogenously.* In our formulation, the DM’s problem lies not in determining which alternatives to examine but which to accept.

As we conceive it here, a multi-attribute satisficing procedure substitutes a simplified, multi-threshold sorting rule for what is, in fact, a more complex multi-attribute utility function. The simplification may be driven by a desire to avoid the complex assessments and computations involved in making explicit multi-attribute trade-offs, by political considerations such as opposition to putting explicit dollar values on saving human lives, or as a device to accommodate the desires of multiple actors whose preferences may not be entirely compatible. The simplification will, in general, exact a payoff penalty. A DM might accept an option that narrowly satisfies all her criteria when another available option would have yielded a higher overall payoff—for example, by offering
high values on several important attributes with low, perhaps below-threshold values on other, less important ones. It is also clear that some satisficing strategies are better than others. Some will cause the DM to search for a hopelessly ideal alternative; others will cause her to accept inferior alternatives.

There are thus two components to the overall loss in expected payoff associated with using a satisficing rather than a maximizing strategy. One component, which we will refer to as inherent penalty, is the result of substituting binary, good/bad evaluations for what are in reality continuous multi-attribute utility functions. The second, which we refer to as aspiration error, is loss of expected payoff resulting from the poor choice of aspiration levels. Assessing aspiration error obviously requires us to specify what good aspiration levels would be. Thus the title, and primary purpose, of our paper: We want to be able to specify, for one important class of problems, the set of aspiration levels that yields the best possible expected payoff to a DM who is constrained to use a satisficing strategy. We want, in short, a tool that can allow us to distinguish between payoff loss resulting from satisficing and payoff loss resulting from satisficing badly.

The rest of the paper is organized as follows. Section 2 presents a formal description of a satisficing problem involving sequential search through a finite set of decision alternatives, and also a general procedure for computing its optimal policy. In Section 3, we present a number of special cases of the general problem and show the optimal policies for each. The general problem can be specified in a very large number of ways, so we have attempted to select special cases that broadly span the space of possibilities. We examine the inherent penalty for following satisficing policies in Section 4 by presenting a procedure for solving the maximization version of our problem, and then comparing its results to those obtained by satisficing. Section 5 describes a method for computing optimal heuristic policies, which are constrained versions of the policies discussed in Sections 2 and 3. Results from applying the heuristic policies to the problems studied in Section 3 suggest that little is lost when the satisficing heuristics are constrained to be quite simple. An infinite-horizon version of the satisficing problem in which the DM pays search costs is presented in Section 6. We conclude in Section 7 and discuss some basic implications of this work.
2. Problem Statement and Optimal Policy

Suppose a DM may examine as many as \( N \) decision alternatives. Each alternative \( n \in \{1, \ldots, N\} \) is represented by a \( K \)-dimensional vector of random variables \( X_n = (X_n^1, X_n^2, \ldots, X_n^K) \), the realization of which is denoted \( x_n = (x_n^1, x_n^2, \ldots, x_n^K) \). The vector elements correspond to the attributes of the decision alternatives. For each attribute \( k \) for each alternative \( n \), we assume that the DM has an aspiration level \( \theta_{nk}^k \), which determines which of that attribute’s values she finds acceptable; specifically, an attribute is acceptable if and only if \( x_n^k \geq \theta_{nk}^k \). The set of all stage \( n \) aspiration levels is represented by \( \theta_n = (\theta_{n1}^1, \theta_{n2}^2, \ldots, \theta_{nK}^K) \). An alternative \( n \) is acceptable (or satisfices) if and only if \( x_n^k \geq \theta_{nk}^k, \forall k \).

A DM’s payoff for selecting an alternative \( n \), denoted \( \phi(x_n) \), is some monotonically increasing function of the alternative’s attribute values. The function should be increasing in its arguments, as this is most consistent with using aspiration levels to make selection decisions. For example, the payoff function can be additive in its arguments, \( \phi(x_n) = \sum_k x_n^k \); or multiplicative, \( \phi(x_n) = \prod_k x_n^k \), with \( x_n^k \geq 0, \forall n; k \); or some combination such as, \( \phi(x_n) = x_n^1 + x_n^2 + \cdots + x_n^K \), \( x_n^k \geq 0, \forall n, k \).

