CONSIDER-THEN-CHOOSE MODELS IN DECISION-BASED DESIGN OPTIMIZATION

W. Ross Morrow  
Mechanical Engineering  
Iowa State University  
Ames, Iowa  
wrmmorrow@iastate.edu

Minhua Long  
Mechanical Engineering  
Iowa State University  
Ames, Iowa  
mhlong@iastate.edu

Erin F. MacDonald  
Mechanical Engineering  
Iowa State University  
Ames, Iowa  
erinmacd@iastate.edu

ABSTRACT

This article describes an advancement in design optimization that includes consumer purchasing decisions. Decision-based design optimization commonly relies on Discrete Choice Analysis (DCA) to forecast sales and revenues for different product variants. Conventional DCA represents consumer choice as a compensatory process through maximization of a smooth utility function, has proved to be reasonably accurate at predicting choice, and interfaces easily with engineering models. However the marketing literature has documented significant improvement in modeling customer choice with the use of models that incorporate descriptive (non-compensatory) and predictive (compensatory) components. The descriptive non-compensatory component can model people that narrow their decisions to a small set of products using heuristic decision rules—a process referred to as “consider-then-choose”. This article demonstrates that ignoring consider-then-choose behavior can lead to sub-optimal designs, and that optimality cannot be “recovered” by changing marketing variables alone. A new computational approach is proposed for solving optimal design problems with consider-then-choose models whose screening rules are based on logical conjunctions. Computational results are provided using three state-of-the-art commercial solvers: matlab, KNITRO, and SNOPT.

1 INTRODUCTION

Researchers in Decision-Based Design (DBD), Enterprise-System Design, and Value-Driven Design assert that business objectives should replace engineering requirements or performance metrics as the objective for engineering design; see, e.g., [1–3]. Wassener & Chen [4] have specifically proposed the use of economic value to the firm — e.g., profits — as the metric against which different designs should be judged. Profits have since been applied as a design objective in numerous applications of DBD. In order to forecast sales for alternative designs, a pre-requisite to using profits as a design objective, engineering researchers have integrated Discrete Choice Analysis (DCA) [5–7] with engineering design optimization; see, e.g., [4, 8–16]. In DCA, every product is assigned a utility that depends on the product’s performance characteristics and price. Most applications use utilities that are continuous in product characteristics and prices, and thus the corresponding choice model can only represent compensatory decisions in which product attributes and price are traded off against one another.

Unfortunately, compensatory models imperfectly characterize real choice behaviors, and may thus generate misleading portraits of profits as a function of design decisions. Researchers in economics, psychology, and marketing have studied non-compensatory decision processes in which small changes in product attributes do not compensate for a current combination of attributes that is not preferred. In 1957, Simon proposed the principle of satisficing [17]: individuals search through the available options only until a suitable option is discovered, regardless of whether that option objectively maximizes utility or value. In the mid-late 1960’s, several researchers introduced the conjunctive model in which an option must attain a minimum value for several criteria to be selected [18–20]. In 1972, Tversky intro-
duced the Elimination-By-Aspects model, in which individuals select a particular product attribute and eliminate all options that fail to meet some minimum criteria for that attribute, and repeat this process over all product attributes until all but one alternative is eliminated [21]; attribute selection was proposed to be probabilistically proportional to the relative importance of that attribute. The use of non-compensatory rules in decision processing can be “procedurally rational” [22], if not “substantively” rational, especially when information about product attributes is expensive to obtain or process [23, 24].

The generic model form considered in this article, termed the consider-then-choose model by Hauser & Wernerfelt, posits that individuals use non-compensatory screening rules to limit the number of options they consider; individuals then choose from these limited options with a compensatory evaluation [23, 25]. Experimental evidence that individuals use non-compensatory rules to form consideration sets was first presented in 1976 [26] and has had a long tradition in marketing; see, e.g., [24] for numerous references. Including consideration is known to significantly improve choice model accuracy: in one early study [25], “consideration explains approximately 80% of the uncertainty in deodorant choice.” [27] More importantly, marketing researchers have recently built statistical methods for inferring screening rules from choice experiments [28–32]. One practical consequence of a recent choice experiment is that addressing consider-then-choose behavior will be vital to achieving fast progress towards sustainability in the U.S. transportation sector: A 2008 vehicle choice survey discovered that 44% of respondents would not have considered purchasing a Hybrid Electric Vehicle (HEV) [31, 33]. This lack of consideration has broad implications for the viability of alternative fuel vehicles and the impact of incentives—from either vehicle manufacturers, retailers, or government agencies—on alternative vehicle purchase.

This article initiates the integration of consider-then-choose type models into engineering design optimization. A prerequisite for this integration is to have a non-compensatory model that accurately reflects choice behavior; capturing the “right” consideration rules is particularly important. We, however, do not study or describe the estimation of such models. Interested readers should see the growing literature in marketing science on this topic; e.g., [28–32] and references therein. Currently there is no literature on optimal design once such a model is constructed, either in marketing or engineering design. The primary obstacle to efficiently solving optimal design problems with a consider-then-choose model is the fact that the choice probabilities, and hence profits, are a discontinuous function of product design and pricing decisions. Thus it is not obvious how to apply existing Nonlinear Programming (NLP) algorithms such as Sequential Quadratic Programming (SQP) [34, 35] or Interior-Point (IP) methods [35, 36]. Moreover, the discontinuous objective may imply that many local optima exist and makes validating local optimality of a potential solution challenging. We focus on the simplest type of non-compensatory screening rule, conjunctive rules, because they can be cast as a smooth vector inequality. Even with the simplest type of rule, the associated optimal design problem is significantly more complex than with purely compensatory DCA models.

To motivate research with this new model type, we first provide a single vehicle design example for which ignoring consideration behavior leads to the wrong design. Moreover, changing price alone cannot “recover” the majority of profits lost by failing to account for a non-compensatory screening rule in consumer choice during the design phase. Combined, these results imply that consideration is a problem for designers, not just for marketers. To facilitate further applications of consider-then-choose models in engineering design, we then investigate numerical methods that may apply to models with multiple conjunctive screening rules and many products. Our approach to solving the single vehicle design example essentially casts the design problem as a discrete optimization problem over the potential collections of consideration sets. For reasons discussed below, we are skeptical that this technique will scale to problems with multiple screening rules or many products. A novel relaxation method applying techniques from Mathematical Programming with Complementarity Constraints (MPCC) is proposed and validated on our single vehicle design example. To investigate the scalability of this approach, we generalize our example to include consumer heterogeneity in both the screening rules and the utility function used in the compensatory evaluation. Computational trials with matlab, KNITRO, and SNOPT suggest that this MPCC relaxation has promise as a scalable technique for solving practical optimal design problems with consider-then-choose models.

2 CONSIDER-THEN-CHOOSE MODELS

A variety of probabilistic choice models exist in DCA, most of which have been applied in engineering design [4, 8–16, 37, 38]. (For a good introduction to DCA, see [7].) The consider-then-choose model introduces non-compensatory decisions into DCA as follows: Each individual \( i \in \{1, \ldots, I\} \) “considers” the products indexed by the elements of an individual-specific consideration set \( \mathcal{C}(X, p) \subset \{1, \ldots, J\} \) that depends on the characteristics of all products, \( X = (x_1, \ldots, x_J) \in \mathcal{X}^J \), and on all product prices \( p = (p_1, \ldots, p_J) \in \mathcal{R}^J \). Some collection of screening rules \( s_i : \mathcal{X}^J \times \mathcal{R}^J \rightarrow \mathcal{R}^J \), define this set:

\[
\mathcal{C}(X, p) = \left\{ j \in \{1, \ldots, J\} : s_i(x_j, p_j) \leq 0 \right\} \tag{1}
\]

Given a collection of screening rules and the associated consideration set the probability that individual \( i \) chooses product
\( j \in \{1, \ldots, J\} \) is given by
\[
P_{C,j}(X, p) = \begin{cases} 
e^{-x_i(X, p) / \varphi} & \text{if } j \in C_i(X, p) \\ 0 & \text{if } j \notin C_i(X, p) \end{cases}
\] (2)

Eqn. (2) reflects the model form currently being applied in marketing studies [28, 31, 32]. The non-compensatory stage is embodied by the consideration set, while the compensatory evaluation stage is based on the multiple individual Logit model as it appears in Latent-Class and Conjoint analysis.

Inclusion of choice sets, rather than consideration sets, has always been a component of DCA [5]. For example, choice surveys often cover a broad spectrum of product features while posing choice tasks with relatively few products at a time [6]. Model estimation then requires connecting the observed choices directly with the specific alternatives in the particular choice task; i.e., connecting choices with the task-specific choice set. Choice sets, at least as applied in model estimation, are controlled by the modeler and can even be manipulated to obtain desirable statistical properties [6]. Consideration sets, on the other hand, are determined by the individual as a function of product attributes, prices, and other elements of the choice context. They are, in a sense, self-selected choice sets that can only be understood by capturing the underlying screening rules.

Consider-then-choose choice probabilities are distinguished from typical Logit choice probabilities in that they are discontinuous functions of product characteristics and prices. The following example illustrates this fact: Suppose individuals will consider a product if \( p \leq 1 \), and if they consider the product have utilities given by \( u(p) = -p \) (for the product) and \( \varphi = 0 \) (for the “outside good” or no-purchase option). The consider-then-choose choice probabilities are then given by \( P^C(p) = e^{-p} / (1 + e^{-p}) \) if \( p \leq 1 \) and 0 if \( p > 1 \). Note, however, that \( \lim_{p \to 1} P^C(p) = 1/2 \), and thus \( P^C \) is discontinuous at \( p = 1 \). In general, the ability of changes to product characteristics and prices to change the consideration sets generates discontinuities in the choice probabilities.

