Signaling under Double-Crossing Preferences: The Case of Discrete Types*

CHIA-HUI CHEN
Kyoto University

JUNICHIRO ISHIDA
Osaka University

WING SUEN
University of Hong Kong

July 6, 2023

Abstract. The class of double-crossing preferences, where signaling is cheaper for higher types than for lower types at low signaling levels and the opposite is true at high signaling levels, underlines the phenomenon of countersignaling, with intermediate types choosing higher actions than higher and lower types. We provide an algorithm to systematically construct an equilibrium and thus establish equilibrium existence for this general class of preferences with an arbitrary discrete-type distribution. Our analysis sheds light on the connection between discrete-type and continuous-type models and clarifies robust predictions of signaling under double-crossing preferences.

Keywords. countersignaling; weak pairwise-matching condition; minimum allocation; low types separate and high types pairwise-pool

JEL Classification. D82; I21

*The first author acknowledges financial support from JSPS KAKENHI (20K01560). The second author acknowledges financial support from JSPS KAKENHI (18H03640, 19H01468, 20H05631, 21H04398), and the Joint Usage/Research Center for Behavioral Economics at ISER and Osaka University for its International Joint Research Promotion Program.
1. Introduction

Despite its widespread use in signaling models, the single-crossing property imposes strong restrictions on the structure of preferences, and its validity is not necessarily evident in economic applications. In a review article Hörner (2008) notes, “Little is known about equilibria when single-crossing fails, as may occur in applications.” More recent advances in the literature have partly redressed this gap in the literature. There are now several papers which deal with situations where the single-crossing property fails to hold, with applications to a variety of contexts including countersignaling (Feltovich et al., 2002; Araujo et al., 2007), signaling with grades or muddled information (Daley and Green, 2014; Frankel and Kartik, 2019), persuasion (Degan and Li, 2021), trade policy (Kolev and Prusa, 2002), timing of experimentation (Bobtcheff and Levy, 2017; Chen et al., 2021), and hypocrisy (Hallman and Spiro, 2023). Given this recent development, in Chen et al. (2022), we introduce a double-crossing property and provide a general analysis of signaling under this property. Loosely speaking, preferences exhibit the double-crossing property if signaling is cheaper for higher types than for lower types at low signaling levels but the opposite is true at high signaling levels. That paper identifies some fundamental features of double-crossing preferences, which enables us to obtain a full characterization of equilibria and establish equilibrium existence under such environments, when agent types are distributed continuously. From the practical point of view, however, discrete-type models are often more useful, or even indispensable, as many applied settings are naturally cast in discrete types. For this framework to be of use for applied work, it is important to discern which insights of the continuous-type model hold generally and which need amendment when agent types are discrete.

In this paper, we extend the continuous-type model of Chen et al. (2022) to an arbitrary number of types. The extension is not as straightforward as it may appear. Specifically, there are two technical issues that confront us. One issue is that in Chen et al. (2022), we first establish the continuity of the equilibrium action above some threshold and then exploit this feature to derive other equilibrium properties. This approach is not applicable to a discrete-type model that has no corresponding notion of continuity. The other issue is that with discrete types, agents may randomize over different actions, so we need to explicitly consider mixed strategies. This substantially enlarges the strategy space, as it is now a set of probability distributions over actions, whereas the equilibrium strategy can be represented simply by a (piecewise continuous) function of agent type in the continuous-type model. These issues prevent us from applying the results of Chen et al. (2022) directly to a
discrete-type model, both for obtaining a characterization and for establishing equilibrium existence, and force us to explore different avenues.

Central to our analysis are the general characterization and existence results that hold for any number of types. In Section 3, we provide a characterization of equilibria that satisfy the D1 refinement (hereafter, D1 equilibria): types above some threshold are pairwise matched, and the equilibrium action is quasi-concave in type. Our characterization result thus confirms that the key insights of Chen et al. (2022) hold in a qualitative sense, but also reveals that there are some properties that are specific to the discrete-type model. The most symbolic difference is captured by what we call the weak pairwise-matching condition, which plays an essential role in ensuring global incentive compatibility. In the continuous-type model, pairwise matching is characterized by a bijective mapping, where all types above the threshold pool with some other type. In the discrete-type model, the mapping is often an injection, and the lower end of types may be left unmatched when types are not dense enough. This result illuminates a fundamental property of signaling under double-crossing preferences: it is the higher end of types above the threshold who need to be pooled with lower types, but not vice versa. Understanding this property is crucial when we construct an equilibrium.

In Section 5, we provide an algorithm to construct a D1 equilibrium that works for any double-crossing preferences and any number of types, and show by construction that a D1 equilibrium always exists. In the continuous-type model, equilibrium is essentially characterized by a system of differential equations, and the problem of finding an equilibrium reduces to a standard initial value problem, to which we can apply standard arguments to establish the existence of solutions. No such techniques can be used when types are discrete. In the discrete-type model, we must consider the possibility of mixed strategies, and equilibrium is characterized by a set of probability distributions over actions. The strategy space thus expands substantially as the number of types increases. Moreover, since the equilibrium conditions (i.e., incentive compatibility and D1) in the discrete-type model generally appear as inequality constraints, there is a continuum of possible actions for each type to be considered, making it complicated to pin down an equilibrium.

A conceptual advancement on this front is what we call the minimum allocation. Our equilibrium construction centers around this notion. Given our characterization result on pairwise matching, we choose types in the lower end of the type distribution to match with the highest remaining type in each round of our algorithm, and move down until we run out of types to match. The problem is that in each round, we need to determine
which fractions of types to match with the highest remaining type and which fractions to remain fully separated, so as to satisfy all the equilibrium conditions. The adjustment process could be highly complicated and easily become intractable, as there are a multitude of possible choices in each round due to inequality constraints. We show that this process can be substantially simplified and guaranteed to always end up with a well-behaved equilibrium by searching for the minimum allocation repeatedly in each round.

In Section 6 we compare the discrete-type case to the continuous-type case by studying convergence properties. We show that for every equilibrium in the continuous-type model, there is a sequence of equilibria in the discrete-type model that converges to the continuous-type equilibrium in the limit. In that sense the analysis of the continuous-type model in Chen et al. (2022) is robust. However we also show that there exist other sequences of discrete-type equilibria that converge to the continuous-type limit, where for each each equilibrium along such sequences, the weak pairwise-matching condition holds by construction but pairwise matching as a bijective mapping does not hold. This result suggests that the weak pairwise-matching condition is a robust prediction of signaling models under the double-crossing property, while the bijective pairwise matching is not necessarily so.

One important economic insight we gain from the class of models with double-crossing preferences is countersignaling—a phenomenon where higher and lower types are pooled at a low signaling action while some intermediate types separate by choosing a higher action (Feltovich et al., 2002; Araujo et al., 2007). While real-life examples of countersignaling abound, analyzing this phenomenon in general terms has proved to be difficult and elusive, and our understanding of countersignaling has been limited to specific contexts despite its potential social implications: for instance, Feltovich et al. (2002)—the seminal work on countersignaling—consider only three types and restrict attention to pure-strategy equilibria. Our analysis provides a flexible framework to understand countersignaling under general payoff and distribution functions and can be used to analyze this perverse yet pervasive phenomenon in a variety of applied contexts.

---

1 Feltovich et al. (2002) raise a number of examples drawn from common observations, such as “The nouveau riche flaunt their wealth, but the old rich scorn such gauche displays,” and provide experimental evidence in support of countersignaling. Also see Araujo et al. (2007) and the references therein. Dixit and Nalebuff’s (2008) book, *The Art of Strategy*, has a section devoted to countersignaling.
2. Model

We consider a model with an arbitrary number $I \geq 2$ of types. Let the type space be given by $\{\theta_1, \ldots, \theta_I\}$, where $\theta_i < \theta_{i+1}$ for all $i = 1, \ldots, I - 1$. The type of an agent is his private information. Since we consider general payoff functions, the difference between $\theta_i$ and $i$ is superficial, and we can use them interchangeably. For most of our analysis, we thus identify each type directly with $i$, instead of $\theta_i$, to avoid clutter.\(^2\)

In the following analysis, it is often convenient to distinguish between agent $n$ and type $i$. We suppose that there is a continuum of agents with unit measure, each indexed by $n \in [0, 1]$. Agents are partitioned into types $i = 1, \ldots, I$ by $(F_0, \ldots, F_I)$, where $F_0 = 0$, $F_I = 1$, and $F_i < F_{i+1}$ for $i = 0, 1, \ldots, I - 1$. The partition $(F_0, \ldots, F_I)$ summarizes the type distribution with $F_i = \mathbb{P}(i \leq i')$. Let $\iota(n)$ indicate the type of agent $n$, where

$$\iota(n) = \begin{cases} i & \text{if } n \in [F_{i-1}, F_i) \text{ for } i < I, \\ I & \text{if } n \in [F_{I-1}, F_I]. \end{cases}$$

We adopt the above convention because it is useful for representing mixed strategies. Suppose type $i$ randomizes between actions $a''$ and $a'$ with probabilities $z$ and $1 - z$ respectively. This is equivalent to having agent $n'' \in [F_{i-1}, (1 - z)F_{i-1} + zF_i)$ choose $a''$ and agent $n' \in [(1 - z)F_{i-1} + zF_i, F_i)$ choose $a'$\.\(^3\) Under this convention, we need to pay special attention to agents in the set $B := \{F_0, \ldots, F_{I-1}\}$. We refer to $B$ as the set of “threshold agents.”