To make clear why we impose the monotonicity constraint on the payoff function, consider one that does not have this property. Restricting \( 0 \leq x^k \leq 110, k \in \{1, 2\} \), suppose that

\[
\phi(x) = \begin{cases} 
  x^1 + x^2 & \text{if } 0 \leq x^2 < 99 \\
  x^1 - 10x^2 & \text{if } 99 \leq x^2 \leq 100 \\
  x^1 + 10x^2 & \text{otherwise.}
\end{cases}
\]

Under this payoff scheme, the DM generally prefers larger \( x^2 \) values to smaller ones, except when 99 \( \leq x^2 \leq 100 \), in which case she prefers smaller values of \( x^2 \). Though a DM could set an aspiration level for \( x^2 \) to be at least 100, it seems unrealistic that anyone whose payoffs depended on this function would use a single aspiration level such as this. More plausibly, the DM might specify that the second attribute is acceptable only if 0 \( \leq x^1 < 99 \) or 100 \( < x^2 \). Our interest is not in these unusual cases, so, again, we will restrict consideration to monotonically increasing payoff functions (cf. Lim et al. 2005).

The DM sees the \( N \) alternatives sequentially and must select one and only one of them. The alternative that is selected is the first one for which \( \sigma_n = 1 \), that is, the first one that satisfices. Once passed, an alternative cannot be recalled. The DM’s objective is to maximize her expected payoff. She controls her fate only by setting her aspiration levels.

In order to set sensible aspiration levels, the DM must know something about the distribution of the alternatives in the world. For tractability, we assume that the DM has full knowledge of the multivariate distribution from which the alternatives are taken, and denote the density for a vector of attributes \( x \) at stage \( n \) by \( f_n(x) \). By allowing the density to vary over \( n \), we can capture, for example, situations in which the quality of encountered alternatives is expected to increase \( E[\phi(X_n)] < E[\phi(X_{n+1})] \), decrease \( E[\phi(X_n)] > E[\phi(X_{n+1})] \), or stay the same \( E[\phi(X_n)] = E[\phi(X_{n+1})] \) over time. Pairs of attributes \( i \) and \( j \) within an alternative may be correlated \( \rho_{ij} \neq 0 \) or uncorrelated \( \rho_{ij} = 0 \). But we assume that the alternative vectors themselves are uncorrelated.

Given a set of aspiration levels at stage \( n \), the probability that an alternative will be acceptable (i.e., that \( \sigma_n = 1 \)) can be expressed as
When the DM does not know whether the attributes of an alternative are acceptable, the expected payoff for selecting that alternative is denoted $E[\phi(X)]$. When the DM does know that the alternative is acceptable, we denote the expected payoff for selecting it by $E[\phi(X)|\sigma_n = 1]$. For an arbitrary payoff function $\phi$, these expectations can be computed at stage $n$ by

$$E[\phi(X_n)] = \int \ldots \int f_n(x)\phi(x)dx^1 \ldots dx^K$$

and

$$E[\phi(X_n)|\sigma_n = 1] = \left[\int \ldots \int f_n(x)dx^1 \ldots dx^K\right]^{-1} \left[\int \ldots \int f_n(x)\phi(x)dx^1 \ldots dx^K\right].$$

Given that the DM will stop on the first alternative that she finds acceptable, how should she set her aspiration levels so that the expected value of her selected alternative will be maximized? Put differently, how can she maximize her expected payoff while satisficing?

Note that the DM faces a straightforward multi-stage decision problem. And, if we can optimally solve the problem at each stage $n$, we can solve the DM’s full problem, from $n = 1$ to $n = N$. Given this property, we can find optimal decision policies for the DM using dynamic programming methods (Bellman 1957). The line of reasoning we evoke here to solve the satisficing problem is closely related to one typically used to solve related optimal stopping problems (see, e.g., Gilbert and Mosteller 1966, Lindley 1961).

At any stage $n$ in the decision problem, the expected payoff or \textit{value} for following some partial policy $\theta_n \cup \{\theta_{n+1}, \ldots, \theta_N\}$ is denoted $V_n(\theta_n)$. Since for all feasible policies the DM must accept the Nth alternative, if reached, $V_N(\theta_N) = E[\phi(X)]$. Then, for any feasible set of aspiration levels at $n < N$, we have

$$V_n(\theta) = P(\sigma_n = 1|\theta)E[\phi(X_n)|\sigma_n = 1] + P(\sigma_n = 0|\theta)V_{n+1}. \quad (1)$$

For each stage $n$, we wish to find the $\theta$ that maximizes $V_n(\theta)$, which we denote $\theta^*_n$. The value of a partial optimal policy $\theta^*_n \cup \{\theta^*_{n+1}, \ldots, \theta^*_N\}$ is represented by $V^*_n \equiv V_n(\theta^*_n)$. Formally, at each stage $n = N - 1$ to $n = 1$, we must solve

$$\theta^*_n = \arg \max_{\theta \in \Omega} V_n(\theta), \quad (2)$$

where $\Omega \subseteq \mathbb{R}^K$ is the set of feasible aspiration levels. Optimal policies can, therefore, be found by coupling numerical optimization (to find the $\theta$s) with dynamic programming (to update the $V$s). Note that when $K = 1$ this problem reduces to a standard full-information optimal stopping problem (e.g., Gilbert and Mosteller 1966). This equivalence is illustrated below for a special case of the satisficing problem.