Recent marketing models allow for complex logical relationships to be encoded as functional inequalities through screening rules; see, e.g., [30, 32]. As a consequence, the screening rules that appear in Eqn. (1) may not be smooth functions of product characteristics and prices; see the appendices for several examples. For simplicity, this article addresses only conjunctive rules that can be defined as vector inequalities with smooth screening functions. That is, we assume that the vector function \( s_i \) appearing in Eqn. (1) is smooth. Even though marketing researchers have extended beyond the representation of choice enabled by conjunctive rules, optimal design with conjunctive rules alone has sufficient technical difficulty to warrant special attention. Further work will be needed to extend the proposals in this article to more general types of screening rules, such as disjunctions and disjunctions-of-conjunctions [30, 31].

3 VEHICLE PURCHASE WITH AN OWNING AND OPERATING COST BUDGET

This section examines a simple model of vehicle design with demand that exhibits consider-then-choose behavior with a simple conjunctive screening rule to demonstrate two important facts: First, significantly sub-optimal designs may be obtained if consider-then-choose behavior is ignored in the product design phase. Second, firms may not be able to “recover” from sub-optimal design decisions by manipulating marketing variables. These facts justify identifying and planning for consider-then-choose behavior during product design, rather than leaving product consideration to marketing activities. Our example modifies the model of vehicle design recently built by Whitefoot et al [39]; see Appendix B. The purpose of this example is not to give an accurate representation of vehicle design, but rather to identify what could happen if consideration behavior is ignored during design.

The Model

A firm designs a vehicle by choosing its 0-60 acceleration time and Manufacturer’s Suggested Retail Price (MSRP). Fuel consumption (g, in gpm) and 0-60 acceleration time (a, in s) are related by a function \( g(a) \); see Eqn. (24) and Fig. 1. Unit costs are also a function of acceleration/fuel economy performance, given by a function \( c(a) \); see Eqn. (25) and Fig. 2. This formula is also modeled after the estimates in [39]. For simplicity, fixed costs of production are not modeled. We assume acceleration is bounded below by 2.5s, roughly the acceleration time of the Bugatti Veyron, and above by 15.5s, a reasonable upper bound on most new vehicles available in recent years.

There is one screening rule: an annual owning and operating cost constraint. Specifically, if the annual owning and operating costs are greater than some budget level \( B \), then the vehicle will not be considered. Annual owning and operating costs for a vehicle purchased on credit are modeled by

\[ k(a, p) = Rp + mg(a), \quad R = \frac{r(1 + r)^N}{(1 + r)^N - 1} \] (3)

where \( p \) is the vehicle purchase price, \( r \) is an annual interest rate, \( N \) is the number of loan periods, \( m \) is the number of miles driven per year, and \( p \) is the price of gasoline. An alternative interpretation of this model is that the vehicle is leased and \( Rp \) represents the amortized annual cost of the lease as a function of MSRP \( p \). With this definition, \( s(a, p) = k(a, p) - B \).
Consider-then-choose choice probabilities are then

\[ P^C(a, g(a), p) = \begin{cases} P(a, g(a), p) & \text{if } k(a, p) \leq B \\ 0 & \text{if } k(a, p) > B \end{cases} \]

where \( P(a, g, p) \) is the Logit choice probability (with an outside good)

\[ P(a, g, p) = \frac{e^{u(a, g, p)}}{1 + e^{u(a, g, p)}} \]

and the utility function \( u \) depends on acceleration, fuel consumption, and vehicle price as follows:

\[ u(a, g, p) = -3.6p - 36.8g + \frac{11.262}{a} + 23.2. \] (4)

This utility model is based on the model estimated in [39].

**Optimal Design-and-Pricing**

If the firm were to ignore consider-then-choose behavior, they would solve

\[
\text{maximize } P(a, g(a), p)(p - c(a)) \\
\text{with respect to } 2.5 \leq a \leq 15, \ p \geq 0
\] (5)

This is a smooth Nonlinear Program (NLP) and is easily solved with existing techniques. Acknowledging consider-then-choose behavior, however, requires solving

\[
\text{maximize } P^C(a, g(a), p)(p - c(a)) \\
\text{with respect to } 2.5 \leq a \leq 15, \ p \geq 0
\] (6)

an optimization problem with a discontinuous objective that cannot be solved with existing NLP techniques.

We apply an ad hoc approach suitable for this problem. Because profits are zero if the budget constraint is not satisfied \((k(a, p) > B)\) but are positive for any choice of \((a, p)\) such that \(p > c(a)\) and \(k(a, p) \leq B\), the budget constraint must be satisfied by the optimal design and price. Thus Eqn. (6) can be solved by solving the constrained NLP

\[
\text{maximize } P^C(a, g(a), p)(p - c(a)) \\
\text{with respect to } 2.5 \leq a \leq 15, \ p \geq 0 \\
\text{subject to } B - k(a, p) \geq 0
\] (7)

The contours of the objectives, the budget constraint, and the optimal solutions to Eqns. (5) and (7) are illustrated in Fig. 3. Solution details are given in Table 1. Eqn. (5) is solved by designing a vehicle with 0-60 acceleration time of 4.5 seconds (with a corresponding fuel economy of 10.2 mpg) and pricing this vehicle at roughly $55,100. The expected profits with this design and price, normalized by market size, are roughly $27,700. However, this vehicle does not satisfy the budget constraint. Thus, under consider-then-choose behavior, profits for this vehicle would be $0. Acknowledging consider-then-choose behavior by solving Eqn. (7) leads to a slightly higher price of $57,200, but a
TABLE 1. Optimal solutions to Eqns. (5), (7), and (8). All solutions computed in matlab using fmincon’s SQP algorithm. See also Fig. 3.

<table>
<thead>
<tr>
<th>Problem</th>
<th>(a) (s)</th>
<th>(e) (mpg)</th>
<th>(p) ($)</th>
<th>(\pi(a,p))</th>
<th>(\pi^C(a,p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignoring Consideration Egn. (5)</td>
<td>4.5</td>
<td>10.2</td>
<td>55,100</td>
<td>27,700</td>
<td>0</td>
</tr>
<tr>
<td>Including Consideration Egn. (7)</td>
<td>13.4</td>
<td>36.5</td>
<td>57,200</td>
<td>22,900</td>
<td>22,900</td>
</tr>
<tr>
<td>Ignoring Consideration with Egn. (8)</td>
<td>4.5</td>
<td>10.2</td>
<td>33,900</td>
<td>9,312</td>
<td>9,312</td>
</tr>
<tr>
<td>Recovery using Price Alone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 3. Contours of profits optimal solutions to Eqns. (5-8). Colors indicate magnitude of profits, with red being high and blue low.

A radically different design: acceleration should be 13.4 seconds (with a fuel economy of 36.5 mpg). Optimal profits for this vehicle are roughly $22,900, and though this is 17.3% lower than the optimal profits expected when ignoring consider-then-choose behavior, these expectations were incorrect. This illustrates our first point: significantly sub-optimal designs may be obtained if consideration behavior is ignored during product design.

Suppose the firm were to choose acceleration and price ignoring consider-then-choose behavior, but recognized this behavior existed once they tried to sell the vehicle that was thus designed (by, for example, interviewing potential customers through dealers or online). While the firm would have full flexibility over price (or advertising, or buy-here-pay-here interest rates), it is unlikely the firm could significantly change acceleration through vehicle re-design. Fixing acceleration at the optimal level from Eqn. (5) and optimizing profits over price alone leads to the following “profit recovery” problem:

\[
\begin{align*}
\text{maximize} & \quad P(4.5, g(4.5), p)(p - c(4.5)) \\
\text{with respect to} & \quad p \geq 0 \\
\text{subject to} & \quad k(4.5, p) - B \geq 0
\end{align*}
\]

The optimal price for Eqn. (8) is $33,900, corresponding to profits per vehicle of $9,312 (see Table 1). This is 66.4% lower than the (incorrectly) expected optimal profits from Eqn. (5), and 59.3% lower than the optimal profits for Eqn. (7) that would be obtained if the vehicle were originally designed with consideration behavior in mind. This illustrates our second point: the firm cannot recover lost profits by changing prices (or other marketing variables) alone.

4 GENERAL COMPUTATIONAL APPROACHES

The example in Section 3 justifies further attention to consider-then-choose models for use within engineering design. This section considers scalable numerical methods for the general optimal design problem with consider-then-choose models based on conjunctive screening rules. While the strategy for overcoming the discontinuity in the objective function of Eqn. (6) can be generalized, this strategy will become intractable as larger and more complex consumer populations, screening rules, and product portfolios are studied. We propose a potentially scalable computational method based on a novel MPCC relaxation of
the general optimal design problem with consider-then-choose models defined by conjunctive screening rules. Proofs of all significant results are given in Appendix C.