Let $u(a, t, i)$ denote the payoff to an agent of type $i$, where $a$ is the signaling action he chooses and $t$ is the market’s perception of his type, or his “reputation,” i.e., $t = \mathbb{E}[i \mid a] \in [1, I]$. Our specification of double-crossing preferences follows Chen et al. (2022).

**Assumption 1.** $u : \mathbb{R}_+ \times [1, I] \times \{1, \ldots, I\} \to \mathbb{R}$ is twice continuously differentiable, strictly increasing in $t$ and strictly decreasing in $a$.

**Assumption 2.** For any $i' > i''$, there exists a continuous function $D(\cdot; i', i'') : [1, I] \to \mathbb{R}_+$ such that:

\(^2\) We define the type space in this way only to be consistent with the continuous-type model of Chen et al. (2022). We will come back to this notation when we discuss convergence to the continuous-type model in Section 6.

\(^3\) In this example, if $i = I$, then our convention would have agent $n' = F_I$ also choose action $a'$. 

4
Assumption 3. For any $t$, $D(t; i', i'')$ strictly decreases in $i'$ and in $i''$.

Assumption 1 states that signaling is costly and that the agent benefits from a higher reputation. The “dividing line” $D(\cdot; i', i'')$ specified in Assumption 2 partitions the $(a, t)$-space into two regions for each pair of types. For actions to the left of the dividing line, the standard single-crossing property holds for types $i'$ and $i''$. To the right of the dividing line, the reverse single-crossing property holds: whenever the lower type $i''$ is indifferent between two allocations, the higher type $i'$ strictly prefers the one with the lower action. We place no restriction on the shape of $D(\cdot; i', i'')$, other than that it is a continuous function of $t$.

When there are only two types, Assumptions 1 and 2 provide enough structure to analyze signaling under double-crossing preferences because $D(\cdot; 2, 1)$ is the only dividing line. When there are more than two types, there are $I(I-1)/2$ dividing lines to be considered. The analysis would clearly become unmanageable without further restriction. Assumption 3 provides the restriction we adopt.

The restrictions imposed by Assumptions 1 to 3 can also be expressed in terms of the marginal rate of substitution between $a$ and $t$. Let

$$MRS(a, t, i) := -\frac{u_a(a, t, i)}{u_t(a, t, i)}$$

represent the marginal rate for type $i$ at $(a, t)$, and let $t = \phi(a, u_0, i)$ represent the indifference curve of type $i$ at utility level $u_0$. Since $u_a(\cdot) > 0$ and $u_t(\cdot) < 0$ under Assumption 1, the marginal rate of substitution is always positive. If type $i''$ attains utility level $u_0$ at $(a_0, t_0)$, then for $i' > i''$, the difference in their marginal rates along the indifference curve of type $i''$ is

$$MRS(a, \phi(a, u_0, i''), i') - MRS(a, \phi(a, u_0, i''), i'').$$

Chen et al. (2022) show that Assumption 2 is equivalent to the requirement that the above difference is strictly single-crossing from below in $a$, with crossing point at $a =$
\[ a_0 = D(t_0; i', i'') \]. Furthermore, if an indifference curve \( \phi(\cdot, u_0, i'') \) crosses the dividing line \( D(\cdot; i', i'') \) at \( (a_0, t_0) \), then \( a < D(\phi(a, u_0, i''); i', i'') \) for all \( a < a_0 \), i.e., an indifference curve can cross a dividing line only once in the \((a, t)\)-space.\(^4\)

Assumption 3 is related to how the marginal rate behaves with respect to type. Specifically, it requires that for any \((a, t)\),

\[
MRS(a, t, i) - MRS(a, t, i - 1)
\]
is strictly single-crossing from below in \( i \). This is different from the standard setting with single-crossing preferences, in which a higher type always has a lower marginal rate of substitution than a lower type at all levels of \( a \) and \( t \); in the current setup, some intermediate type may have the lowest marginal rate of substitution. As emphasized in Chen et al. (2022), this assumption captures an important class of economic environments in which the single-crossing property breaks down: gains from signaling are bounded and diminish as an agent invests more in signaling; moreover, higher types often reach this point of diminishing returns at lower signaling levels than do lower types.

The model described here is identical to that of Chen et al. (2022), except that the type space is now discrete. As noted in Chen et al. (2022), a leading example of double-crossing preferences is “signaling with grades,” where the market observes an exogenous source of information about the agent. Consider, for instance, an environment where the market can observe a test outcome in addition to a signaling action. For simplicity, we suppose that the test outcome is binary, either pass or fail, and the agent passes the test with probability \( \theta \). If the agent passes the test, he is promoted and earns some lump-sum payoff \( V \); if not, he is fired and his outside payoff is determined by his reputation. The agent’s utility is then given by

\[
u(a, t, \theta) = \theta V + (1 - \theta)t - \left( \frac{\gamma a}{\theta} + \frac{a^2}{2} \right),\]

where \( \gamma > 0 \) is a cost parameter. The marginal rate of substitution is obtained as

\[
MRS(a, t, \theta) = \frac{\gamma + a \theta}{\theta(1 - \theta)}.
\]

One can verify that this model satisfies all the assumptions.

\(^4\) Suppose an indifference curve \( \phi(\cdot, u_0, i'') \) of type \( i'' \) crosses \( D(\cdot; i', i'') \) at \( a_1 \) and \( a_2 \). Indifference curves of the higher type \( i' \) that cross \( \phi(\cdot; u_0, i'') \) must be steeper than the latter between \( a_1 \) and \( a_2 \), which implies that type \( i' \) strictly prefers the lower action \( a_1 \) to the higher action \( a_2 \). This would violate Assumption 2(a).
3. Equilibrium Characterization

Let \((a(n), t(n))\) denote the allocation of agent \(n\) where \(a(\cdot)\) and \(t(\cdot)\) are generally step functions. It is without loss of generality to assume that \(a(\cdot)\) is weakly monotone within each type interval. If \(a(\cdot)\) is constant on the type interval \([F_{i-1}, F_i)\), type \(i\) adopts a pure strategy; otherwise, type \(i\) randomizes over different actions. Let \(Q(a)\) be the set of agents who choose action \(a\) in equilibrium. We say that \(Q(a)\) is a pooling set if it contains more than one type. For the subsequent analysis, it is convenient to define for a given \(n \in [0, 1]\),

\[
\phi^*(\cdot; t(n)) := \phi(\cdot, u(a(n), t(n)), t(n)),
\]

which is the indifference curve of type \(i(n)\) that passes through his equilibrium allocation \((a(n), t(n))\); for brevity, we refer to it as the equilibrium indifference curve of type \(i(n)\). Let \(\beta(a)\) be the belief associated with action \(a\).

Definition 1. A signaling equilibrium is a pair of strategy \(a : [0, 1] \rightarrow \mathbb{R}_+\) and belief \(\beta : \mathbb{R}_+ \rightarrow \Delta \theta\) such that the following hold:

(a) given \(\beta(\cdot)\), \(a(n) \in \arg \max_a u(a, E_{\beta(a)}[\theta], i(n))\) for each \(n \in [0, 1]\); and

(b) \(\beta(\cdot)\) is consistent with \(a(\cdot)\) and Bayes’ rule whenever applicable.

Throughout the analysis, we focus on signaling equilibria that satisfy the D1 criterion (Cho and Kreps, 1987), and simply refer to them as equilibria.\(^5\) Chen et al. (2022) show that any equilibrium under double-crossing preferences with a continuum of types is characterized as Low types Separate and High types Pairwise-Pool (LSHPP). There are two important properties of an LSHPP equilibrium. First, the equilibrium action is quasi-concave in type. Second, there is a threshold type above which types are pooled in a pairwise manner, where two distinct types, or two disjoint intervals of types, choose the same action. While intuition suggests that some extended version of these properties would continue to hold for an arbitrary number of types, the results of Chen et al. (2022) cannot be applied directly to the present context because the proof in that paper exploits the continuity of \(a(\cdot)\) above the threshold type—a notion that has no counterpart in the discrete-type model.

\(^5\) Signaling equilibrium is equivalent to perfect Bayesian equilibrium if we introduce a fictitious player (“the market”) who chooses some action \(t\) after observing \(a\) to minimize the loss function \((t - \theta)^2\). The D1 criterion requires that for any off-equilibrium action \(a'\), the belief \(\beta(a')\) assigns zero probability to type \(i\) if the set of reputations that would induce type \(i\) to deviate to \(a'\) is strictly smaller than the set of reputations that would induce another type \(i'\) to deviate to \(a'\).
In the continuous-type model, incentive compatibility and D1 impose tight restrictions on the feasible range of equilibrium actions. Specifically, suppose that the type space is given by \([\theta, \overline{\theta}]\), where we use \(\theta\) as the generic notation of the agent type. Then, there are two threshold types \(\theta_0\) and \(\theta^*\) and a mapping \(p : (\theta^*, \overline{\theta}) \rightarrow [\theta_0, \theta^*]\) such that for each type \(\theta \in (\theta^*, \overline{\theta})\), there is a paired type \(p(\theta) \in [\theta_0, \theta^*]\) who chooses the same action. Due to the denseness of the type space, the mapping is always bijective, and hence \(Q(a(\theta))\) is a pooling set for all \(\theta \in [\theta_0, \theta^*]\). Moreover, the marginal rates of substitution must be exactly equalized at the end points of each pooling set (except when the pooling set includes the highest type): if there is any pooling at \((a_p, t_p)\), then types \(\min Q(a_p)\) and \(\max Q(a_p)\) must have the same marginal rate of substitution.