In the next section, we present optimal policies for a broad range of problems. These problems differ in a number of ways and demonstrate that the nature of the optimal satisficing policies are not always obvious \textit{a priori}. 
3. Some Numerical Examples

The problems described here vary in terms of the properties of the distribution from which the attributes are taken, the nature of the payoff function \( \phi \), and the number of attributes \( K \). We restrict most of our examples to problems with three or fewer attributes (\( K \leq 3 \)), which themselves are computationally quite involved. However, the algorithm described above can be implemented for any arbitrary \( K \).

For the examples reported here that required numerical optimization, we used the fmincon function in MATLAB 6.5, which performs constrained non-linear optimization. Besides the results we report here on the optimal policies themselves, we did extensive analyses of the nature of the solution space. Fig. 2 shows an instance of typical behavior of \( V_n \) as a function of \( \theta_n \) for two-dimensional problems. Fortunately, \( V_n \) tends to be quite well-behaved, making finding the maximum on the surface relatively straightforward.

Example 1: Additive Payoffs with Uniform Attributes

Suppose \( N = 5 \), \( K = 2 \), and, for all \( n \), \( x_n^1 \sim \text{Uni}[0, 1] \), \( x_n^2 \sim \text{Uni}[0, 2] \), \( \rho = 0 \), and \( \phi(x_n) = x_n^1 + x_n^2 \).

The optimal policy and its values for this case are shown in Table 1. We can see that the optimal aspiration levels for the second attribute, which has a greater mean (\( \mu = 1 \)), are greater than those for the first attribute (\( \mu = .50 \)). Also note that the aspiration levels become progressively less strict as the horizon approaches. This is consistent with Simon’s (1955) house-seller who adjusts her aspiration levels downward as time passes and she fails to sell her house.

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_n^* )</td>
<td>2.07</td>
<td>1.99</td>
<td>1.89</td>
<td>1.75</td>
<td>1.50</td>
</tr>
<tr>
<td>( \theta_n^1^* )</td>
<td>0.33</td>
<td>0.26</td>
<td>0.17</td>
<td>0.00</td>
<td>—</td>
</tr>
<tr>
<td>( \theta_n^2^* )</td>
<td>1.33</td>
<td>1.26</td>
<td>1.16</td>
<td>1.00</td>
<td>—</td>
</tr>
</tbody>
</table>

Example 2: Additive Payoffs with Normal Attributes
First, imagine that \( N = 5, K = 2, \) and for all \( n \), \( x_1^n \sim N(0,1), x_2^n \sim N(1,1), \rho = 0, \) and \( \phi(x_n) = x_1^n + x_2^n. \) The optimal policy and its values for this case are shown in the top half of Table 2. Again, the attribute with the greater mean has greater aspiration levels. The behavior of the optimal policy for additive payoffs is quite sensible and consistent with our \textit{a priori} expectations.

But what happens if the attributes are correlated? Using the same parameters but increasing the attribute correlation from \( \rho = 0 \) to \( \rho = .80, \) we obtain the policy shown in the lower half of Table 2. The optimal DM’s aspiration levels increase considerably when the attributes are correlated, as do her expected payoffs \( V^* \). Given that many things in nature will tend to have correlated attributes, this is a desirable consequence. The use of a satisficing search heuristic can, perhaps, be justified by the nature of the alternatives through which the DM searches, viz. that the attributes that compose the alternatives tend to be correlated in predictable ways.

### Table 2  
Optimal policies for Example 2 for both uncorrelated and correlated attributes.

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho = 0 )</td>
<td>( V_1^* )</td>
<td>2.14</td>
<td>1.98</td>
<td>1.78</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>( \theta_1^{1*} )</td>
<td>0.11</td>
<td>-0.01</td>
<td>-0.19</td>
<td>-0.51</td>
</tr>
<tr>
<td></td>
<td>( \theta_2^{1*} )</td>
<td>1.11</td>
<td>0.99</td>
<td>0.81</td>
<td>0.49</td>
</tr>
<tr>
<td>( \rho = .80 )</td>
<td>( V_1^* )</td>
<td>2.70</td>
<td>2.47</td>
<td>2.17</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td>( \theta_1^{1*} )</td>
<td>0.51</td>
<td>0.36</td>
<td>0.14</td>
<td>-0.25</td>
</tr>
<tr>
<td></td>
<td>( \theta_2^{1*} )</td>
<td>1.51</td>
<td>1.36</td>
<td>1.14</td>
<td>0.75</td>
</tr>
</tbody>
</table>

### Example 3: Multiplicative Payoffs with Uniform Attributes

The previous examples were based on additive payoff functions. We next examine a multiplicative function, which might be taken to represent payoffs for alternatives with non-compensatory (or non-substitutable) attributes. Let us see what happens when \( N = 5, K = 2, \) and for all \( n, x_1^n \sim Unif[0,2], x_2^n \sim Unif[2,3], \rho = 0, \) and \( \phi(x_n) = x_1^n x_2^n. \) The results are shown in Table 3. The optimal policy is not absolutely intuitive: Above all, the DM should ensure that she does not select an alternative with a small value on the first attribute; the aspiration level for the second attribute is near its minimum and does not play much of a role. Thus, if a house-hunter would be miserable in a house near a freeway—no matter how cheap the house—then she will be well-advised to set a high aspiration level for her location attribute.