### The General Design-and-Pricing Problem

The general consider-then-choose optimization problem can be written as follows:

\[
\begin{align*}
\text{maximize} & \quad \pi^C(X, p) \\
\text{with respect to} & \quad 1_j \leq x_j \leq u_j, \quad p_j \geq 0 \text{ for all } j \\
\text{subject to} & \quad c_j^E(x_j) = 0, \quad c_j^F(x_j) \geq 0 \text{ for all } j
\end{align*}
\]

where \(\pi^C\) are expected profits normalized by the number of individuals \(I\),

\[
\pi^C(X, p) = \sum_{j=1}^I \sum_{i=1}^N \hat{P}_{ij}^C(X, p)(p_j - c_j(x_j)) - \frac{c^E}{T},
\]

\(x_j \in \mathbb{R}^N\) is a vector of design variables for product \(j\), \(1_j, u_j \in \mathbb{R}^N\) are lower and upper bounds on \(x_j\), \(p_j\) is the price of product \(j\), \(c_j(x_j) \geq 0\) is a unit cost as a function of design decisions, \(c_j^E : \mathbb{R}^N \rightarrow \mathbb{R}^M\) are equality constraints, \(c_j^F : \mathbb{R}^N \rightarrow \mathbb{R}^M\) are inequality constraints, and \(c^E\) are (constant) fixed costs for the firm. Competing firms and their products are easily incorporated but significantly complicate the notation. As pointed out in Section 3, the primary difficulty with Eqn. (9) is that \(\hat{P}_{ij}^C\) is not necessarily continuous, much less smooth, when \(P_{ij}\) is defined by Eqn. (2). Standard NLP techniques such as SQP or IP methods typically require the objective and constraint functions to be twice continuously differentiable [35], and thus do not obviously apply to Eqn. (9).

### A Bi-Level Approach

A first attempt to solve Eqn. (9), patterned on the ad hoc approach used in Section 3, is outlined as follows:

1. **Assume** a certain collection of consideration sets: that is, individual \(i\) considers products \(j_1, j_2, \ldots, j\) etc., for all \(i\).
2. Define a NLP for this collection by adding constraints representing the screening rule satisfaction and violation necessary to generate these exact considerations;
3. Find the optimal collection of consideration sets by solving the resulting NLPs over all feasible consideration patterns.

For example, in Section 3 the single vehicle could be considered or not. If the vehicle was not considered, profits were (trivially) zero, which was easily seen to be less than the consideration-constrained optimal profits.

This approach can be formalized for the general case as follows. Let \(\gamma \in \{0, 1\}^M\), and define

\[
\hat{P}_{ij}^C(\gamma, X, p) = \frac{\gamma_j \text{en}(x_j, p_j)}{1 + \sum_{k=1}^I \gamma_k \text{en}(x_k, p_k)}
\]

The quantity \(\gamma_j\) expresses whether individual \(i\) considers product \(j\) (\(\gamma_{ij} = 1\)) or not (\(\gamma_{ij} = 0\)). Thus, \(\gamma\) encodes an assumed collection of consideration sets. Note that there are \(2^M\) possible values of \(\gamma\) (step 1). To ensure the products designed and priced are consistent with an assumed collection of consideration sets, \(R + 1\) constraints are required for each individual and product: \(s_i(x_j, p_j) \leq 0\) if \(\gamma_{ij} = 1\) and \(\max_r \{s_r(x_j, p_j)\} > 0\) if \(\gamma_{ij} = 0\) (step 2). The former ensures that if we assume individual \(i\) considers product \(j\), then individual \(i\)’s screening rules are each satisfied by the characteristics and price of product \(j\). The latter ensures that if individual \(i\) does not consider product \(j\), then at least one of individual \(i\)’s screening rules is violated by the characteristics and price of product \(j\). Both of these constraints are specific to the assumption of conjunctive screening rules.

Given an assumed collection of consideration sets \(\gamma\), optimal designs and prices consistent with this collection solve

\[
\begin{align*}
\text{maximize} & \quad \hat{\pi}^C(\gamma, X, p) \\
\text{with respect to} & \quad 1_j \leq x_j \leq u_j, \quad p_j \geq 0 \text{ for all } j \\
\text{subject to} & \quad c_j^E(x_j) = 0, \quad c_j^F(x_j) \geq 0 \text{ for all } j \\
& \quad \begin{cases} 
\{s_i(x_j, p_j) \leq 0 \quad \text{if } \gamma_{ij} = 1 \\
\max_r \{s_r(x_j, p_j)\} > 0 \quad \text{if } \gamma_{ij} = 0 
\end{cases}
\end{align*}
\]

where

\[
\hat{\pi}^C(\gamma, X, p) = \sum_{j=1}^J \left( \sum_{i=1}^I \hat{P}_{ij}^C(\gamma, X, p) (p_j - c_j(x_j)) - \frac{c^E}{T} \right).
\]

The constraint \(\max_r \{s_r(x_j, p_j)\} > 0\) cannot be implemented in a conventional NLP setting. Some form of relaxation, such as \(\max_m \{s_m(x_j, p_j)\} \geq \varepsilon\) for some \(\varepsilon > 0\), is required to implement the strict inequality with existing NLP techniques. When \(M > 1\), the non-smoothness associated with the max operator will also require special treatment.

If the optimal value \(\hat{\pi}^{C+}(\gamma)\) of Eqn. (12) can be computed with standard NLP techniques, this value can be used as the objective for an upper-level discrete optimization problem

\[
\begin{align*}
\text{maximize} & \quad \hat{\pi}^{C+}(\gamma) \\
\text{with respect to} & \quad \gamma \in \{0, 1\}^M
\end{align*}
\]
over the assumed collection of consideration sets (step 3). Again, this reduces to the approach in Section 3 for a single product and a single screening rule. This discrete optimization could conceivably be solved with heuristic search methods such as simulated annealing, pattern search, or evolutionary algorithms. We investigate a version problem that might be suitable for relaxation methods, e.g. Branch-and-Bound [40], below.

Two issues suggest that this approach will be practical only for small problems or problems with specific structure that is known and exploited a priori. First, the $IJ$ binary variables in Eqn. (13) may result in intractable discrete optimization problems for moderate $I$ and $J$, even if the Eqn. (12) can be solved efficiently—and globally—for any $γ$. Even for models with a single type of individual ($I = 1$), the exponential growth in the number of collections of consideration sets along with the growing complexity of Eqn. (12) threatens to quickly make this approach intractable for realistic product portfolios. For example, Ford offered 70 distinct cars and trucks in their 2012 model year, according to EPA [41]. The corresponding instance of Eqn. (13) would have 701 binary variables and roughly $10^{211}$ distinct values of $γ$. Recalling that every evaluation of $\hat{π}^{C,*}(γ)$ requires (globally) solving Eqn. (12) suggests that this approach will not be feasible at this scale, even for a single type of individual ($I = 1$), unless the underlying design problems (Eqn. (12)) can be solved very efficiently.

Second, it may be challenging to determine whether the NLP (12) is even feasible for given $γ$. Suppose, for example, that individuals $i$ and $i'$ use the same screening rules (but possibly make different compensatory decisions over those products considered). Then the NLP corresponding to any $γ$ with $γ_{i,j} = 1$ and $γ_{i',j} = 0$ for any $j$ is infeasible, as is the NLP with $γ_{i,j} = 0$ and $γ_{i',j} = 1$. If there are many screening rules or some screening rules are nonlinear (as in Section 3) verifying feasibility of a particular collection of consideration sets could be an intractable feasibility program. An extension of the vehicle design example to arbitrary numbers of individuals (Section 6) has sufficient structure to determine whether any collection of consideration sets is feasible; however doing so requires solving a linear program with as many constraints as individuals. Moreover, in this example a tiny percentage of the possible collections of consideration sets are feasible: when there are 20 individuals, less than 0.01% of the possible collections of consideration sets are feasible; when there are 50 individuals, less than $10^{-10}$ are feasible. Thus, unless it is remarkably easy to detect feasibility, ensuring that $\hat{π}^{C,*}(γ)$ even exists could impede efficient solution of Eqn. (13) in the general case.

**Relaxing the Choice Probabilities**

The constrained NLP approach was based on one relaxation of the choice probabilities, Eqn. (11). A different relaxation of the choice probabilities overcomes some of the difficulties observed in the previous section, and yields the two continuous relaxations of Eqn. (9) discussed below.

For any $ω = \{ω_{i,j,r} : 1 ≤ i ≤ I, 1 ≤ j ≤ J, 1 ≤ r ≤ R_i\}$, let

$$P_{i,j}^C(\omega, Y, p) = \frac{\left(\prod_{r=1}^{R_i} ω_{i,j,r}\right) e^{ω_{i,j,r}(y_j,p_j)}}{1 + \sum_{j=1}^{J} \left(\prod_{r=1}^{R_i} ω_{i,j,r}\right) e^{ω_{i,j,r}(y_j,p_j)}},$$

(14)

Associate $ω_{i,j,r}$ with product $j$ satisfying individual $i$’s $r$th screening rule, and $ω_{i,j,r} = 0$ with product $j$ violating individual $i$’s $r$th screening rule. The products $\prod_{r=1}^{R_i} ω_{i,j,r}$ are thus analogous to the $γ_{i,j}$ terms in Eqn. (11): $\prod_{r=1}^{R_i} ω_{i,j,r} = 1$ means individual $i$ considers product $j$, and $\prod_{r=1}^{R_i} ω_{i,j,r} = 0$ means individual $i$ does not consider product $j$. Let $Ω$ denote the set of all such vectors $ω$ such that $ω_{i,j,r} ∈ [0, 1]$ for all $1 ≤ i ≤ I, 1 ≤ j ≤ J, 1 ≤ r ≤ R_i$.