Equilibrium characterization is more involved in the discrete-type model because those nice properties do not hold exactly when there are no arbitrarily close “adjacent types.” We can still show, however, that a weaker version of pairwise matching holds.

**Definition 2.** The weak pairwise-matching condition holds if the following two conditions are satisfied:

(a) There exists \(n^* \in (0, 1)\) such that for any action \(a(n')\) chosen by agent \(n' > n^*\), there exists \(n'' < n^*\) such that agent \(n''\) chooses the same action \(a(n')\).

(b) If there is pooling at \((a_p, t_p)\), then \(Q(a_p) = [n_1, n_2) \cup [n_3, n_4)\) with \(n_2 \leq n^* \leq n_3\), where

\[
MRS(a_p, t_p, \iota(n_3)) \geq \lim_{n \uparrow n_2} MRS(a_p, t_p, \iota(n)),
\]

\[
MRS(a_p, t_p, \iota(n_4 - \epsilon)) \leq \begin{cases} 
MRS(a_p, t_p, \iota(n_1)) & \text{if } n_1 \notin B, \\
MRS(a_p, t_p, \max\{\iota(n_1) - 1, 1\}) & \text{if } n_1 \in B.
\end{cases}
\]

Part (a) of the above definition requires that every action chosen by a type higher than \(\iota(n^*)\) is also chosen by another type below \(\iota(n^*)\). In other words, every type above \(\iota(n^*)\) is pooled with some type(s) below \(\iota(n^*)\). This result stems from the fact that when \(a(\cdot)\) is not monotone, the corresponding reputation \(t(\cdot)\) must also decrease at the higher end of types, which precludes the possibility of full separation for types higher than \(\iota(n^*)\). It is interesting to note, however, that the flip side of this is not required by the weak pairwise-matching condition. In particular, suppose the lowest type that pools with type \(I\) is type \(i_0\). Then Definition 2(a) does not rule out the possibility that some type between \(i_0\) and \(\iota(n^*)\) may choose an action that separates from all other types. This is qualitatively
different from the continuous-type case where no type above a gap type \( \theta_0 \) chooses a fully separating action.

Since each agent type is defined by a left-closed, right-open interval, in Definition 2(b) \( Q(\cdot) \) is also defined as the union of left-closed, right-open intervals (except when it includes agent \( F_i = 1 \), in which case the set is defined by a closed interval). If \( Q(a_p) = [n_1, n_2) \cup [n_3, n_4) \) and \( n_2 \) is a “threshold agent,” then for a small \( \varepsilon > 0 \), type \( \iota(n_2 - \varepsilon) \) chooses \( a_p \) but type \( \iota(n_2) \) does not. This is why we take the left limit in Definition 2(b). Instead of requiring equality of marginal rates of substitution, Definition 2(b) leaves some degree of freedom for the marginal rates of substitution among types that are pooled. The rationale for the direction of inequalities stated in the definition will become apparent when we explain below the restrictions on off-equilibrium beliefs imposed by the D1 refinement.

**Proposition 1.** In any equilibrium, \( a(\cdot) \) is weakly quasi-concave and the weak pairwise-matching condition holds.

**Proof.** Quasi-concavity. Suppose to the contrary that \( \min\{a(n_1), a(n_3)\} > a(n_2) \) for some \( n_1 < n_2 < n_3 \), with \( \iota(n_1) < \iota(n_2) < \iota(n_3) \). There are three possibilities: (i) \( a(n_3) > a(n_1) > a(n_2) \); (ii) \( a(n_1) = a(n_3) > a(n_2) \); and (iii) \( a(n_1) > a(n_3) > a(n_2) \).

Case (i). Since agent \( n_1 \) chooses a higher action than agent \( n_2 \), type \( \iota(n_1) \) must have a lower marginal rate of substitution at \( (a(n_1), t(n_1)) \). By definition, this is equivalent to

\[
a(n_1) > D(t(n_1), \iota(n_1), \iota(n_1)).
\]

Moreover, since agent \( n_1 \) chooses a lower action than agent \( n_3 \), type \( \iota(n_1) \) must have a lower marginal rate of substitution then type \( \iota(n_3) \). Similarly as above, this leads to

\[
a(n_1) < D(t(n_1), \iota(n_3), \iota(n_1)).
\]

These two equations imply that \( D(t(n_1); \cdot, \iota(n_1)) \) is increasing, which is a contradiction.

Case (ii). Let \( (a_p, t_p) \) represent the pooling action and the associated reputation chosen by agents \( n_1 \) and \( n_3 \) in this case. Let \( \tilde{i} \) and \( \tilde{\iota} \) be respectively the lowest and the highest type that choose \( a_p \) in equilibrium. Note that \( MRS(a_p, t_p, i) - MRS(a_p, t_p, i-1) \) is single-crossing from below in \( i \) if and only if \( MRS(a_p, t_p, \iota(n)) \) is quasi-convex in \( n \). Hence either type \( \tilde{i} \) or type \( \tilde{\iota} \) has the highest marginal rate of substitution among all types pooling at action \( a_p \). Because types are discrete, actions slightly below \( a_p \) are off-path. By D1, a slight downward deviation from \( a_p \) is attributed to either type \( \tilde{i} \) or type \( \tilde{\iota} \). If \( a_p > D(t_p, \tilde{i}, \iota) \), then the off-equilibrium belief associated with downward deviation is \( \tilde{\iota} \). This cannot be an equilibrium,
because $\bar{t} > t_p$. So we must have $a_p \leq D(t_p; \hat{t}, \bar{t}, i) \leq D(t_p; \iota(n_3), i)$. But $a(n_2) < a_p$ implies $a_p \geq D(t_p; \iota(n_2), i)$. This means that $D(t_p; \iota(n_2), i) < a_p \leq D(t_p; \iota(n_3), i)$, which contradicts the monotonicity of $D(t_p; \cdot, i)$.

Case (iii). Because signaling is costly, $a(n_1) > a(n_3)$ implies $t(n_1) > t(n_3)$. Since $n_1 < n_3$, either $Q(a(n_1))$ or $Q(a(n_3))$ (or both) must be a pooling set. Furthermore, if we let $n' = \sup Q(a(n_1))$ and $n'' = \inf Q(a(n_3))$, we must have $n' > n''$. If $n' > n_2$, then for agents $n_1 < n_2 < n'$, we have $a(n') = a(n_1) > a(n_2)$. This reduces to case (ii) above. If $n' \leq n_2$, then for agents $n'' < n_2 < n_3$, we have $a(n'') = a(n_3) > a(n_2)$. Again this reduces to case (ii).

Weak pairwise-matching condition. Quasi-concavity of $a(\cdot)$ implies that any pooling set must consist of at most two intervals. This suggests that we can find $n^*$ such that: (i) $a(n^*)$ is the highest action chosen on the equilibrium path; (ii) for any pooling action $a_p$ chosen on the equilibrium path, $Q(a_p) = [n_1, n_2) \cup [n_3, n_4)$ with $n_2 \leq n^* \leq n_3$.

To show that every type above $\iota(n^*)$ pools with a type below $\iota(n^*)$, suppose this is not true. Specifically let $n'$ be the lowest agent in $[n^*, F_t]$ who chooses an action $a'$ which is not chosen by agents in $[F_0, n^*]$. The corresponding reputation satisfies $t(n') \geq n'$. By quasi-concavity of $a(\cdot)$, $Q(a(n'))$ does not contain any $n \geq n'$. Therefore, we have $\sup Q(a(n')) < n'$, and so $t(n^*) \leq t(n')$. Since $a(n^*) > a(n')$ by definition, every agent would strictly prefer $(a(n'), t(n'))$ to $(a(n^*), t(n'))$, a contradiction.

Let the reputation corresponding to pooling action $a_p$ be $t_p$. Recall that $MRS(a_p, t_p, \iota(n))$ is quasi-convex in $n$, and hence the type in the pooling set $Q(a_p)$ with the highest marginal rate of substitution is either $\iota(n_1)$ or $\iota(n_4 - \epsilon)$ (for some small $\epsilon > 0$). When agent $n_1$ is not a “threshold agent” (i.e., when $n_1 \notin B$), if $MRS(a_p, t_p, \iota(n_4 - \epsilon)) > MRS(a_p, t_p, \iota(n_1))$, then a downward deviation from $a_p$ to an off-equilibrium action $a' < a_p$ would be attributed to type $\iota(n_4 - \epsilon)$ under D1, leading to a profitable deviation. When $n_1$ is a “threshold agent,” the same conclusion holds unless type $\iota(n_1) - 1$ has even greater incentive to deviate to $a'$. Only if type $\iota(n_1) - 1$ is indifferent between his own allocation and $(a_p, t_p)$, and if $MRS(a_p, t_p, \iota(n_4 - \epsilon)) \leq MRS(a_p, t_p, \iota(n_1) - 1)$, would the off-equilibrium belief be assigned to type $\iota(n_1) - 1$ rather than type $\iota(n_4 - \epsilon)$, preventing such a deviation.

Finally, we show $MRS(a_p, t_p, \iota(n_3)) \geq \lim_{n \to n_3} MRS(a_p, t_p, \iota(n))$. Note that quasi-concavity of $a(\cdot)$ implies $a(n') > a_p$ for $n' \in [n_2, n_3)$. Costly signaling in turn implies that $t(n') > t_p$. On one hand, the higher type $\iota(n')$ prefers the higher action $a(n')$ to the lower action $a_p$ while the lower type $\iota(n_2 - \epsilon)$ has the opposite preference for an arbitrarily small $\epsilon > 0$.  