### Table 3  
Optimal policies for Example 3.

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_1^* )</td>
<td>3.88</td>
<td>3.71</td>
<td>3.48</td>
<td>3.12</td>
<td>2.50</td>
</tr>
<tr>
<td>( \theta_1^{1*} )</td>
<td>1.44</td>
<td>1.37</td>
<td>1.25</td>
<td>1.00</td>
<td>---</td>
</tr>
<tr>
<td>( \theta_2^{2*} )</td>
<td>2.16</td>
<td>2.06</td>
<td>2.00</td>
<td>2.00</td>
<td>---</td>
</tr>
</tbody>
</table>

### Example 4: A Three Attribute Problem

Think about the dilemma a satisficing DM faces as \( K \) increases. If she sets her aspiration levels too high, she can expect to find that most alternatives are not acceptable, since it only takes \( x_k^n < \theta_k^n \) for a single \( k \) to make alternative \( n \) unacceptable; and, all things equal, the probability of
at least one attribute being unacceptable increases with the number of attributes. On the other hand, if she sets her aspiration levels too low, finding an acceptable alternative will be easy, but she may end up with a relatively undesirable alternative. What to do?

Consider a problem in which \( x_n^1 \sim N(0, 1), x_n^2 \sim N(5, 1), x_n^3 \sim N(1, 1) \), \( \rho = 0 \), and \( \phi(x_n) = x_n^1 + x_n^2 + x_n^3 \) for all \( n \). Results are shown in Table 4. In isolation, the general nature of the optimal policy is consistent with expectations. However, the results are more interesting if we compare them to those presented in Example 2 in which the DM faced a related problem. In fact, if we replace the payoff function for the current problem with \( \phi(x_n) = x_n^1 + 0(x_n^2) + x_n^3 \), we get the Example 2 problem (with \( \rho = 0 \)). Note that the optimal aspiration levels for the \( K = 3 \) problem are uniformly smaller than the corresponding aspiration levels from Example 2. The intuition for this is quite simple: Suppose a DM used the same aspiration levels for attributes 1 and 3 in the current problem and \( K \) alternatives when she used for attributes 1 and 2 in the Example 2 problem. Wherever she sets \( \theta_n^1, \theta_n^2, \theta_n^3 \) smaller than the corresponding aspiration levels from Example 2, the fact that to determine how to optimally satisfice over just five three-attribute alternatives requires intensive and quite time-consuming computations.

**Example 5: A Directly Solvable Problem: Uniform Attributes**

In this example, we demonstrate a special case of the satisficing problem in which the optimal policy can be solved analytically—i.e., without resorting to numerical optimization. Assume that we have two uncorrelated attributes \( x_n^k \sim \text{Unif}[0, 1], k \in \{1, 2\} \), and \( \phi(x_n) = x_n^1 + x_n^2 \), for all \( n \). Substituting into Eq. 1, we get

\[
V_n(\theta_n) = (1 - \theta_n^1)(1 - \theta_n^2) \left( \theta_n^1 + \theta_n^2 + \frac{2 - \theta_n^1 - \theta_n^2}{2} \right) + (\theta_n^1 + \theta_n^2 - \theta_n^1 \theta_n^2) V_{n+1}. \tag{3}
\]

Our objective is to find the \( \theta_n^1 \) and \( \theta_n^2 \) that maximize this expression. The first-order conditions are

\[
\frac{\partial V}{\partial \theta_n^1} = \frac{1}{2} \left( \theta_n^2 \right)^2 + \theta_n^1 \theta_n^2 - \theta_n^1 - \theta_n^2 V_{n+1} + V_{n+1} - \frac{1}{2} = 0
\]

and

\[
\frac{\partial V}{\partial \theta_n^2} = \frac{1}{2} \left( \theta_n^1 \right)^2 + \theta_n^1 \theta_n^2 - \theta_n^2 - \theta_n^1 V_{n+1} + V_{n+1} - \frac{1}{2} = 0.
\]