**Definition 1.** $ω ∈ Ω$ is called (strictly) choice-consistent with $(X, p)$ if $ω_{i,j,r} = 1$ for all $(i,j,r)$ such that $s_{i,r}(x_j, p_j) ≤ 0$ and $ω_{i,j,r} = 0$ for all $(i,j,r)$ such that $s_{i,r}(x_j, p_j) > 0$. $ω$ is called almost choice-consistent with $(X, p)$ if $ω_{i,j,r} = 1$ for all $(i,j,r)$ such that $s_{i,r}(x_j, p_j) < 0$ and $ω_{i,j,r} = 0$ for all $(i,j,r)$ such that $s_{i,r}(x_j, p_j) > 0$.

Formally, choice consistency also depends on the screening rules; we leave this dependency implicit.

**Lemma 1. (i)** For any $(X, p)$, there is a unique vector $\bar{ω}(X, p) ∈ Ω$ that is strictly choice-consistent with $(X, p)$. (ii) If $ω$ is strictly choice-consistent at $(X, p)$ then $P_{i,j}^C(X, p) = P_{i,j}^C(\bar{ω}(X, p), X, p)$. (iii) Subsequently, $π^C(X, p) = \hat{π}^C(\bar{ω}(X, p), X, p)$ where

$$\hat{π}^C(ω, X, p) = \sum_{j=1}^{J} \left(\sum_{i=1}^{I} P_{i,j}^C(ω, X, p)\right) \left(p_j - c_j(x_j)\right) - \frac{c^F}{I},$$

(16)

**An Mixed Variable Relaxation**

The first relaxation of Eqn. (9) based on Eqn. (14) is the following Binary Nonlinear Program (BNLP):

Maximize

$$\hat{π}^C(ω, X, p)$$

with respect to $ω_{i,j,r} ∈ \{0, 1\}$ for all $i, j, r$

$$l_j ≤ x_j ≤ u_j, \ p_j ≥ 0 \text{ for all } j$$

subject to

$$c^F_j(x_j) = 0, \ c^F_j(x_j) ≥ 0 \text{ for all } j$$

$$-ω_{i,j,r}s_{i,r}(x_j, p_j) ≥ 0$$

$$\left(1 - ω_{i,j,r}\right)s_{i,r}(x_j, p_j) ≥ 0$$

∀ $i, j, r$

(15)

The constraints in Eqn. (15) have the following purpose:
Lemma 2. If \((\omega, X, p)\) is feasible for Eqn. (15), then \(\omega\) is almost choice-consistent with \((X, p)\).

Note, however, that some choice-inconsistent \(\omega\)'s are feasible for Eqn. (15), because the constraints allow \(\omega_{i,j,r} = 0\) when \(s_{i,r}(x_j, p_j) = 0\).

Eqn. (15) is a relaxation of Eqn. (9) in the following sense:

Lemma 3. \(\pi_{R*} \geq \pi_{C*}\) where \(\pi_{R*}\) is the optimal value of Eqn. (15) and \(\pi_{C*}\) is the optimal value of Eqn. (9). Moreover, if \((\omega, X, p)\) is a local solution of Eqn. (15) and \(\omega\) is strictly choice-consistent with \((X, p)\), then \((X, p)\) is a local solution to Eqn. (9).

The relationship between Eqn. (15) and Eqn. (9) is analogous to LP relaxations of ILPs: An integer solution to the LP is a solution to the ILP, but solutions to the LP are not always solutions to the relaxed LP [42]. ILP strategies that use LP relaxations solve the relaxations and then add constraints to progressively enforce integrality in the LP solution.

Branch-and-Bound (B&B) heuristics may be an appropriate solution strategy because the binary variables in Eqn. (15) are not "categorical" [43] and thus Eqn. (15) admits continuous relaxations. (Note that because Eqn. (15) is non-convex, B&B techniques are global optimization heuristics [40].) One potential difficulty with this approach could be the resulting problem size: there are \(J\sum_{i=1}^{l} R_i\) new variables and \(2J\sum_{i=1}^{l} R_i\) new inequality constraints that form this relaxation, even before including constraints that would be added to force the \(\omega\)'s into \(\{0, 1\}\). Moreover, the continuous relaxations for Eqn. (15) will, implicitly, have complementarity constraints [44]: if \(\omega_{i,j,r} \in [0, 1]\), then the constraints \(\omega_{i,j,r} s_{i,r}(x_j, p_j) \leq 0\) and \((1 - \omega_{i,j,r}) s_{i,r}(x_j, p_j) \geq 0\) mean that

\[
\begin{align*}
s_{i,r}(x_j, p_j) < 0 & \iff \omega_{i,j,r} = 1 \\
s_{i,r}(x_j, p_j) = 0 & \iff \omega_{i,j,r} \in [0, 1] \\
s_{i,r}(x_j, p_j) > 0 & \iff \omega_{i,j,r} = 0
\end{align*}
\]

These relations are commonly called a "Mixed Complementarity Problem" (MCP) and denoted by \(0 \leq \omega_{i,j,r} \leq 1 \perp s_{i,r}(x_j, p_j)\) [45–49]. Mathematical Programs with Complementarity Constraints (MPCCs) are challenging mathematical programs, as discussed further below in Section 5. Existing Branch & Bound techniques for solving Eqn. (15) will have difficulties with the underlying NLPs without special treatment of the implicit complementarity constraints in the continuous relaxations of Eqn. (15).

An Explicit MPCC Relaxation

Because complementarity constraints must be addressed to solve Eqn. (15), we must investigate the associated MPCC:

\[
\begin{align*}
\max \quad & \hat{\pi}^C(\omega, X, p) \\
\text{subject to} \quad & a_{k,j,r} \in [0, 1] \text{ for all } i, j, r \\
& b_j \leq x_j \leq u_j, \quad p_j \geq 0 \text{ for all } j \\
& c^L_j(x_j) = 0, \quad c^L_j(x_j) \leq 0 \text{ for all } j \\
& 0 \leq \omega_{i,j,r} \leq 1 \perp s_{i,r}(x_j, p_j) \\
& \text{for all } i, j, r
\end{align*}
\]

Eqn. (16) is the continuous relaxation of Eqn. (15) before constraints are added to force the \(\omega\)'s to take binary values. Being able to efficiently solve Eqn. (16) may be considered a prerequisite to efficiently solving Eqn. (15).

Like Eqn. (15), Eqn. (16) has the following properties:

Lemma 4. If \((\omega, X, p)\) is feasible for Eqn. (16), then \(\omega\) is almost choice-consistent with \((X, p)\).

Lemma 5. \(\hat{\pi}^{CC*} \geq \pi_{C*}\) where \(\hat{\pi}^{CC*}\) is the optimal value of Eqn. (16) and \(\pi_{C*}\) is the optimal value of Eqn. (9). Moreover, if \((\omega, X, p)\) is a local solution of Eqn. (16) and \(\omega\) is strictly choice-consistent with \((X, p)\), then \((X, p)\) is a local solution to Eqn. (9).

Directly solving Eqn. (16) may avoid the need to enforce integrality with B&B techniques. However, B&B techniques will probably encourage convergence to local optima, and thus should be considered if the MPCC relaxation can be solved reliably and efficiently.

5 SOLVING MPCC RELAXATIONS

MPCCs are hard NLPs largely because standard constraint qualifications fail to hold at any feasible point [44]. Specifically, the active constraint gradients (including bounds) cannot be linearly independent at any feasible point. This property can be easily seen in the NLP form of the MPCC relaxation for our simple example, Eqn. (??) below. Related problems are that the feasible region may not have a topological interior, making convergence difficult for IP methods, and that local linearizations may be inconsistent, making convergence difficult for SQP methods [50, 51].

Solving MPCCs as NLPs

The mathematical programming literature has dealt with many of these challenges, and identified the appropriate ways in which standard NLP solvers can be used to solve MPCCs; see, e.g., [50, 51] or the review in [44]. IP methods can be used
by either relaxing the constraints corresponding to the MPCC, thus creating a (relaxed) feasible region with a topological interior [52–54], or by including complementarity in the objective, instead of the constraints, through penalty terms [50]. With the right problem formulation, SQP algorithms can be superlinearly convergent to “strongly stationary” solutions to MPCCs [51]. SQP algorithms with an “elastic mode” (see [34]) turn out to be an efficient solution to the problem of inconsistent constraint linearizations, and can ensure global convergence of SQP methods for some MPCCs [55].

Most of the recent mathematical literature justifying the use of IP and SQP methods on MPCCs assume the problem has complementary variables [44, 50, 51]. While Eqn. (16) is not of this form, slack variables can be added to convert the problem to one with complementary variables:

\[
\begin{align*}
\text{maximize} & \quad \hat{\pi}^C(w, X, p) \\
\text{with respect to} & \quad w_{ij,r}, v_{ij,r}, y_{ij,r}, z_{ij,r}, s_{ij,r}^+, s_{ij,r}^- \geq 0 \text{ for all } i, j, r \\
\text{subject to} & \quad c^E_j(x_j) = 0, c^f_j(x_j) \leq 0 \text{ for all } j \\
& \quad s_{ij,r}(x_j, p_j) - s_{ij,r}^- + s_{ij,r}^+ = 0 \\
& \quad w_{ij,r} + v_{ij,r} - 1 = 0 \\
& \quad y_{ij,r} - s_{ij,r}^- = 0 \\
& \quad z_{ij,r} - s_{ij,r}^+ = 0 \\
& \quad w_{ij,r} \perp s_{ij,r}^+ \quad (w_{ij,r}, s_{ij,r}^+ \leq 0) \\
& \quad v_{ij,r} \perp s_{ij,r}^- \quad (v_{ij,r}, s_{ij,r}^- \leq 0) \\
& \quad y_{ij,r} \perp z_{ij,r} \quad (y_{ij,r}, z_{ij,r} \leq 0)
\end{align*}
\]

(17)

See also Eqns. (18) and (19) below. KNITRO’s tools for solving MPCC’s allow the user to state which variables are complementary to one another, and thus require the problem to be in complementary-variable form [50]. Solving MPCCs with SNOPT in a manner consistent with existing convergence theory (e.g., [51]) requires entering the complementarity conditions as bilinear inequalities (see Eqn. (17) or (19)).