10
Thus the standard single-crossing property holds between these two types. On the other hand, the higher type \( t(n_3) \) prefers the lower action \( a_p \) to \( a(n') \) while the lower type \( t(n') \) has the opposite preference. Thus the reverse single-crossing property holds between \( n' \) and \( n_3 \). These two observations imply that

\[
MRS(a_p, t_p, t(n')) < \min \left\{ \lim_{n \nrightarrow n_2} MRS(a_p, t_p, t(n)), MRS(a_p, t_p, t(n_3)) \right\}
\]

Because \( MRS(a_p, t_p, t(\cdot)) \) is quasi-convex in \( n \), this in turn implies that the type with the lowest marginal rate of substitution in the pooling set \( Q(a_p) \) is either \( t(n_2 - \varepsilon) \) or \( t(n_3) \). If, contrary to the weak pairwise-matching condition, we have \( MRS(a_p, t_p, t(n_3)) < \lim_{n \nrightarrow n_2} MRS(a_p, t_p, t(n)) \), then an upward deviation from \( a_p \) would be attributed to type \( t(n_3) \) under D1. To prevent off-equilibrium deviations, it must be that \( t_p \geq n_3 \). But this would lead to a contradiction, because \( t(n') > t_p \geq n_3 \) while \( Q(a(n')) \) consists of types weakly lower than \( t(n_3) \).

**Proposition 2.** In any equilibrium, the following properties hold:

(a) If type \( i < t(n^*) \) randomizes over a set of allocations, only the lowest allocation in the set can be separating;

(b) Types \( i \geq t(n^*) \) do not randomize over different allocations.

**Proof.** Part (a). Suppose some type \( i' < t(n^*) \) randomizes over \( (a_1, t_1) \) and \( (a_2, t_2) \), with \( a_1 < a_2 \) and \( t_1 < t_2 \), where \( (a_2, t_2) \) is separating. Let \( Q(a_1) = [n_1, n_2) \cup [n_3, n_4) \). Recall that the type with the lowest marginal rate of substitution in the pooling set \( Q(a_1) \) is either \( t(n_2 - \varepsilon) \) or \( t(n_3) \) for an arbitrarily small \( \varepsilon > 0 \). Therefore, under D1, a deviation to \( a' \) slightly above \( a_1 \) (but below \( a_2 \)) would be given an off-equilibrium belief \( t' \geq t(n_2 - \varepsilon) \geq i' = t_2 \). Thus type \( i' \) would strictly prefer \( (a', t') \) to \( (a_2, t_2) \).

Part (b). Suppose first that there is some type \( i' > t(n^*) \) who randomizes over two different allocations \( (a_1, t_1) \) and \( (a_2, t_2) \), where \( a_1 < a_2 \) and \( t_1 < t_2 \). Let \( i'' \) be the largest type below \( t(n^*) \) who pools at \( (a_1, t_1) \). Then, by the weak pairwise-matching condition, we have \( MRS(a_1, t_1, i'') \leq MRS(a_1, t_1, i') \), which in turn implies that \( \phi^*(a; i'') < \phi^*(a; i') \) for all \( a > a_1 \). This is a contradiction because type \( i'' \) must strictly prefer \( (a_2, t_2) \) to \( (a_1, t_1) \). To show that type \( t(n^*) \) cannot randomize, suppose to the contrary that type \( t(n^*) \) randomizes over \( (a_1, t_1) \) and \( (a_2, t_2) \) where \( a_1 < a_2 \) and \( t_1 < t_2 \). Note that we only need to consider the case where \( (a_2, t_2) \) is separating, because if \( Q(a_2) \) is a pooling set, all type \( t(n^*) + 1 \) agents must choose \( (a_2, t_2) \), in which case we can just redefine \( n^* \) to be in \([F_{(n')}, F_{(n') + 1})\).
Figure 1. The signaling function is quasi-concave and satisfies the weak pairwise-matching condition.

This means that some of type $\iota(n^*)$ agents pool with type $\iota(n^*) + 1$ at $(a_1, t_1)$ while others are fully separated at $(a_2, t_2)$. We can then apply the argument in part (a) to derive a contradiction.

The equilibrium requirements described in Propositions 1 and 2 are illustrated in Figure 1. Agent $n_0$ in this figure is the lowest agent pooling with type $l = 15$. The signaling function $a(\cdot)$ below $n_0$ is determined by the least-cost separating solution. We pick an arbitrary agent choosing the highest action and label this agent $n^*$. Each type higher than $\iota(n^*)$ (i.e., types 11 to 15) adopt a pure strategy, and each type pools with some type below $\iota(n^*)$. Note that some types between $\iota(n_0)$ and $\iota(n^*)$ (specifically, types 6 and 7) may choose separating actions, and when type 7 randomizes between a separating action and a pooling action, the separating action is lower. The signaling function in Figure 1 is quasi-concave.

4. Equilibrium Construction with a Small Number of Types

In contrast to the extant literature on countersignaling, the general model in this paper allows an arbitrary number of discrete types. We will provide in Section 5 an algorithm
to construct an equilibrium for the general case. Because this algorithm is unavoidably complex in order to accommodate all possible double-crossing preferences and all type distributions, in this section we first illustrate the main ideas and difficulties of our equilibrium construction using very simple type distributions.

Consider the simplest case with only two types, in which case there is only one dividing line \( D(\cdot; 2, 1) \). Let \( s^*(i) \) denote the least-cost separating solution, where type 1 is indifferent between \( (s^*(1), 1) \) and \( (s^*(2), 2) \), with \( s^*(1) = 0 \). When \( s^*(2) \leq D(2; 2, 1) \), the equilibrium indifference curve of type 1 stays strictly to the left of the dividing line for \( a \leq s^*(2) \). Because the single-crossing property prevails in the relevant region, the least-cost separating solution is an equilibrium. Also, by standard argument, no other equilibrium can satisfy \( D_1 \) (Cho and Kreps, 1987).

When \( s^*(2) > D(2; 2, 1) \), no separating equilibrium exists because type 2 has a higher marginal rate of substitution at \( (s^*(2), 2) \), giving him an incentive to deviate to a lower action. In an equilibrium that involves some pooling, the pooling allocation must be on \( D(\cdot; 2, 1) \), for otherwise the higher type would have an incentive to deviate either to a slightly higher action or to a slightly lower action.

Let \( (\hat{a}, \hat{t}) \) be the intersection of the dividing line \( D(\cdot; 2, 1) \) and the indifference curve of type 1 passing through \((0, 1)\). If \( \hat{t} > E[i] \), the only possible equilibrium is a semi-pooling equilibrium in which type 1 randomizes between \((0, 1)\) and \((\hat{a}, \hat{t})\) and type 2 chooses \((\hat{a}, \hat{t})\). Observe that the equilibrium indifference curve of type 2 is flatter than that of type 1 to the left of the dividing line \( D(\cdot; 2, 1) \), and is steeper than that of type 1 to the right of the dividing line. Hence any deviation from that allocation which weakly benefits type 2 would strictly benefit type 1. Under \( D_1 \), such a deviation would be attributed to type 1, which makes the deviation unprofitable. If \( \hat{t} \leq E[i] \), both types strictly prefer to choose \((\hat{a}, \hat{t})\), and the only equilibrium is a fully pooling equilibrium at \( a_p = D(E[i]; 2, 1) \) and \( t_p = E[i] \).

**Proposition 3.** When there are two types, there exists a unique \( D_1 \) equilibrium. Moreover, if such an equilibrium is not fully separating, the pooling allocation must lie on the dividing line \( D(\cdot; 2, 1) \).

This proposition is established in the preceding discussion. The uniqueness of \( D_1 \) equilibrium in specific instances of double-crossing preferences has been pointed out in Daley and Green (2014) and Chen et al. (2021)

Next consider a model with three types. Although it is possible to enumerate all the
possible equilibrium configurations, such an exercise would be rather tedious. Instead we
will explain why equilibrium construction may be difficult even in this very simple case
and how our characterization results help narrowing down the set of possible equilibrium
configurations. Suppose we are interested in finding an equilibrium that exhibits coun-
tersignaling with two actions (i.e., the signaling function \( a(\cdot) \) is nonmonotone and takes
two values). Even in this particular class of equilibria, there are still many possible config-
urations: some type 1 or type 3 agents (or both) may pool with type 2 agents at the high
action; or some type 2 agents may pool with type 1 and type 3 agents at the low action.
Our characterization result suggests, however, that many of these configurations are inco-
sistent with the equilibrium conditions and can hence be ruled out at the outset. More pre-
cisely, Proposition 2(a) states that neither type 2 nor type 3 can randomize, meaning that
all type 2 agents must choose the high action while all type 3 agents must choose the low
action in any countersignaling equilibrium with three types and two actions. This leaves us
with only two possible forms of countersignaling equilibrium, depending on whether type
1 randomizes between the two actions or not. Moreover, Proposition 2(a) also implies that
the pooling action chosen by type 3 must be the second highest in any countersignaling
equilibrium with three types,\(^6\) which further narrows down the set of possible equilibrium
configurations.