**Table 4** Optimal policies for Example 4.

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_n^* )</td>
<td>2.78</td>
<td>2.60</td>
<td>2.37</td>
<td>2.04</td>
<td>1.50</td>
</tr>
<tr>
<td>( \theta_n^{1*} )</td>
<td>-0.22</td>
<td>-0.33</td>
<td>-0.49</td>
<td>-0.76</td>
<td>—</td>
</tr>
<tr>
<td>( \theta_n^{2*} )</td>
<td>0.28</td>
<td>0.17</td>
<td>0.01</td>
<td>-0.27</td>
<td>—</td>
</tr>
<tr>
<td>( \theta_n^{3*} )</td>
<td>0.78</td>
<td>0.67</td>
<td>0.51</td>
<td>0.23</td>
<td>—</td>
</tr>
</tbody>
</table>

We should comment that the complexity of finding optimal policies grows rapidly in the number of attributes. The time required to solve the \( K = 3 \) case was greater than the time required for the corresponding \( K = 2 \) problem by roughly an order of magnitude. There is some irony in the fact that to determine how to optimally satisfice over just five three-attribute alternatives requires

Bearden and Connolly: Optimal Satisficing

Article submitted to Management Science; manuscript no. (Please, provide the manuscript number!)
Solving the system of equations, we conclude that \( \theta_1^n = \theta_2^n = \frac{2}{3}V_{n+1} - \frac{1}{3} \) is a stationary point. (There is another stationary point at \( \theta_1^n = \theta_2^n = 1 \), which can be disregarded since it will always leave \( V_n = V_{n+1} \).) Evaluating the second partial derivatives for permissible \( V_{n+1} \), we find that this point is indeed a maximum. Hence, starting with \( V^*_n = 1 \) and working from \( n = N \) to \( n = 1 \), \( \theta_k^n = \frac{2}{3}V^*_n + 1 \) for \( k \in \{1, 2\} \). Results for a problem with \( N = 5 \) are shown in Table 5. Obviously, with \( \theta^*_n \) in hand for the current problem, optimal policies can be obtained for uniformly distributed attributes on any interval \([a, b]\).

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V^*_n )</td>
<td>1.35</td>
<td>1.30</td>
<td>1.24</td>
<td>1.15</td>
<td>1.00</td>
</tr>
<tr>
<td>( \theta^1_n, \theta^2_n )</td>
<td>0.54</td>
<td>0.49</td>
<td>0.43</td>
<td>0.33</td>
<td>—</td>
</tr>
</tbody>
</table>

To see the relation to standard optimal stopping problems, consider applying the same reasoning to a problem with a single attribute (\( K = 1 \)). We have

\[
V_n(\theta) = (1 - \theta_n) \left( \theta_n + \frac{1 - \theta_n}{2} \right) + \theta_n V_{n+1},
\]

and

\[
\frac{\partial V}{\partial \theta_n} = V_{n+1} - \theta_n.
\]

Setting \( \frac{\partial V}{\partial \theta_n} = 0 \) and solving for \( \theta_n \), we find that Eq. 4 is maximized when \( \theta_n = V_{n+1} \). This is solution to the standard full-information optimal stopping problem (Gilbert and Mosteller 1966). It is easy to see that (and why) optimal satisficing and standard optimal stopping are equivalent when \( K = 1 \) for arbitrary densities.

Most generally, assuming that \( \theta^k_n = \theta^{k'}_n, \forall k, k' \in \{1, \ldots, K\} \), for independent uniformly distributed attributes on \([0, 1]\), we get

\[
V_n(\theta) = (1 - \theta_n)^K \left( K\theta_n^{K} + K - K\theta_n^K \right) + \left[ 1 - (1 - \theta_n)^K \right] V_{n+1}
\]

and

\[
\theta^*_n = \frac{2V_{n+1}^* + 1 - K}{K + 1}, \quad k \in \{1, \ldots, K\},
\]

where \( V_{n\downarrow}^* = K \frac{1}{2} \). Expressed this way, we can see what happens to the aspiration levels as \( K \) grows. Consider stage \( N - 1 \) aspiration levels; for these we can observe that \( \theta^*_{N-1} \rightarrow 0 \) as \( K \rightarrow \infty \). For large \( K \), it is unlikely that the DM will ever find an alternative that satisfies unless she sets her aspiration levels very low.\(^1\)

**Example 6: Non-stationary Attributes**

Recall that the distributions from which the alternatives are drawn may vary with \( n \). Imagine a house-hunting DM who believes that the quality of the houses she will encounter during her search will tend to decrease in time. Let us assume that she believes that \( x_1^n \sim N(\frac{N-n}{N}, 1) \) and \( x_2^n \sim N(2\frac{N-n}{N}, 1) \), with \( \rho = 0 \), and that, for all \( n \), \( \phi(x_n) = x_1^n + x_2^n \). The optimal policy for \( N = 5 \)

\(^1\)We have also obtained the analogous solution for exponentially distributed attributes.
Table 6  Optimal policies for Example 6 for non-stationary and stationary distributions.