Trivial KKT Points

One potential concern with solving the MPCC relaxations is that they can have “trivial” KKT points. For example, if \( \omega_{i,j,r} = 0 \) for all \((i, j, r)\), then profits vanish on a neighborhood of any \( X \) and thus any feasible \( X \) may satisfy the KKT conditions; see Appendix D for a derivation in the case of our simple example. Our computational results demonstrate that such points can be computed by both SQP and IP methods. We have had some success in using regularization techniques, particularly Tikhonov regularization (e.g., [56, 57]), to eliminate convergence to trivial KKT points but do not detail this approach here.

Verifying Choice Consistency

When some screening rule is “active” in the sense that \( s_{ij}(x_j, p_j) \leq 0 \) but \( s_{ij}(x_j, p_j) = 0 \) for some \( r \) at a solution to the MPCC relaxation, the MCP constraints only require that \( \omega_{i,j,r} \in [0, 1] \). To relate local solutions to the MPCC relaxation (16) to local solutions of the original problem, Eqn. (9), the values of \( \omega \) at a solution must be strictly choice consistent. Practically, this poses a numerical termination problem: if \( s_{ij}(x_j, p_j) \approx 0 \), then \( \omega_{i,j,r}s_{ij}(x_j, p_j) \approx 0 \) for any \( \omega_{i,j,r} \in [0, 1] \). In our numerical trials we have found that weak feasibility tolerances result in computation of “solutions” that are not strictly choice-consistent, but that very tight feasibility tolerances promote solver failure.

Our solution to this conundrum is to transition from the MPCC relaxation Eqn. (16) to the NLP Eqn. (12) when a solution that is not strictly choice consistent is computed. Specifically, suppose that \( s_i(x_j, p_j) \leq 0 \) but \( s_i(x_j, p_j) = 0 \) for some \( r \) and \( \omega_{i,j,r} \in [0, 1] \); i.e., the computed solution is not strictly choice-consistent. We set \( \omega_{i,j,r} = 1 \) for all \( \omega_{i,j,r} < 1 \) for which \( s_{ij}(x_j, p_j) = 0 \), form \( \gamma \) by taking \( \gamma_{i,j} = \prod_r \omega_{ij,r} \), and solve Eqn. (12) with this \( \gamma \) to “refine” the solution. If successful, this results in either a strictly choice-consistent solution or a solution that has some \( \omega_{i,j,r} = 0 \) and \( s_{ij}(x_j, p_j) = 0 \) (\( \omega_{i,j,r} \) does not change, but \( s_{ij}(x_j, p_j) \) could be driven to zero for some such \( \omega \)). In the former case, the same process can be repeated until either (a) the solver fails or (b) a strictly choice consistent solution is found. All subsequent solutions will have \( \omega_{i,j,r} \in \{0, 1\} \) for all \((i, j, r)\), and because the only changes to the values of \( \omega \) are to take values that were 0 to 1, this process cannot cycle. This strategy is thus guaranteed to either compute a strictly choice-consistent solution or encounter a solver failure in a finite number of steps.

Computational Results for the Budget-Constrained Vehicle Choice Example

The MPCC relaxation of Eqn. (6) is

\[
\begin{align*}
\text{maximize} & \quad \hat{\pi}^C(w, a, g(a), p)(p - c(a)) \\
\text{with respect to} & \quad \omega, 2.5 \leq a \leq 15, p \geq 0 \\
\text{subject to} & \quad 0 \leq \omega \leq 1 - s(a, p)
\end{align*}
\]

(18)

Because there are only three variables, this problem admits an illustration of how the MPCC relaxation makes Eqn. (6) solvable. Fig. 4 plots profits over a surface defining the relaxed feasible region in Eqn. (18). Conceptually speaking, the introduction of \( \omega \) extends the problem into an additional dimension and interpolates profits along the discontinuity in the choice probabilities caused by the screening rule.

Consistent with the literature on solving MPCCs, we solve a version of Eqn. (18) for which the complementarity constraints are formulated as complementarity conditions between problem
FIGURE 4. Profits (colors) plotted over a surface representing the feasible region for Eqn. (18), the MPCC relaxation of Eqn. (6). Adding \( \omega \) and the MCP constraint \( 0 \leq \omega \leq 1 \perp s(a,p) \) leaves the smooth regions of the profit function unchanged, but interpolates profits over the discontinuity introduced by the screening rule.

variables:

\[
\begin{align*}
\text{maximize} & \quad \hat{P}^C (w, a, g(a), p) (p - c(a)) \\
\text{with respect to} & \quad 2.5 \leq a \leq 15, \ p \geq 0 \\
& \quad w, v, y, z, s_+, s_- \geq 0 \\
\text{subject to} & \quad 1 - w - v = 0 \\
& \quad s(a, p) - s_+ + s_- = 0 \\
& \quad y - s_+ = 0 \\
& \quad z - s_- = 0 \\
& \quad w \perp s_+ \quad (ws_+ \leq 0) \\
& \quad v \perp s_- \quad (vs_- \leq 0) \\
& \quad y \perp z \quad (yz \leq 0)
\end{align*}
\]

\( (19) \)

KNITRO’s tools allow the user to state which variables are complementary to one another, and thus the last three constraints are not literally programmed. Solving Eqn. (19) with SNOPT requires entering the last three constraints as the bilinear inequalities shown in parentheses.

Table 2 provides statistics of 10,000 attempted solves of Eqns. (7) and (19) using \text{matlab} (version 7.10), KNITRO (version 8), and SNOPT (version 7). All computations reported in this article were undertaken on a single Mac Pro tower with 2, quad-core 2.26GHz processors and 32 GB of RAM running OS X (10.6.8). All solvers perform well for Eqn. (7), a standard inequality constrained NLP. KNITRO’s complementarity tools for Eqn. (19) have some difficulties, solving the problem in only 53% of the trials. SNOPT’s elastic-mode SQP techniques result in more reliable computations: the true solution was computed in 99% of the trials with only 3% of the runs resulting in a trivial KKT point. It is possible that KNITRO’s Active Set strategy would have better performance.

6 AN EXAMPLE WITH HETEROGENEOUS SCREENING RULES

While the MPCC relaxation applies to the simple example discussed in Section 3, using a standard inequality constrained NLP resulted in (marginally) more reliable computations. As discussed in Section 4, however, this two-level NLP approach may not easily generalize to problems with multiple products or heterogeneous populations in which individuals have distinct screening rules. This section extends the simple example to provide a demonstration of the applicability of the MPCC relaxation approach to problems with heterogeneous screening rules and compensatory preferences.

The Model

Suppose there are \( I \in \mathbb{N} \) individuals, and let each individual \( i \in \{1, \ldots, I\} \) have an individual-specific screening rule

\[
s_i(a, p) = R_i p + m_i p_i^G g(a) - B_i
\]

for individual-specific interest rate \( r_i \), annual miles travelled \( m_i \), gasoline price \( p_i^G \) and annual budget level \( B_i \). Each individual

<table>
<thead>
<tr>
<th>Solver</th>
<th>Eqn.</th>
<th>Success</th>
<th>Trivial(a)</th>
<th>Failure(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{matlab} (SQP)</td>
<td>(7)</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>(19)</td>
<td>95%</td>
<td>4%</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>\text{KNITRO} (IP)</td>
<td>(7)</td>
<td>99%</td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>(19)</td>
<td>53%</td>
<td>24%</td>
<td>23%</td>
<td></td>
</tr>
<tr>
<td>\text{SNOPT} (SQP)</td>
<td>(7)</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>(19)</td>
<td>99%</td>
<td>0.4%</td>
<td>0.04%</td>
<td></td>
</tr>
</tbody>
</table>

(a) Solver terminated successfully but computed a solution with \( \omega = 0 \) (and thus \( \pi^C = 0 \)). See Appendix D.

(b) Solver failed to converge.
also has different compensatory preferences over fuel economy and acceleration. Our utility function is a variant of the utility in the Mixed Logit model estimated by Whitefoot et al. [39]:

\[ u_i(a, g, p) = \beta_{p,i}p + \beta_{g,i}g + \beta_{a,i}a + \vartheta_i. \]

The coefficients in this equation are as follows: \( \beta_{p,i} = -[4.591 + 0.1756/t_i - 0.377N_{p,i}] \) where \( t_i \) denotes household income; \( \beta_{g,i} = -36.77 + 2.2N_{g,i}; \beta_{a,i} = 11.262 + 0.321N_{a,i}; \) and \( \vartheta_i = 23.178 + 0.5N_{0,i}. \) \( N_{p,i}, N_{g,i}, N_{a,i}, \) and \( N_{0,i} \) each denote independent samples from a standard normal distribution.