When there are four or more types, the situation becomes much more complicated and
increasingly intractable, even with the help of our characterization. For instance, with
four types, even if we just consider pure-strategy countersignaling equilibria, there are
five possible patterns of countersignaling.\(^7\) Added to the complication is that Assumption
3 does not impose any restriction on the relative location of the dividing lines \( D(\cdot; 4, 1) \)
and \( D(\cdot; 3, 2) \). Clearly a more systematic approach to finding an equilibrium is called for
whenever the model has many types.

An interesting point to note is that although the analysis becomes more difficult when

---

\(^6\) If the action chosen by type 3 is the highest, we cannot have countersignaling because the equilibrium
action must be quasi-concave. Suppose there are two actions above the action chosen by type 3. Then, for
each of these actions, there must be a type adopting a pure strategy of choosing that action by Proposition
2(a). This implies that type 1 must adopt a pure strategy and choose a higher action than type 3, which
necessarily violates the weak pairwise-matching condition.

\(^7\) We can have types 1 and 4 pool while types 2 and 3 pool at a higher action; types 1 and 4 pool while
types 2 and 3 separate at two higher actions; types 1 and 2 pool with type 4 while type 3 separates; types
1 pools with types 3 and 4 while type 2 separates; or types 2 and 4 pool while type 1 separates at a lower
action and type 3 separates at a higher action.
there are more than three types, this is precisely where it becomes more intriguing under double-crossing preferences. To see this point, under single-crossing preferences, any D1 equilibrium must be the least-cost separating equilibrium, for which having two types is enough to capture its essence. This is not the case under double-crossing preferences. Evidently, to have countersignaling under this class of preferences, we must have at least three types as in Feltovich et al. (2002). With only three types, however, the highest type must pool with the lowest type at the lowest action and the middle type choose the highest action in any counter-signaling equilibrium by design. This leaves many interesting questions unanswered, such as what action the highest type would take and which type would take the highest action. To address these questions, therefore, we need at least four types, and in general we can provide a more precise view of signaling behavior as we have more types under double-crossing preferences.

5. Equilibrium Existence

Under single-crossing preferences, it is well known that the least-cost separating equilibrium satisfies the D1 refinement. Because of this, finding a D1 equilibrium in this environment is conceptually straightforward and equilibrium existence is never an issue, even when the number of types is very large: each separating action can be pinned down by the corresponding local incentive compatibility constraint alone, and an equilibrium can be constructed from bottom up. This is not the case under double-crossing preferences, where equilibrium may entail some pooling and the equilibrium conditions (i.e., incentive compatibility and D1) impose joint restrictions on payoff and type distribution functions. As discussed in the previous section, finding an equilibrium under double-crossing preferences by brute force can be extremely tedious, especially when there are many types. Moreover, there is no easy way to tell whether a D1 equilibrium exists for any given preferences and type distribution.

In this section, we provide a systematic way to find a D1 equilibrium under double-crossing preferences. The algorithm developed here works for any double-crossing preferences and type distribution, and can be used to establish the existence of a D1 equilibrium by construction.
5.1. An overview of the algorithm for equilibrium construction

Because the signaling function is quasi-concave, our construction finds the actions chosen by agents near the two ends of the agent distribution first. Then, in successive rounds, we find the actions chosen by agents nearer the middle part of the agent distribution until the highest equilibrium action is pinned down. One of the agents choosing that highest action corresponds to the equilibrium \( n^* \). Each round of this construction is designed to respect local incentive compatibility (each agent has no incentive to deviate to adjacent allocations) and the weak pairwise-matching condition. We show later that these two restrictions together are sufficient to guarantee global incentive compatibility.

We also make sure that Bayes’ rule for belief consistency is satisfied at any equilibrium pooling action. Suppose \( Q(a_p) = [n_1, n_2) \cup [n_3, n_4) \) for some \( a_p \). Bayes’ rule requires that for all \( n \in Q(a_p) \),

\[
t(n) = \frac{\int_{n_1}^{n_2} \iota(n) \, dn + \int_{n_3}^{n_4} \iota(n) \, dn}{(n_2 - n_1) + (n_4 - n_3)}.
\]

It is particularly convenient to define

\[
\mu(n'', n'; i) := \frac{\int_{n''}^{n'} \iota(n) \, dn + \int_{F_{i-1}}^{F_i} \iota(n) \, dn}{(n' - n'') + (F_i - F_{i-1})},
\]

for \( n'' \leq n' \leq F_{i-1} \). This formula gives the expected type when agents from \( n'' \) to \( n' \) pool with all type \( i \) agents. When agents in \([q, r)\) pool with all agents of type \( j \), Bayes’ rule can be written as \( t(n) = \mu(q, r; j) \) for \( n \in [q, r) \). The following properties of \( \mu(\cdot) \) will be useful: (a) \( \mu(n'', n'; i) \) is continuously increasing in \( n'' \) and equals \( i \) at \( n'' = n' \); and (b) \( \mu(n'', n'; i) \) decreases in \( n' \) when \( \mu(n'', n'; i) > \iota(n') \) and increases in \( n' \) when \( \mu(n'', n'; i) < \iota(n') \).

Our algorithm for equilibrium construction exploits a key property established in Propositions 1 and 2—namely, each type \( i \geq \iota(n^*) \) adopts a pure strategy and pools with some agents below \( n^* \). We therefore propose an algorithm with multiple rounds. In the first round, we find a set of agents, say \([q_1, r_1)\), who pool with type \( I \) at some candidate allocation \((a_1, t_1)\). (If \( q_1 > 0 \), we also determine the tentative allocations \((a(n), t(n))\) for \( n \in [0, q_1) \) based on that candidate allocation). In round 2, the allocations to agents in \([r_1, F_{i-1})\) remains to be determined. Among these remaining agents, we find a set of agents, \([q_2, r_2)\), who pool with type \( I - 1 \) at a weakly higher tentative allocation \((a_2, t_2)\) (along with the allocations chosen by \( n \in [r_1, q_2) \) if \( q_2 > r_1 \)). In round 3, the allocations...
to agents in \([r_2, F_{j-2}]\) remains to be determined, and so on. The algorithm finds an equilibrium when by the end of some round, the allocations to all agents in \([0, 1]\) have been determined. It is possible that we cannot find a tentative allocation at some round \(k\) before the algorithm stops with a full solution. In that case, the algorithm specifies how the tentative allocation at some round \(k' < k\) should be adjusted and the algorithm proceeds to round \(k' + 1\) again with the adjusted tentative allocation for round \(k'\) as the starting point.

The proof of equilibrium existence essentially boils down to showing that this algorithm will always end with a full solution for the allocations of all agents in \([0, 1]\).

To introduce the notation for this algorithm, for \(k \geq 1\), define \(j := I + 1 - k\) to be the highest type among all agents whose allocation remains to be determined in round \(k\). The tentative solution in round \(k\) is represented by a 4-tuple, \((a_k, t_k, q_k, r_k)\), with the interpretation that agents in \([q_k, r_k)\) pool with all type \(j\) agents at allocation \((a_k, t_k)\). Let \(s_k := (a_{k-1}, t_{k-1}, n_k)\) represent the state in round \(k\), which is inherited from the tentative solution in round \(k-1\): \((a_{k-1}, t_{k-1})\) is the allocation chosen in the previous round, and \(n_k = r_{k-1}\) is the lowest agent whose allocation is yet to be determined. At the beginning of the algorithm, we initialize the state to \(s_1 = (0, 1, 0)\) where the allocation \((0, 1)\) corresponds to the least-cost separating solution for type 1.

As we will detail below, the pair \((t_k, q_k)\) provides enough information for finding an equilibrium, while \(a_k\) and \(r_k\) can be uniquely pinned down from \((t_k, q_k)\). Instead of working with the 4-tuple \((a_k, t_k, q_k, r_k)\), we thus denote the tentative allocation in round \(k\) simply by \(c_k := (t_k, q_k) \in [t_{k-1}, I] \times [r_{k-1}, F_{j-1}]\). Let \((c_1, \ldots, c_{k'})\) be the allocation path up to round \(k'\).

Among all allocations \(c_k\) in round \(k\) that will satisfy incentive compatibility and the weak pairwise-matching condition, the algorithm only considers a special subset of these allocations, which we call the candidate set and denote by \(C_k\). The candidate set depends on the state \(s_k = (a_{k-1}, t_{k-1}, n_k)\), but since \(a_{k-1}\) and \(n_k = r_{k-1}\) can be derived from \(c_{k-1}\) we sometimes write \(C_k(c_{k-1})\) to emphasize this dependence. When the candidate set is well defined, it can be a singleton or it can contain more than one element. If \(C_k\) contains more than one element, we argue below that there is a linear order \(\succeq\) defined on \(C_k\) such that any two elements in this set can be compared, despite the fact that a typical element of \(C_k\) is two dimensional. We use \(\bar{c}_k\) and \(c_k\) to denote the highest and lowest element of \(C_k\).

We are now in a position to provide a high-level description of our main algorithm for equilibrium construction:
Main algorithm. Each round $k \geq 2$ inherits the allocation path $(c_1, \ldots, c_{k-1})$ from earlier iterations of the algorithm. Given $c_{k-1} = (t_{k-1}, q_{k-1})$, apply Bayes’ rule to find the $r_{k-1}$ which is the smallest solution to $\mu(q_{k-1}, r_{k-1}; j + 1) = t_{k-1}$.