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_n^*</td>
<td>2.77</td>
<td>2.15</td>
<td>1.52</td>
<td>0.83</td>
<td>0.00</td>
</tr>
<tr>
<td>θ_1^n</td>
<td>0.13</td>
<td>-0.10</td>
<td>-0.35</td>
<td>-0.72</td>
<td>—</td>
</tr>
<tr>
<td>θ_2^n</td>
<td>0.93</td>
<td>0.50</td>
<td>0.05</td>
<td>-0.52</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_n^*</td>
<td>3.99</td>
<td>3.79</td>
<td>3.51</td>
<td>3.05</td>
<td>1.50</td>
</tr>
<tr>
<td>θ_1^n</td>
<td>1.00</td>
<td>0.82</td>
<td>0.53</td>
<td>-0.61</td>
<td>—</td>
</tr>
<tr>
<td>θ_2^n</td>
<td>2.00</td>
<td>1.82</td>
<td>1.53</td>
<td>0.38</td>
<td>—</td>
</tr>
</tbody>
</table>

4. Comparing “Maximizing” and “Satisficing” Policies

We assume that the maximizing DM evaluates the decision alternatives using her multi-attribute utility function. She does not select an alternative that is acceptable on each of its attributes; instead, she makes her accept-reject decisions on the basis of the actual payoff φ(x) for selecting an alternative. It turns out, then, that the maximizing problem is just a standard full-information optimal stopping problem (e.g. Gilbert and Mosteller 1966, MacQueen and Miller 1960). One of the questions we wish to answer is: How great is the inherent penalty for satisficing? That is, in comparison to maximizing, how much does one lose by satisficing?

The optimal maximizing policy for each stage n is an acceptability threshold s⋆n; under it, the DM stops at the first stage n at which φ(X_n) ≥ s⋆n. The value of some stage n < N policy s is

$$\tilde{V}_n(s) = P[\phi(X_n) \geq s]E[\phi(X_n)|\phi(X_n) \geq s] + P[\phi(X_n) < s] \tilde{V}_{n+1}.$$  (5)

As before, $\tilde{V}_N = E[\phi(X)]$. Our objective for n = N − 1 to n = 1 is to find

$$s_n^* = \arg \max_{s \in R} \tilde{V}_n(s).$$  (6)

In order to determine the optimal maximizing policy, we must know the density of φ(x), denoted g[φ(x)], which is a function of both φ(x) and f(x). When the payoff function is additive in its arguments and the attributes are independent, g[φ(x)] can be obtained by convolution. For instance, if two independent normally distributed attributes have means $\mu_1$ and $\mu_2$ and variances $\sigma_1^2$ and $\sigma_2^2$, then g[φ(x)] is normal with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$. More generally, the convolution can be applied recursively to get the desired density for K normally distributed attributes when the payoff function is additive.

However, in general, an analytic expression for g[φ(x)] will be difficult to obtain. Fortunately, in practice, we do not actually need it. The stage n stopping probability for some policy s can be obtained numerically by evaluating
Figure 3  Geometric relationship between optimal maximizing and optimal satisficing policies for a single stage $n$. The policies are for a problem with $x^k \sim Un[0,1], k \in \{1,2\}$ and $\phi(x) = x^1 + 2x^2$.

The expectation $E[\phi(X_n)|\phi(X_n) \geq s]$ can be similarly expressed. Given this, Eq. 6 can itself be solved quite easily, since, by the Principle of Optimality, we know that $s^*_n = \tilde{V}^*_n + 1$.

Fig. 3 graphically depicts the relationship between satisficing and maximizing policies at a single stage $n$. Under the maximizing policy, the DM takes an alternative whose attribute values both fall above the dotted line, which represents attribute combinations with $\phi(x_n) = V^*_n + 1$. In contrast, the satisficing DM accepts only those alternatives for which $x^1 \geq \theta^1$ and $x^2 \geq \theta^2$, which fall in the darkened rectangular area in the top right. Note that the satisficing policy both accepts some alternatives that are rejected by the maximizing policy and also rejects some that are accepted by it. It is easy to see that the maximizing policy is more compensatory than the satisficing policy. Under the maximizing policy, the DM will take an alternative with relatively small values on $x^1$ ($x^2$) as long as the alternative’s $x^2$ ($x^1$) value compensates.