The individualized budget constraints are sampled as follows: We sample interest rates \( r_i \) from a uniform distribution on \([0.03, 0.08]\) (3%-8% per year). We restrict income to be larger than 50,000$, assuming that new car purchases are dominated by households having annual incomes larger than this value, and sample \( t_i \) from an empirical frequency distribution based on data in the 2006 Current Population Survey (CPS). This income is applied in both the utility model and estimates of miles driven and annual budget. Annual miles driven are assumed to follow a power law with income: \( m_i = 6639_{t_i}^{0.288} \cdot 0.253N_{m,i} \) where \( N_{m,i} \) is a sample from a standard normal distribution. The coefficients were estimated based on the data in the National Household Transportation Survey. The annual budget is assumed to follow a linear relationship with income: \( B_i = 0.5 + 0.06t_i + 0.0512N_{B,i} \) where again \( N_{B,i} \) is a sample from a standard normal distribution. The coefficients in this model were estimated based on data in the Consumer Expenditure Survey. \( N \) and \( N^G \) are assumed to be fixed at 10 and 3.5, respectively, for all individuals.

The optimal vehicle design problem with this heterogeneous population is then

\[
\begin{aligned}
\text{maximize } & \left( \sum_{i=1}^{I} P^c_i(w_i, a, g(a), p) \right) (p - c(a)) \\
\text{with respect to } & 2.5 \leq a \leq 15, \ p \geq 0 \\
\text{subject to } & p - c(a) \geq 0
\end{aligned}
\]

We solve the complementarity variables formulation given in Eqn. (21). In this formulation, the symbol “\( \circ \)” denotes component-wise multiplication of two vectors, and all bold face vectors are in \( \mathbb{R}^I \). We add \( p \geq c(a) \) as a constraint to avoid computing local solutions that have negative profits because prices are less than costs. Such points can never be globally profit-optimal, but may be locally optimal for some budget constraints that can be sampled from the distributions specified above.

\[
\begin{aligned}
\text{maximize } & \left( \sum_{i=1}^{I} P^c_i(w_i, a, g(a), p) \right) (p - c(a)) \\
\text{with respect to } & 2.5 \leq a \leq 15, \ p \geq 0 \\
\text{subject to } & p - c(a) \geq 0
\end{aligned}
\]

\( I \), \( w, v, y, z, s_+, s_- \geq 0 \)

\( 1 - w - v = 0 \)

\( y - s_+ = 0 \)

\( z - s_- = 0 \)

\( w \circ s_+ \leq 0 \)

\( v \circ s_- \leq 0 \)

\( y \circ z \leq 0 \)

### Computational Results with the MPCC Relaxation

Table 3 provides problem data and solution statistics for 10,000 attempted solves of Eqn. (21) using SNOPT for \( I = 5, 10, 15, 20, 25, 30, 35, 40, 45 \) and 50. These problems represent relatively small-scale NLPs with less than around 300 variables and constraints (see Table 3). However, the corresponding space of collections of consideration sets (values of \( \omega \in \{0,1\}^I \) can be quite large; when \( I = 50 \), there are roughly \( 10^{15} \) potential consideration sets. Unlike the example in Section 3, in which we assumed a single type of screening rule, problems with multiple screening rules may have several non-trivial local solutions; see Fig. 5. However, the number of local solutions appears to grow linearly in \( I \), while the number of collections of consideration sets grows exponentially in \( I \). SNOPT tends to terminate very quickly, taking 2-13 \( \mu s \) (on average) for Eqn. (21) as \( I \) grows from 5 to 50. This linear scalability in compute time is very promising for fast, large-scale computations, if SNOPT often converges to “good” solutions to Eqn. (20).

In contrast to our experience for the single individual example, Eqn. (21) provided more robust computations of local solutions to Eqn. (20). The success rate, defined as successful termination at a locally optimal solution to Eqn. (20), is over 70% for \( I \leq 30 \), but decreases to just over 50% with \( I = 50 \). Compared to the single-individual example considered above, there are significant fractions of runs that compute trivial KKT points (\( \omega_i = 0 \) for all \( i \)), points that are locally optimal for Eqn. (21) but not locally optimal with respect to Eqn. (20), and solver failures. SNOPT applied to Eqn. (21) is initially very robust, with 97% and 86% success rates for \( I = 5 \) and 10, but becomes unreliable as \( I \) grows resulting in only 5% success when \( I = 50 \). These failures are a consequence of computing locally sub-optimal solutions. SNOPT applied to Eqn. (21) almost never computes a
trivial KKT point and fails to converge at roughly the same rate as SNOPT applied to Eqn. (??). Future work will deal with these issues to achieve more robust solution of the MPCC relaxations.

The Bi-Level NLP Approach

Feasibility of an assumed collection of consideration sets is the first problem that must be addressed in considering solution of Eqn. (20) using the bi-level constrained NLP approach. In this specific case, feasibility can be determined algorithmically through a linear feasibility problem: Because $g$ is one-to-one over the domain of feasible $a$ (Fig. 1), a weak version of feasibility is the following: there exists $(p, g)$, $p \geq 0$ and $g \in \text{range}(g(a)) = [l, u]$ such that $R_i p + m_i g^\text{c} - B_i \leq 0$ if $\omega_i = 1$, and $R_i p + m_i g^\text{c} - B_i \geq 0$ if $\omega_i = 0$ for all $i$. These conditions are equivalent to the linear feasibility problem

$$\text{minimize } 0$$
$$\text{with respect to } l \leq g \leq u, \ p \geq 0$$
$$\text{subject to } \Sigma \left( A \begin{bmatrix} p \\ g \end{bmatrix} - B \right) \geq 0$$

(22)

where $l = \min\{\text{range}(g(a))\}$, $u = \max\{\text{range}(g(a))\}$, $\Sigma$ is the $I \times I$ diagonal matrix with diagonal entries $\sigma_i = 1$ if $\omega_i = 0$ and $\sigma_i = -1$ if $\omega_i = 1$, and $A$ is the $I \times 2$ matrix with columns $R_i = (R_1, \ldots, R_I)$ and $p^\text{c} = p^\text{c}(m_1, \ldots, m_I)$. Any good linear programming solver could be applied to identify feasibility of $\omega$ by solving this problem; SNOPT applies the primal simplex method for linear programs [34].

Intelligent search techniques will ultimately require solution of Eqn. (20). Because each individual has a single screening rule ($R_i = 1$), we associate $\gamma$ and $\omega$ and refer only to $\omega$ below. If a feasible point for Eqn. (22) cannot be found (and thus Eqn. (20) is infeasible), we set $\pi^*(\omega) = -\infty$. If, on the other hand, $\omega$ is such that Eqn. (20) is feasible, we estimate $\pi^*(\omega)$ by solving

$$\text{maximize } \left( \sum_{i=1}^{I} p_i^\text{c}(w_i(a), g(a), p) \right) (p - c(a))$$
$$\text{with respect to } 2.5 \leq a \leq 15, \ p \geq 0$$
$$\text{subject to } s_i(a, p) \leq 0 \text{ for all } i \text{ such that } \omega_i = 1$$
$$s_i(a, p) \geq 0 \text{ for all } i \text{ such that } \omega_i = 0$$

(23)

Strictly speaking, there is no guarantee that successful solution of Eqn. (23) gives the global maximum because Eqn. (23) is a nonlinear program. Moreover there is no guarantee that solving Eqn. (23) results in a strictly choice-consistent point: an NLP solver could, in principle, terminate at some $(a, p)$ with $s_i(a, p) = 0$ for some $i$ with $\omega_i = 0$. If solution of Eqn. (23) is not strictly choice consistent we force the values of $\omega$ to be choice consistent at the current values of $(a, p)$, correspondingly re-define the current $\omega$, and re-evaluate the associated profits to obtain $\pi^*(\omega)$. We do not re-solve Eqn. (23) with the modified value of $\omega$.

Evaluations of $\pi^*(\omega)$ are used to solve the upper level discrete problem

$$\text{maximize } \pi^*(\omega) \text{ with respect to } \omega \in \{0, 1\}^I.$$

We apply a simple genetic algorithm with proportional-fitness selection, one-point crossover, and random single-bit-flip mutation (at 75% probability). Early tests indicated that randomly sampling initial populations with a significant fraction feasible values of $\omega$ was virtually impossible to do by directly sampling $\omega$ from $I$ independent Bernoulli distributions (with $p = 1/2$). Instead, we select members of a feasible initial population by randomly drawing an acceleration-price pair $(a, p)$, and then letting the collection of consideration sets corresponding to this pair be a member of the initial population.

Currently, we have found the comparison of this evolutionary optimization approach to the MPCC relaxation mixed. Running populations with 20 collections of consideration sets for 50 generations takes significantly longer than solving the MPCC relaxations. For example, when $I = 50$, running 50 generations requires roughly 2.5 s, while running the MPCC relaxation takes 0.03 s (on average). However, the full run of the evolutionary algorithm is virtually assured to find the global optimum, while computing the global optimum using the MPCC relaxation is likely to happen only for a small fraction of initial conditions. Moreover the evolutionary approach often finds the global optimum, or at least a local optimum with similar profits, fairly quickly relative to the total time required to run all 50 generations. Future work will investigate this comparison more closely, and quantify the probability of computing the global optimum in a given amount of time using evolutionary strategies and MPCC relaxation with multistart.

7 CONCLUSIONS

This article has introduced consider-then-choose models, and extension of DCA to include non-smooth compensatory decision processes, to engineering design optimization. Including such models is justified by a simple vehicle design example in which ignoring consideration behavior leads to sub-optimal designs. To support further research in this area, we have undertaken an investigation of numerical methods suitable to a general formulation of optimization with consideration defined by conjunctive screening rules. Two methods are discussed: a discrete optimization approach and a novel MPCC relaxation. Different methods for solving the MPCC relaxation are tested on two problem types to examine convergence behavior. Future work will take several directions including coupling of the MPCC relax-
FIGURE 5. Contours of profits (including consider-then-choose behavior) for the multiple individual model with $I = 5$ (left) and $I = 10$ (right). Dashed black line illustrates costs as a function of fuel economy. Solid black lines denote the curves that exactly satisfy the individual budget constraints; considerable products for each individual lie on or to the left of that individual’s curve. The solid black dots denote the local optima computed by the MPCC relaxation method. For $I = 10$, one of these local optima is in the interior region between two curves (though this is difficult to make out).