1. If $r_{k-1} = F_j$:  
   - stop with a well-defined equilibrium characterized by the existing allocation path $(c_1, \ldots, c_{k-1})$
2. If the equation $\mu(q_{k-1}, r_{k-1}; j + 1) = t_{k-1}$ does not admit a solution for $r_{k-1}$ on $[q_{k-1}, F_j]$:  
   - adopt the adjustment procedure to modify the allocation path to $(c_1', \ldots, c_{k-1}')$
   - return to the beginning of round $k$
3. Otherwise:  
   - update the state to $s_k = (a_{k-1}, t_{k-1}, n_k)$ with $n_k = r_{k-1}$
   - compute the candidate set $C_k(c_{k-1})$
   - if the minimum allocation $c_k^{\text{min}}$ exists in $C_k(c_{k-1})$, pick $c_k = c_k^{\text{min}}$; otherwise pick $c_k = \bar{c}_k \in C_k(c_{k-1})$
   - append $c_k$ to the allocation path $(c_1, \ldots, c_{k-1})$ and go to the beginning of round $k + 1$

The remainder of this section will specify precisely how the candidate set $C_k$ is determined (Section 5.2), define the minimum allocation $c_k^{\text{min}}$ as well as the highest allocation $\bar{c}_k$ (Section 5.3), and elaborate on the adjustment procedure that produces a modified allocation path from the original path (Section 5.4). Finally, we will show that the main algorithm will always stop at some finite round $k$ and produce a well-defined allocation for all agents. Since the equilibrium construction requires many steps and there are a number of different cases to consider, we supplement the description in the text by providing pseudocodes for the algorithm in an Online Appendix to clarify and summarize the program’s flow.

The allocation produced by our algorithm satisfies Bayes’ rule, local incentive compatibility, weak pairwise-matching, and is quasi-concave by construction. Equilibrium existence can be guaranteed by showing that local incentive compatibility also implies global incentive compatibility. Section 5.5 provides such an argument.
5.2. The candidate set

Since agents in \([q_k, r_k]\) pool with type \(j\) to produce reputation \(t_k\), we define the following mapping from \(c_k = (t_k, q_k)\) to \(r_k\) to ensure that Bayes’ rule is satisfied:

\[
\rho_k(c_k) := \min \left\{ r \in [q_k, F_{j-1}] : \mu(q_k, r; j) = t_k \right\}.
\]

Because \(\mu(q_k, r; j)\) is decreasing then increasing, there can be multiple values of \(r\) that satisfy the specified condition, and in that case we pick the smallest solution. It is possible that the specified condition admits no solution, in which case \(\rho_k(c_k)\) is undefined.

The candidate set \(C_k\) depends on the state \(s_k = (a_{k-1}, t_{k-1}, n_k)\), where \(n_k = \rho_{k-1}(c_{k-1})\). This set is undefined whenever \(\rho_{k-1}(c_{k-1})\) is undefined. In this subsection, we assume that \(C_k\) is well defined. The determination of the candidate set depends on whether \(n_k\) is a “threshold agent” (i.e., \(n_k \in B\)) or not (i.e., \(n_k \notin B\)). We deal with these two cases in turn.

Suppose first that \(n_k \notin B\). This means that type \(\iota(n_k)\) must be randomizing between \(a_{k-1}\) (determined in previous iterations of the algorithm) and a higher action \(a_k\) (to be determined). But part (b) of Proposition 2 requires that type \(\iota(n_k)\) agents with \(n > n_k\) cannot take a fully separating action. This implies that we must choose \(q_k = n_k\) for \(c_k\).

Since type \(\iota(n_k)\) agents randomize, any tentative allocation \((a_k, t_k)\) must be on the indifference curve of type \(\iota(n_k)\) that passes through \((a_{k-1}, t_{k-1})\). Also, since they pool with type \(j\) agents, we require the tentative allocation to be on \(D(\cdot; j, \iota(n_k))\) so that it satisfies the weak pairwise-matching condition. Define \((\hat{a}(j, i; \bar{a}_k, \bar{t}_k), \hat{\iota}(j, i; \bar{a}_k, \bar{t}_k))\), or \((\hat{a}_k(j, i), \hat{\iota}_k(j, i))\) for short, to be the intersection of the relevant indifference curve and dividing line when the “reservation allocation” is \((\bar{a}_k, \bar{t}_k)\):

\[
u(a_k, t_k, i) = u(\bar{a}_k, \bar{t}_k, i),
\]

\[
a_k = D(t_k; j, i).
\]

If \(q_k = n_k\), the reservation allocation is simply the allocation chosen in the previous round, i.e., \((\bar{a}_k, \bar{t}_k) = (a_{k-1}, t_{k-1})\). Then, we let

\[
c_k = \begin{cases} (\hat{\iota}_k(j, \iota(n_k)), n_k) & \text{if } \hat{\iota}_k(j, \iota(n_k)) > t_{k-1}, \\ (t_{k-1}, n_k) & \text{if } \hat{\iota}_k(j, \iota(n_k)) \leq t_{k-1}. \end{cases}
\]

The candidate set \(C_k\) is a singleton containing \(c_k\) when \(n_k \notin B\).

---

\(\hat{a}_k, \hat{t}_k\) are determined by \(c_{k-1}\) and \(q_k\).
Note that the action $a_k$ can be determined once $c_k$ is pinned down, and we denote this relationship by $a_k = \alpha_k(c_k)$, where

$$
\alpha_k(c_k) = \begin{cases} 
\hat{a}_k(j, \mu(n_k)) & \text{if } \hat{t}_k(j, \mu(n_k)) > s_{k-1}, \\
 a_{k-1} & \text{if } \hat{t}_k(j, \mu(n_k)) \leq s_{k-1}.
\end{cases}
$$

The case where $\alpha_k(c_k) = a_{k-1}$ corresponds to “mass pooling” of the continuous-type case of Chen et al. (2022), where more than two types pool at the same allocation.

Next suppose that $n_k \in B$. This means that type $\mu(n_k) - 1$ chooses action $a_{k-1}$, while types $\mu(n_k)$ and above may potentially choose separating actions. In this case, we first construct a least-cost separating solution given the initial condition.\(^9\) We denote this solution by $s_k^*(i)$. There are two possibilities.

Case (i). If $s_k^*(j) \leq D(j; j, j-1)$, then there is a separating equilibrium for types between $\mu(n_k)$ and $j$ because the standard single-crossing property holds in the relevant region. In this case, the separating solution for these types, together with the allocation path derived from previous iterations of the algorithm, will constitute an equilibrium. Formally, we specify $c_k = (j, F_{j-1})$ to be the only element of our candidate set $C_k$, together with $\alpha_k(c_k) = s_k^*(j)$. With $q_k = F_{j-1}$, the solution to $\mu(q_k, r_k; j) = j$ is $r_k = F_{j-1}$, and so the main algorithm will stop with a well-defined equilibrium.

Case (ii). Otherwise, we find $i_k^* := \max\{i : s_k^*(i) < D(i; i, i)\}$; this is the largest type that can choose a separating action.\(^{10}\) If $s_k^*(\mu(n_k)) \geq D(\mu(n_k); j, \mu(n_k))$ and $i_k^*$ is not well defined, we adopt the convention to denote $i_k^* = \mu(n_k) - 1$. Observe that when $i_k^* = \mu(n_k) - 1$, type $\mu(n_k)$ agents cannot take a separating action, and the only feasible choice of $q_k$ is $n_k$. If $i_k^* \geq \mu(n_k)$, we may pick any $q_k \in [n_k, F_{i_k^*}]$. In either case, once we pick $q_k$, we can identify the feasible range of $t_k$. If $q_k \notin B$, the relevant indifference curve of type $\mu(q_k)$ is the one that passes through his reservation allocation $(s_k^*(\mu(q_k)), \mu(q_k))$, and the relevant dividing line is $D(\cdot; j, \mu(q_k))$. We can uniquely pin down the intersection $(\hat{a}_k(j, \mu(q_k)), \hat{t}_k(j, \mu(q_k)))$ and set $t_k = \hat{t}_k(j, \mu(q_k))$. If $q_k \in B$ and $\mu(q_k) > 1$, any candidate allocation must be bounded between the indifference curves of types $\mu(q_k) - 1$ and $\mu(q_k)$ that pass through the reservation

---

\(^9\) In round 1, the initial condition requires $s_1^*(1) = 0$; in all subsequent rounds, type $\mu(n_k) - 1$ must be indifferent between $(s_k^*(\mu(n_k)), \mu(n_k))$ and $(a_{k-1}, s_{k-1})$.

\(^{10}\) To see this, note that if type $i_k^* + 1$ chooses a separating action, it must be to the right of $D(i_k^* + 1; j, i_k^* + 1)$ by definition. As type $i_k^* + 1$ has a lower marginal rate of substitution than type $j$, it is not possible to find a pooling allocation that satisfies the weak pairwise-matching condition.
allegation given by

\[
(a_k, \bar{t}_k) = \begin{cases} 
(s_k^\ast(t(q_k) - 1), t(q_k) - 1) & \text{if } q_k > n_k, \\
(a_{k-1}, t_{k-1}) & \text{if } q_k = n_k.
\end{cases}
\]  

(1)

Moreover, the weak pairwise-matching condition requires that, for \(q_k \in B\) such that \(t(q_k) > 1\), any candidate allocation \((a_k, t_k)\) must satisfy:

\[
D(t_k; j, t(q_k) - 1) \geq a_k \geq D(t_k; j, t(q_k)).
\]

Therefore, the feasible range of \(t_k\) is \([\max\{\hat{t}_k(j, t(q_k)), t_{k-1}, \hat{t}_k(j, t(q_k) - 1)\}\]. Finally, if \(q_k \in B\) and \(t(q_k) = 1\) (i.e., \(q_k = 0\)), we have a special case because there are no lower types to separate from and hence no payoff upper bound. The feasible range of \(t_k\) in this case is unbounded and is given by \([\max\{\hat{t}_k(j, 1), 1\}, \infty)\). This fact will play a crucial role in our algorithm.