Optimal maximizing policies were obtained for a subset of the problems reported in Section 3. Our interest is not so much in the policies themselves but in the values of those policies. These are shown in Table 7. The most striking finding is that very little is lost by satisficing. In most of the cases, optimal satisficing earns the DM about 99% of what she would earn maximizing. Importantly, though, how much is lost by satisficing depends, among other things, on the nature of the distribution from which the alternatives are taken. Considering those examples with additive payoff functions (Examples 1, 2, and 4), we can see that the relative efficiency of satisficing is worse when the attribute values are taken from normal distributions (Examples 2 and 4) than when they are taken from uniform ones (Example 1). However, when the attributes are correlated, the performance of satisficing improves; and, in the case of normally distributed attribute values,
satisficing earns 99% of what is earned by the maximizing policy. Thus, in the wild, where attributes will likely be correlated, the inherent penalty for satisficing may be negligible.

Table 7  Value of optimal maximizing policies for Example 1-4. \( \frac{\tilde{V}_1}{V_1^*} \) is the ratio of the expected payoffs of the optimal satisficing policy to those of the optimal maximizing policy. Example 2a corresponds to the problem in Example 2 with \( \rho = 0 \); 2b corresponds to the one with \( \rho = .80 \).

<table>
<thead>
<tr>
<th>Example</th>
<th>1</th>
<th>2a</th>
<th>2b</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{V}_1 )</td>
<td>2.10</td>
<td>2.29</td>
<td>2.73</td>
<td>3.90</td>
<td>3.08</td>
</tr>
<tr>
<td>( \frac{\tilde{V}_1}{V_1^*} )</td>
<td>0.99</td>
<td>0.93</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

5. Optimal “Heuristic” Satisficing

In the general formulation of our problem, we allowed the satisficing DM to set aspiration levels separately for each stage \( n \). One might complain that the complexity of the resulting policies, which have \( K(N - 1) \) separate aspiration levels, is not in the spirit of satisficing. In anticipation of this objection, we will now examine a constrained version of the problem in which the DM must use the same aspiration levels \( \theta_H = \{ \theta_1^H, \ldots, \theta_K^H \} \) for all \( n \in \{1, \ldots, N\} \). To keep our expressions from becoming too cumbersome, let us denote \( P(\sigma_n = 1|\theta_H) \), which is constant for all \( n \), by \( P_\sigma \). Now, assuming that the attribute density function does not vary with \( n \) (i.e., \( f_n = f, \forall n \)), the value for some policy \( \theta_H \) can be expressed as

\[
V_H (\theta_H) = \sum_{n=1}^{N-1} \left[ (1 - P_\sigma)^{n-1} P_\sigma E \left[ \phi(X) | \sigma_n = 1 \right] \right] + (1 - P_\sigma)^{N-1} E \left[ \phi(X) \right].
\]

The last term on the right side of Eq. 7 corresponds to the event that the DM reaches the final alternative and must accept it. We denote the \( \theta_H \) that maximizes Eq. 7 by \( \theta_H^* = \{ \theta_1^H*, \ldots, \theta_K^H* \} \), and let \( V_H^* = V_H(\theta_H^*) \).

How much is optimal satisficing affected by the constraint that the aspiration levels must be fixed across the alternatives? To address this question, we found optimal policies for Examples 1 through 4. The results are summarized in Table 8. Most interestingly, we observe that the expected payoffs for following \( \theta_H^* \) are quite close to those for following the optimal unconstrained satisficing policy: In all cases, the heuristic policy earns at least 98% of the expected payoff for the optimal unconstrained policy.

Table 8  Optimal heuristic policies for Example 1-4. \( \frac{V_H^*}{V_1^*} \) is the ratio of the expected payoffs for of the heuristic policy to those of the unconstrained policy. Example 2a corresponds to the problem in Example 2 with \( \rho = 0 \); 2b corresponds to the one with \( \rho = .80 \).

<table>
<thead>
<tr>
<th>Example</th>
<th>1</th>
<th>2a</th>
<th>2b</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_H^* )</td>
<td>2.05</td>
<td>2.10</td>
<td>2.64</td>
<td>3.84</td>
<td>1.00</td>
</tr>
<tr>
<td>( \frac{V_H^<em>}{V_1^</em>} )</td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>( \theta_1^H* )</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \theta_2^H* )</td>
<td>0.22</td>
<td>-0.09</td>
<td>0.26</td>
<td>1.34</td>
<td>-0.40</td>
</tr>
<tr>
<td>( \theta_3^H* )</td>
<td>1.22</td>
<td>0.91</td>
<td>1.26</td>
<td>2.00</td>
<td>0.10</td>
</tr>
<tr>
<td>( \theta_4^H* )</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.60</td>
</tr>
</tbody>
</table>
6. Infinite-Horizon Satisficing