ACKNOWLEDGMENT

The authors wish to thank Iowa State University for supporting this research.

References

REFERENCES


TABLE 3. Statistics for 10,000 attempted solves of Eqn. (21) using SNOPT for various values of $I$. For each value of $I$, a single set of samples of $i, r, m, B_i$ and the random utility coefficients are drawn as specified in the text. Different trials for each $I$ are started at randomly drawn initial conditions, using the same set of samples.

<table>
<thead>
<tr>
<th>Number of Individuals ($I$)</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections of Choice Sets (#)</td>
<td>$10^{1.5}$</td>
<td>$10^{3}$</td>
<td>$10^{4.5}$</td>
<td>$10^{6}$</td>
<td>$10^{7.5}$</td>
<td>$10^{9}$</td>
<td>$10^{10.5}$</td>
<td>$10^{12}$</td>
<td>$10^{13.5}$</td>
<td>$10^{15.1}$</td>
</tr>
<tr>
<td>Statistics from 100,000 solves of Eqn. (23) starting at randomly drawn initial conditions using SNOPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feasible Collections of Choice Sets (#)</td>
<td>10</td>
<td>26</td>
<td>48</td>
<td>77</td>
<td>108</td>
<td>133</td>
<td>158</td>
<td>207</td>
<td>257</td>
<td>327</td>
</tr>
<tr>
<td>Locally Optimal Solutions (#)</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>13</td>
<td>11</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>Optimal Profits ($10^4$)</td>
<td>1.60</td>
<td>1.38</td>
<td>1.15</td>
<td>1.16</td>
<td>1.22</td>
<td>1.19</td>
<td>1.10</td>
<td>1.10</td>
<td>1.15</td>
<td>1.10</td>
</tr>
<tr>
<td>“Volume” of Optimal Consideration Set (%)</td>
<td>50.1</td>
<td>12.7</td>
<td>6.8</td>
<td>6.6</td>
<td>6.1</td>
<td>3.1</td>
<td>1.9</td>
<td>1.9</td>
<td>1.9</td>
<td>1.5</td>
</tr>
<tr>
<td>Statistics from 10,000 attempted solves of Eqn. (21) starting at randomly drawn initial conditions using SNOPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Problem Variables (#)</td>
<td>32</td>
<td>62</td>
<td>92</td>
<td>122</td>
<td>152</td>
<td>182</td>
<td>212</td>
<td>242</td>
<td>272</td>
<td>302</td>
</tr>
<tr>
<td>Number of Problem Constraints (#)</td>
<td>36</td>
<td>71</td>
<td>106</td>
<td>141</td>
<td>176</td>
<td>211</td>
<td>246</td>
<td>281</td>
<td>316</td>
<td>351</td>
</tr>
<tr>
<td>Number of Local Solutions Computed (#)</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>13</td>
<td>15</td>
<td>16</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>Maximum Profits Computed ($10^4$)</td>
<td>1.60</td>
<td>1.38</td>
<td>1.15</td>
<td>1.16</td>
<td>1.22</td>
<td>1.19</td>
<td>1.10</td>
<td>1.10</td>
<td>1.15</td>
<td>1.08</td>
</tr>
<tr>
<td>Average Compute Time (all runs) ($\mu s$)</td>
<td>4.1</td>
<td>9.3</td>
<td>10.9</td>
<td>15.4</td>
<td>18.6</td>
<td>23.8</td>
<td>26.6</td>
<td>33.3</td>
<td>40.8</td>
<td>49.5</td>
</tr>
<tr>
<td>Successful Solves (%)</td>
<td>99.7</td>
<td>99.4</td>
<td>96.6</td>
<td>95.8</td>
<td>97.7</td>
<td>97.4</td>
<td>97.9</td>
<td>96.6</td>
<td>97.3</td>
<td>92.9</td>
</tr>
<tr>
<td>Terminated at a Trivial KKT Point (%)</td>
<td>0.03</td>
<td>0.03</td>
<td>0.20</td>
<td>0.19</td>
<td>0.12</td>
<td>0.16</td>
<td>0.07</td>
<td>0.16</td>
<td>0.11</td>
<td>0.26</td>
</tr>
<tr>
<td>Solver Failed to Converge (%)</td>
<td>0.19</td>
<td>0.52</td>
<td>3.10</td>
<td>3.91</td>
<td>2.06</td>
<td>2.28</td>
<td>1.81</td>
<td>1.94</td>
<td>2.19</td>
<td>6.43</td>
</tr>
</tbody>
</table>


Several General Screening Rules

Suppose there are $R$ smooth screening “functions” $\sigma_i = (\sigma_{i,1}, \ldots, \sigma_{i,R})$, $\sigma_i : \mathcal{A} \times \mathbb{R} \to \mathbb{R}$, from which screening rules can be derived. A (multiple) conjunctive screening rule requires the simultaneous satisfaction of one or more rules indexed by $\mathcal{R} = (r_1, \ldots, r_{|\mathcal{R}|}) \subset \{1, \ldots, R\}$ and can be written as the vector inequality

$$s_i(x, p) = \begin{bmatrix} \sigma_{i,r_1}(x, p) \\ \vdots \\ \sigma_{i,r_{|\mathcal{R}|}}(x, p) \end{bmatrix} \leq 0;$$

equivalently, $s_i(x, p) = \max_{\mathcal{R} \in \mathcal{R}} \{ \sigma_i, \mathcal{R}(x, p) \} \leq 0$. Disjunctions, on the other hand, represent the satisfaction of any individual rule from some set $\mathcal{R} \subset \{1, \ldots, R\}$ and can be written

$$s_i(x, p) = \min_{r} \{ \sigma_{i,r}(x, p) \} \leq 0;$$

(see, e.g., [58]). Disjunctions of conjunctions are maximally general [30], and require the satisfaction of at least of C conjunctive rules each defined as the simultaneous satisfaction of the rules indexed by $\mathcal{R} \subset \{1, \ldots, R\}$; i.e.,

$$s_i(x, p) = \min_{\mathcal{R} \in \mathcal{R}} \left\{ \max_{r \in \mathcal{R}} \{ \sigma_{i,r}(x, p) \} \right\} \leq 0$$

(see [58]). Subset conjunctive rules, a special case of disjunctions of conjunctions, require the satisfaction of any fixed number $R'$ of a set of $R$ screening rules, where $R' \leq R$. While the exponential growth in the potential number of such rules threatens to make any model of non-compensatory decision processes intractable, there is evidence that people tend to use limited, simple rules in their decision-making [30, 59].

Consider vehicle choice as an example. Suppose $x = (x_1, x_2, x_3)$ where $x_1$ is a numeric index for brand, $x_2$ is a numeric index for body style, and $x_3$ is a numeric index for powertrain type (Gasoline ICE, Diesel, Hybrid, etc.). Let $I = 1$ (and suppress indexing with respect to $i$), $R = 3$, and suppose

$$\sigma_1(x, p) = -1 \text{ if } x_1 \text{ is "Toyota", and is } +1 \text{ otherwise;}$$

$$\sigma_2(x, p) = -1 \text{ if } x_2 \text{ is "sedan", and is } +1 \text{ otherwise;}$$

$$\sigma_3(x, p) = -1 \text{ if } x_3 \text{ is "Hybrid", and is } +1 \text{ otherwise.}$$

The following illustrate the categories of rules discussed above: The conjunctive rule “I’ll only consider hybrids” is $\sigma_3(x, p) \leq 0$. The negated conjunctive rule “I won’t consider hybrids” is $-\sigma_3(x, p) \leq 0$. The conjunctive rule “I’ll consider any Toyota hybrid” is

$$s(x, p) = \begin{bmatrix} \sigma_1(x, p) \\ \sigma_3(x, p) \end{bmatrix} \leq 0.$$

The disjunctive rule “I’ll consider any Toyota or any hybrid” is

$$s(x, p) = \min \{ \sigma_1(x, p), \sigma_3(x, p) \} \leq 0.$$

The subset conjunctive rule “I’ll consider a Toyota sedan, a Toyota hybrid, a hybrid sedan, or a Toyota hybrid sedan” is

$$s(x, p) = \min \begin{bmatrix} \max \{ \sigma_1(x, p), \sigma_2(x, p) \} \\ \max \{ \sigma_2(x, p), \sigma_3(x, p) \} \\ \max \{ \sigma_1(x, p), \sigma_3(x, p) \} \end{bmatrix} \leq 0.$$

The general disjunction-of-conjunctions rule “I’ll consider a Toyota hybrid or a sedan” is

$$s(x, p) = \min \begin{bmatrix} \max \{ \sigma_1(x, p), \sigma_3(x, p) \} \\ \sigma_2(x, p) \end{bmatrix} \leq 0.$$

B VEHICLE DESIGN AND CHOICE MODELS

Fuel consumption (in gpm) and 0-60 acceleration time (in s) are related by

$$g(a) = 0.035 + \frac{53.5 + 69.5e^{-a} - 1.8a^{1.4} + 106.9/a}{1000}$$
This formula is a modified version of the model estimated by [39], and plotted in Fig. 1. Our modifications are meant to ensure that acceleration performance and fuel economy are strongly inversely related: very fast acceleration times can be obtained with very low fuel economy, and very slow acceleration times are required for very high fuel economies. Unit costs are also a function of acceleration/fuel economy performance, given by

\[ c(a) = e^{\omega/12} \left( 1.5 + 1.97 e^{-a} - 0.04a + \frac{1}{a-1.5} \right). \] (25)

See Fig. 2. This formula is also modeled after the estimates in [39].