To sum up, in case (ii), the candidate set \(C_k\) is the set of \((t_k, q_k)\) such that

\[
q_k \in [n_k, F_k] \quad \text{and} \quad t_k \begin{cases} 
= \hat{t}_k(j, t(q_k)) & \text{if } q_k \notin B, \\
\in \left[\max\{\hat{t}_k(j, t(q_k)), \bar{t}_k\}, \hat{t}_k(j, t(q_k) - 1)\right] & \text{if } q_k \in B,
\end{cases}
\]

where \(\hat{t}_k(j, 0) = \infty\).\(^{11}\) Even though the candidate set is more complicated in this case because \(C_k\) is not a singleton, three properties of this set are important. First, when there is a range of feasible choices for \(q_k\), \(t_k\) is uniquely pinned down and fixed (type \(t(q_k)\) is choosing the same \((a_k, t_k)\) with different probabilities); and when there is a range of feasible choices for \(t_k\), \(q_k\) is uniquely pinned down and fixed (type \(t(q_k)\) is choosing different allocations \((a_k, t_k)\) that are all consistent with incentive compatibility and the weak pairwise-matching condition). Second, \(C_k\) is connected in that if \((t'_k, q_k), (t''_k, q'_k) \in C_k\) with \(t' > t''\), then \((t_k, q_k) \in C_k\) for any \(q_k\) and \(t_k \in (t''_k, t'_k)\); and if \((t_k, q''_k), (t'_k, q'_k) \in C_k\) with \(q'_k > q''_k\), then \((t_k, q_k) \in C_k\) for any \(t_k\) and \(q_k \in (q''_k, q'_k)\). Third, if both \((t'_k, q'_k)\) and \((t''_k, q''_k)\) belong to \(C_k\), then \(t''_k \geq t'_k\) if and only if \(q''_k \leq q'_k\). These properties of \(C_k\) are illustrated in Figure 2. They are important because they allow us to order allocations in the candidate set continuously, as we will discuss below.

\(^{11}\) For any \(c_k \in C_k\), we can also determine the corresponding value of \(a_k\), denoted \(a_k(c_k)\). However, since the precise definition of \(a_k(c_k)\) is cumbersome in case (ii), and is not germane to the discussion of the main algorithm, we leave the details to the Appendix A.
Figure 2. The candidate set $C_k$ contains pairs $(t_k, q_k)$ that lie on the blue line. The arrows indicate the linear order on this set: the point at the southeast end of $C_k$ is $c_k$ and the point at the northwest end is $c_k'$.

5.3. The minimum allocation

If the candidate set $C_k$ is not a singleton, there are more than one way to pick a tentative allocation in round $k$ that would satisfy incentive compatibility and the weak pairwise-matching condition. When this allocation is carried to the next round, and when $C_{k+1}$ is also not a singleton, the multiplicity of tentative allocation paths expands, making the search for an equilibrium difficult. Our main algorithm takes advantage of the fact that elements of the candidate set can be linearly ordered, even though they are two-dimensional. In this subsection we formally define this order relation and introduce a way to systematically pick a particular tentative allocation from a candidate set.

Recall that $\mu(q_k, r_k; j)$ is the resulting reputation when agents from $q_k$ to $r_k$ pool with all type $j$ agents; it is hence decreasing in $r_k$ when it is larger than $\iota(r_k)$ and is increasing in $r_k$ when it is smaller than $\iota(r_k)$. Let $\mu^*_k(q_k) := \min_{r \in [q_k, F_{k+1}]} \mu(q_k, r; j)$ represent the lowest possible reputation when type $j$ pools with an interval of agents starting from $q_k$. Because $\mu(\cdot, r; j)$ is continuous and strictly increasing, so is $\mu^*_k(\cdot)$. Define

$$\tau_k(c_k) := t_k - \mu^*_k(q_k).$$

This function is continuous and strictly increasing in $t_k$ and strictly decreasing in $q_k$. We say that $c_k$ is “higher than” $c'_k$, and write $c_k \succeq c'_k$, if and only if $\tau_k(c_k) \geq \tau_k(c'_k)$ (and we
Lemma 1. an allocation path in which the minimum allocation is chosen in the latest round. The connected, and unique of $C$ allocations in $r$ increasing, because

This means $\mu(q_k, r_k; j) = t_k$ for the first time as we gradually increase $c_k$ (according to the order $\succeq$) in $C_k$. In other words, $\pi_k(c_k)$ is undefined whenever $c_k^\text{min} \succ c_k$. It is clear that there exists a unique $c_k^\text{min}$ in $C_k$ if and only if $\tau_k(c_k) \geq 0 \geq \tau_k(c_k)$. This follows from the fact that $C_k$ is connected, and $\tau_k(\cdot)$ is continuous and strictly increasing in $c_k$.

A minimum allocation path is an allocation path $(c_1, \ldots, c_k)$ in which $c_k' = c_k^\text{min}$, i.e., an allocation path in which the minimum allocation is chosen in the latest round. The following statement establishes some convenient properties of a minimum allocation path.

Lemma 1. (a) $\rho_k(c_k^\text{min}) \in B$. (b) If $r_k = \rho_k(c_k^\text{min})$, then $t_k^\text{min} \leq \tau(r_k)$. (c) If $c_k = c_k^\text{min}$, then $0 > \tau_{k+1}(c_{k+1})$. (d) $c_1^\text{min} \in C_1$.

Proof. (a) Because $\mu(q_k^\text{min}, r; j)$ is decreasing in $r$ when it is larger than $\tau(r)$ and it reaches a minimum at $r = \rho(c_k^\text{min})$, we must have $\tau(r) < t_k^\text{min}$ for $r \in [q_k^\text{min}, \rho(c_k^\text{min}))$. Similarly, $\mu(q_k^\text{min}, r; j)$ is increasing in $r$ when it is smaller than $\tau(r)$; therefore, $\tau(r) \geq t_k^\text{min}$ for $r \in (\rho_k(c_k^\text{min}), F_{j-1}]$. This shows that $\tau(\cdot)$ must jump up at $r = \rho_k(c_k^\text{min})$, and so $\rho_k(c_k^\text{min})$ must be a “threshold agent.” Also, if $\mu(q_k^\text{min}, \cdot; j)$ is monotonically decreasing, then $\rho_k(c_k^\text{min}) = F_{j-1} \in B$.

(b) If $r_k = \rho_k(c_k^\text{min}) \in B$, then $\mu(q_k^\text{min}, \cdot; j)$ must be weakly increasing on $(\rho_k(c_k^\text{min}), F_{\tau(r_k)})$. This means $t_k^\text{min} \leq \tau(r_k)$. This property also holds when $\mu(q_k^\text{min}, \cdot; j)$ is monotonically decreasing, because $r_k = F_{j-1}$ and so $t_k^\text{min} < j = \tau(r_k)$.

(c) Since $n_k+1 = r_k$ and $c_k^\text{min}$ is a minimum allocation, by parts (a) and (b), we have $n_k+1 \in B$ and $t_k \leq \tau(n_k+1)$. Moreover, because $n_k+1 \in B$, the lowest allocation in the candidate set $C_{k+1}(c_k^\text{min})$ satisfies:

$$
\tau_{k+1}(c_{k+1}) = \begin{cases} 
\hat{t}_k (j-1, \hat{i}_{k+1}^*, 1) - \mu_{k+1}^* (F_{i_{k+1}^*}) & \text{if } \hat{i}_{k+1}^* \geq \tau(n_k+1), \\
t_k - \mu_{k+1}^* (F_{\tau(n_k+1)-1}) & \text{if } \hat{i}_k^* = \tau(n_k+1) - 1.
\end{cases}
$$
Since $\tilde{t}_k(j-1, i_{k+1}^* + 1) \leq i_{k+1}^* + 1 < \mu_{k+1}(F_{\tilde{t}_k}^{i_{k+1}^*})$, we have $\tau_{k+1}(c_{k+1}) < 0$ if $i_{k+1}^* \geq \mu(n_{k+1})$. If $i_{k+1}^* = \mu(n_{k+1}) - 1$, $t_k \leq \mu(n_{k+1}) < \mu_{k+1}(F_{\mu(n_{k+1})-1})$ implies $\tau_{k+1}(c_{k+1}) < 0$.

(d) The candidate set $C_1$ is special because once $q_1$ reaches 0, we do not require agent
$n = 0$ to be indifferent between $(a_1, t_1)$ and the reference allocation given by the state
$(a_0, t_0) = (0, 1)$ (type 1 may strictly prefer the former allocation to the latter). In other
words, $C_1$ is unbounded because the initial state only specifies a payoff lower bound for
type $\ell(0) = 1$. Once $q_1$ reaches 0, therefore, the reputation $t_1$ can be raised without bounds
along $D(\cdot; I, 1)$, and we will eventually have $\tau_1(t_1, 0) > 0$ for a sufficiently large $t_1$.