We now present a continuous-time infinite-horizon (control-type) satisficing problem in which the DM must pay a fixed cost \( c > 0 \) per unit of time she searches. If the cost were 0, the DM could, if infinitely patient, wait forever to select a “perfect” alternative. To get the value function, let us replace the sum in Eq. 7 with an integral. Then with a slight modification to remove the final term, which vanishes, adding the cost term, and replacing \( n \) with \( t \) (time), we get

\[
V(\theta_\infty) = \int_1^\infty (1 - P_\sigma)^t (P_\sigma E[\phi(X) | \sigma = 1] - c) \, dt,
\]

where \( \theta_\infty = \{\theta_1^\infty, \ldots, \theta^K_\infty\} \) are the aspiration levels for all \( t \). We assume that the attribute density function does not vary with time, \( f_t = f, \forall t \). Now our objective is to solve

\[
\theta_\infty^\star = \arg \max_{\theta \in \Omega} V(\theta_\infty).
\]

As with the previous problems, Eq. 9 can be solved numerically. Optimal policies for the infinite-horizon instance of the problem in Example 1 for different search costs are shown in Table 9. As the search costs increase, the DM lowers her aspiration levels and can also expect to earn less, as we expected.

<table>
<thead>
<tr>
<th>( c )</th>
<th>0.01</th>
<th>0.10</th>
<th>0.25</th>
<th>0.50</th>
<th>1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_\infty^\star )</td>
<td>2.32</td>
<td>1.40</td>
<td>0.84</td>
<td>0.40</td>
<td>0.07</td>
</tr>
<tr>
<td>( \theta_1^\infty )</td>
<td>0.73</td>
<td>0.47</td>
<td>0.29</td>
<td>0.11</td>
<td>0.00</td>
</tr>
<tr>
<td>( \theta_2^\infty )</td>
<td>1.73</td>
<td>1.47</td>
<td>1.29</td>
<td>1.11</td>
<td>0.69</td>
</tr>
</tbody>
</table>

7. Conclusion

Our primary objective in this paper is to make it clear that satisficing does not entail behaving simplistically. In fact, an optimal satisficer is behaving rather cleverly. Another important point is that satisficing does not necessarily leave a lot on the table. A satisficing DM can do quite well compared to a maximizer faced with the same kinds of problems. This is particular true in environments in which the attributes that compose the alternatives are correlated. Hence, satisficing policies may even be ecologically rational (Gigerenzer, Todd, et al. 1999).

Elsewhere (Bearden and Connolly 2005) we have shown that experimental subjects who are required to set aspiration levels and who only learn whether the attribute values of encountered alternatives are acceptable (by the definition given above in the problem statement) do as well as subjects who can see the actual attribute values before making selection decisions. In other words, the subjects who were forced to behave like satisficers did as well as those who were free to maximize. We found this using standard methods from experimental economics, including, most importantly, incentive compatible payoffs. The subjects’ aspiration levels did depart systematically from the optimal aspiration levels; specifically, they tended to set their aspiration levels too high. In sum, those subjects who were free to maximize behaved sub-optimally. Those who were constrained to satisfice satisficed sub-optimally. Overall, the two groups did equally well.

---

The results of this experiment are presented in a paper that is currently under review at Organizational Behavior and Human Decision Processes. It is available online at: [http://www.u.arizona.edu/~nbearden/Papers/BeardenConnolly.pdf](http://www.u.arizona.edu/~nbearden/Papers/BeardenConnolly.pdf)
The core message of the present paper is that we should treat satisficing not as a specific decision strategy but as a class of strategies. As we have shown, some of the strategies in this class are capable of performance essentially equivalent to that of the global optimizing (or maximizing) strategy, at least for a broad class of problems. Other satisficing strategies yield quite poor overall performance. In our terminology the inherent penalty associated with the use of a satisficing strategy may often be quite small, though for an ill-chosen satisficing strategy actual losses may be huge. Clearly, given the very large computational costs incurred in optimal satisficing we are not proposing this as a plausible descriptive model of real DMs. Indeed, it was precisely to avoid such costs that Simon originally proposed the satisficing idea. However, the DM who adopts a satisficing strategy in order to avoid the costs of making explicit inter-attribute trade-offs is not thereby condemned to suffer large utility losses. The procedures outlined here provide a metric by which the loss associated with any given satisficing strategy can be assessed. This opens the way to developing aspiration level setting rules that, if not optimal, can at least keep the expected losses small. Descriptively, it allows investigators to explore the extent to which specific DMs have been able to develop such rules for themselves. It may well be that, in some settings, DMs with or without such help have been able to arrive at satisficing decision strategies that yield excellent overall performance for little effort. By clarifying the notion of optimal satisficing, we hope to have shed some light on the descriptively more realistic notion of acceptable satisficing—surely the notion that Simon originally had in mind.

Acknowledgments
We gratefully acknowledge financial support by a contract F49620-03-1-0377 from the AFOSR/MURI to the Department of Systems and Industrial Engineering and the Department of Management and Policy at the University of Arizona. We also thank Amnon Rapoport for his kind suggestions.

References
The Mathworks, Inc. MATLAB v. 6.5.