C PROOFS

Proof. Proof of Lemma 1: We prove (i), with (ii) and (iii) easy corollaries. Suppose that there are two distinct vectors \( \omega, \omega' \in \Omega \) that are both strictly choice consistent with \((X, p)\). Then for some \((i, j, r)\), \( \omega_{i,j,r} \neq \omega'_{i,j,r}. \) But by strict choice consistency, \( \omega_{i,j,r} = 1 = \omega'_{i,j,r} \) if \( s_{i,r}(x_j, p_j) \leq 0 \) and \( \omega_{i,j,r} = 0 = \omega'_{i,j,r} \) if \( s_{i,r}(x_j, p_j) > 0. \) This contradiction proves that \( \omega = \omega'. \) □

Proof. Proof of Lemma 2: Let \( (\omega, X, p) \) be feasible, and consider any \((i, j, r)\). Suppose \( s_{i,r}(x_j, p_j) > 0. \) Then because \( \omega_{i,j,r}s_{i,r}(x_j, p_j) \leq 0, \omega_{i,j,r} = 0. \) On the other hand, suppose \( s_{i,r}(x_j, p_j) < 0. \) Then because \((1 - \omega_{i,j,r})s_{i,r}(x_j, p_j) \geq 0, \omega_{i,j,r} = 1. \) If \( s_{i,r}(x_j, p_j) = 0) \) then \( \omega_{i,j,r} \) can be either 0 or 1. Because one of these conditions must hold for every \((i, j, r)\), \( \omega \) is almost choice-consistent. □

Proof. Proof of Lemma 3: By Lemma 1(ii), \( \pi^C(X, p) = \hat{\pi}(\hat{\omega}(X, p), X, p) \). Thus, because \( (\hat{\omega}(X, p), X, p) \) is feasible for Eqn. (15) for any feasible \((X, p)\), the optimal value \( \pi^{C,e} \) can be attained in Eqn. (15) by some feasible \((\hat{\omega}, X, p)\). Thus, the optimal value of Eqn. (15) must be at least \( \pi^{C,e} \).

For the second claim, assume that \((\omega, X, p)\) is a local solution of Eqn. (15) and \( \hat{\omega} \) is choice-consistent with \((X, p)\). By Lemma 1, \( \omega = \hat{\omega}(X, p) \) and \( \pi^R(X, p) = \hat{\pi}(\hat{\omega}, X, p) \). Local optimality means that \( \hat{\pi}(\omega, X, p) \geq \hat{\pi}(\omega', X', p') \) for all \((\omega', X', p') \) in a (feasible) neighborhood \( \mathcal{N} \) of \((\omega, X, p)\). Let \( D \) be the projection of \( \mathcal{N} \) onto the \( X \) and \( p \) coordinates. Because \( \hat{\pi}(X', p', X', p') \in \mathcal{D} \) for all \((X', p', X', p') \in \mathcal{D} \),

\[ \pi^C(X, p) = \pi^R(\omega, X, p) \geq \hat{\pi}(\hat{\omega}(X', p'), X', p') = \pi^C(X', p'). \]

Thus \((X, p)\) is locally optimal for Eqn. (9). □

Proof. Proof of Lemma 4: The proof given for Lemma 2 applies essentially unchanged. □

Proof. Proof of Lemma 5: The proof given for Lemma 3 applies essentially unchanged. □

D TRIVIAL KKT POINTS IN MPCC RELAXATIONS

An additional complication makes solution of Eqn. (15) or (16) potentially difficult: the KKT conditions have “trivial” solutions that are not necessarily profit-optimal. Standard NLP methods may be attracted to the spurious solutions represented by these points, rather than to actual solutions. We first prove that there are trivial KKT points in the example from Section 3, and then demonstrate the problem with numerical results.

The MPCC relaxation for Eqn. (6) is

\[
\begin{align*}
\text{maximize} & \quad \hat{\pi}(\omega, a, p) \\
\text{subject to} & \quad 0 \leq \omega \leq 1, \quad 2.5 \leq a \leq 15, \quad p \geq 0 \\
& \quad \omega(k(a, p) - B) \geq 0 \\
& \quad (1 - \omega)(k(a, p) - B) \geq 0
\end{align*}
\] (26)

The KKT conditions for Eqn. (26) are the MCP

\[
\begin{align*}
0 \leq \omega \leq 1 & \quad \Leftrightarrow (D^a \hat{\pi})(\omega, a, p) + (\lambda_1 + \lambda_2)(k(a, p) - B) \\
2.5 \leq a \leq 15 & \quad \Leftrightarrow (D^p \hat{\pi})(\omega, a, p) + (\omega(\lambda_1 + \lambda_2) - \lambda_2)(D^p k)(a, p) \geq 0 \\
0 \leq p & \quad \Leftrightarrow (D^\omega \hat{\pi})(\omega, a, p) - (\omega(\lambda_1 + \lambda_2))k(a, p) \geq 0 \\
0 \leq \lambda_1 & \quad \Leftrightarrow (1 - \omega)(k(a, p) - B) \geq 0 \\
0 \leq \lambda_2 & \quad \Leftrightarrow (1 - \omega)(k(a, p) - B) \geq 0
\end{align*}
\] (27)

This “MCP form” of the KKT conditions follows from the standard KKT conditions [35] and the definition of an MCP [47–49]; see Appendix E.

The existence of trivial solutions to these equations is a consequence of the following result:

Lemma 6. \((D^a \hat{\pi})(0, a, p) = 0, (D^p \hat{\pi})(0, a, p) = 0, \) and \((D^\omega \hat{\pi})(0, a, p) \geq 0 \) for all feasible \( a \) and \( p \geq c(a). \)

Proof. The specific claims are a consequence of the following general formulae:

\[
\begin{align*}
(D^a \hat{\pi})(w, a, p) & = \frac{e^{w(a, p)}}{1 + \text{W}(a, p)}(1 - F^C(w, a, p))(p - c(a)), \\
(D^p \hat{\pi})(w, a, p) & = (D^a a)(a, p)F^C(w, a, p)(1 - F^C(w, a, p))(p - c(a)) - (D^c)(a)F^C(w, a, p), \\
(D^\omega \hat{\pi})(w, a, p) & = (D^a a)(a, p)F^C(w, a, p)(1 - F^C(w, a, p))(p - c(a)) + F^C(w, a, p).
\end{align*}
\]

□

Corollary 1. For any feasible \( a \) and \( \lambda \geq 0 \), there exists some \( \bar{\pi}(a, \lambda) > 0 \) such that the KKT conditions for Eqn. (26), Eqn. (27), are solved by \((0, a, \lambda, 0)\) for any \( p \geq \bar{\pi}(a). \)
Proof. Setting $\omega = 0$, the KKT conditions reduce to $\lambda_1 \geq 0$ and

$$0 \leq -e^{a(a,p)}(p - c(a)) + (\lambda_1 + \lambda_2)(k(a, p) - B)$$

$$2.5 \leq a \leq 15, -\lambda_2(D^l k)(a, p)$$

$$0 \leq p - \lambda_2 R \geq 0$$

$$0 \leq \lambda_2 \leq k(a, p) - B \geq 0$$

The third equation requires $\lambda_2 = 0$, and the fifth simply requires $p \geq (B - mp g(a))/R$. Making these substitutions, we obtain

$$2.5 \leq a \leq 15, \ p \geq \max\{0, (B - mp g(a))/R\}, \ \lambda_1 \geq 0, \ \text{and}$$

$$-e^{a(a,p)}(p - c(a)) + \lambda_1(Rp + mp g(a) - B) \geq 0. \quad (28)$$

### E MCP FORM OF THE KKT CONDITIONS

This appendix proves the following result:

**Lemma 7.** Consider a generic minimization problem in positive-null form [35]:

$$\begin{align*}
\text{minimize} & \quad f(x) \\
\text{with respect to} & \quad 1 \leq x \leq u \\
\text{subject to} & \quad c^E(x) = 0, \ c^I(x) \geq 0
\end{align*}$$

The KKT conditions for this problem can be written in the “Mixed Complementarity Problem (MCP) form”

$$\begin{align*}
1 \leq x \leq u \perp (\nabla f(x) - (Dc^E)(x) \top \mu^E - (Dc^I)(x) \top \mu^I) \\
\infty < \mu^E < \infty \perp c^E(x) \\
0 \leq \mu^I \perp c^I(x) \geq 0
\end{align*}$$

See [45–49] for definitions and discussion related to MCPs.

**Proof.** The KKT conditions for this problem are as follows [35]: $x \in [l, u]$, $c^E(x) = 0$, $c^I(x) \geq 0$, and there exist multipliers $\mu^E \in \mathbb{R}^{M^F}$, $\mu^I \in \mathbb{R}^{M^I}$, $\mu^I \geq 0$, satisfying $\mu^I \perp c^I(x)$, and $\lambda^L, \lambda^U \in \mathbb{R}^{N}$, $\lambda^L \geq 0$ satisfying $\lambda^L \perp x - 1$ and $\lambda^U \perp u - x$ such that

$$\nabla f(x) - (Dc^E)(x) \top \mu^E - (Dc^I)(x) \top \mu^I - \lambda^L + \lambda^U = 0.$$