According to our main algorithm, an allocation path $(c_1, \ldots, c_k)$ constitutes an equi-
librium if $\rho_k(c_{k'}) = F_{j'-1}$ for some $k'$ and $j' = I + 1 - k'$. Parts (a) and (b) of Lemma 1
suggest that if we can find a minimum allocation in a round with two or three remaining
types (i.e., $n_k \in [F_{j-3}, F_{j-1}]$), the resulting allocation path necessarily constitutes an equi-
librium. To see this, consider some round $k'$ such that $n_{k'} \in [F_{j'-3}, F_{j'-1})$ and we find a
minimum allocation $c_{k'}^{\text{min}}$. We then have either (i) $\rho_k(c_{k'}^{\text{min}}) = F_{j'-1}$ or (ii) $\rho_k(c_{k'}^{\text{min}}) = F_{j'-2}$
and $t_{k'}^{\text{min}} \leq \mu(F_{j'-2}) = j' - 1$. In case (i), the allocation path $(c_1, \ldots, c_{k'}^{\text{min}})$ is an equilibrium.
In case (ii), we let all type $j' - 1$ agents separate at the top if $t_{k'}^{\text{min}} < j' - 1$, and pool at
the same allocation $c_{k'}^{\text{min}}$ if $t_{k'}^{\text{min}} = j' - 1$. Since there are only a finite number of types,
this means that we can find an equilibrium, as long as we can consistently find a minimum
allocation in the current round by adjusting the allocation path.

Parts (c) and (d) of Lemma 1 imply that we can always start with the minimum allo-
cation in round 1 of the main algorithm. Moreover, in the next round, we have $n_2 \in B$, which allows us to obtain $c_2$ and $\overline{c}_2$. Since $\tau_2(c_2) < 0$ by Lemma 1(c), we just need to check $\tau_2(\overline{c}_2)$. If $\tau_2(\overline{c}_2) \geq 0$, the minimum allocation $c_2^{\text{min}}$ exists in $C_2(e_1^{\text{min}})$. We can then
continue with the minimum allocation path and proceed to round 3 with $n_3 \in B$. If we can keep on finding the minimum allocation in every round, the algorithm will eventually find
an equilibrium at some point. If we fail to find the minimum allocation in some round,
the candidate set for the next round is undefined, and we will have to go back to previous
rounds and adopt an adjustment procedure to modify the allocation path.

\footnote{If $t_{k'}^{\text{min}} = j' - 1$, then $\mu(t_{k'}^{\text{min}}; r; j') = t_{k'}^{\text{min}}$ for $r \in [F_{j'-2}, F_{j'-1}]$. Therefore, we can extend pooling all the way to agent $F_j$.}
5.4. The adjustment procedure

Starting with the minimum allocation \( c_1^{\min} \), the main algorithm keeps appending the minimum allocation in the next round to the allocation path. If at some round \( k' \) the algorithm fails to find a minimum allocation, Lemma 1(c) implies that this occurs when \( \tau_{k'}(\bar{c}_k) < 0 \). Because even the highest allocation is “too low” to satisfy the belief consistency requirement, our adjustment procedure will try to pick a “higher” allocation (according to the order \( \geq \)) than the existing one. However, since \( \bar{c}_{k'} \) is already the highest allocation in \( C_{k'} \), we need to go back to previous rounds to make the adjustment. The basic rule of thumb is that we always increase the allocation in the latest round before \( k' \) that has still room to be adjusted upward. Define

\[
\hat{k}(k') := \max\{k \leq k' : \bar{c}_k > c_k\},
\]

as the “round of adjustment.” We often simply write \( \hat{k} \) for short, and let \( \hat{j} := I + 1 - \hat{k} \).

Observe that \( \hat{k} \) is always well defined because \( C_1 \) is not a singleton and is not bounded.

For any round \( k < k' \) with \( c_k = \bar{c}_k \), there is no room to be adjusted upward; note that this includes all rounds with \( n_k \notin B \), because \( C_k \) is a singleton and \( c_k = \bar{c}_k \). Therefore, if \( c_{k+1} = \bar{c}_{k+1} \), we can uniquely pin down \( c_{k+1} \) from \( c_k \). By repeatedly applying this process, we can uniquely pin down \( c_{k+m} \) from \( c_k \) if \( c_k = \bar{c}_k \) for all \( k = \hat{k} + 1, \ldots, \hat{k} + m \). Let \( c_{k+m} = \zeta^m_k(c_k) \) denote this mapping.

**Lemma 2.** Suppose \( c_k = \bar{c}_k \) for \( k = \hat{k} + 1, \ldots, \hat{k} + m \), such that \( \zeta^m_k(\cdot) \) is well defined. Then, \( \zeta^m_k(\cdot) \) is strictly increasing.

**Proof.** When \( c_k = \bar{c}_k \), \( t_k = \hat{t}_k(j, \iota(n_k)) \) and \( q_k = n_k \). Consider a slight increase in \( c_k \) according to the order \( \succeq \). There are two ways to do this, either by increasing \( t_k \) or by decreasing \( q_k \). If \( t_k \) increases while \( q_k \) remains fixed, the indifference curve of type \( \iota(n_{k+1}) \) passing through the point \( (a_k, t_k) \) shifts up, and so \( t_{k+1} = \hat{t}_{k+1}(\hat{j} - 1, \iota(n_{k+1})) \) (given by the intersection of this indifference curve and the dividing line \( D(\cdot; \hat{j} - 1, \iota(n_{k+1})) \)) increases. Moreover, a higher \( t_k \) also reduces \( r_k = n_{k+1} \) (because \( \mu(q_k, r_k; \hat{j}) = t_k \) and \( \mu(\cdot) \) is locally decreasing in \( r_k \)). As a consequence, \( q_{k+1} = n_{k+1} \) also decreases. Therefore \( c_{k+1} \) becomes higher according to \( \succeq \). If \( q_k \) decreases while \( t_k \) remains fixed, this reduces \( r_k \) (because \( \mu(q_k, r_k; \hat{j}) = t_k \) and \( \mu(\cdot) \) is increasing in \( q_k \)). Thus \( q_{k+1} = n_{k+1} = r_k \) falls, while there is no change in \( t_{k+1} \). Again, \( c_{k+1} \) becomes higher according to \( \succeq \). By repeatedly applying this, we show that \( c_k \) increases for all \( k = \hat{k} + 1, \ldots, \hat{k} + m \) if \( c_k \) increases slightly.  

\[\blacksquare\]
We are now in a position to describe our adjustment procedure.

**Adjustment procedure.** The current allocation path is \((c_1, \ldots, c_{k'})\) and the candidate set \(C_{k'+1}\) is undefined. Go to round \(\hat{k}\) and raise \(c_k\). Compute the corresponding \(c_{k+m} = \zeta^m_k(c_k)\) for \(m = 1, \ldots, k' - \hat{k}\). Keep raising \(c_k\) and updating the allocations for rounds \(\hat{k} + 1\) through \(k'\) to the latest values of \(\zeta^m_k(c_k)\) until one of the following occurs.

1. If for some \(m' = 1, \ldots, k' - \hat{k}\), the \(n_{k+m'}\) corresponding to \(\zeta^{m'}_k(c_k)\) decreases to a point \(n_{k+m'} \in B\):
   - reset \(\hat{k}\) to \(\hat{k} + m'\) and go back to the beginning of the adjustment procedure with the updated allocation path.
2. If \(c_k\) is raised to \(\overline{c}_k\):
   - reset \(\hat{k}\) to \(\max\{k \leq \hat{k}: \overline{c}_k > c_k\}\) and go back to the beginning of the adjustment procedure with the updated allocation path.
3. If \(\tau^{k'-\hat{k}}_k(c_k)\) becomes equal to the minimum allocation \(\mathbf{c}^{\text{min}}_{k'} \in C_{k'}\):
   - exit the adjustment procedure and return to the main algorithm at round \(k'\) with the updated allocation path.

We argue that in part 2 of the adjustment procedure, \(c_k\) will reach \(\overline{c}_k\) when \(t_k\) is still below \(\hat{j}\). Suppose we set \(t_{\hat{k}} = \hat{j}\), but we still have \(\tau_{k'}(\overline{c}_{k'}) < 0\). If \(t_{\hat{k}} = \hat{j}\), then \(t_{\hat{k}+1}\) must be greater than \(\hat{j} - 1\). Therefore \(\tau_{k+1}(c_{k+1}) = t_{k+1} - \mu^k_{k+1}(q_{k+1}) > 0\) for any \(q_{k+1}\). Since we change \(\tau_{k+1}(\cdot)\) continuously during the adjustment procedure, and since \(\tau_{k+1}(\cdot) < 0\) at the beginning of the adjustment, there must be a smaller allocation \(c'_{k}\) and the induced allocation \(c'_{k'}\) such that \(\tau_{k+1}(c'_{k+1}) = 0\). By repeating this argument, there is an even smaller allocation \(c''_{k}\) and the induced allocation \(c''_{k'}\) such that \(\tau_{k}(c''_{k'}) = 0\). However, this is a contradiction because we would have reached part 3 of the adjustment procedure before we reach that point.

This argument also implies that our adjustment procedure can always return a minimum allocation path up to round \(k'\) to the main algorithm. Since we always try to find the minimum allocation, whenever we fail to find it for the first time, we know \(\tau_{k'}(\overline{c}_{k'}) < 0\). We then adjust the allocation path continuously until we find the minimum allocation. It is thus without loss of generality to assume \(\tau_{k'}(\overline{c}_{k'}) < 0\) in any round where \(C_{k'+1}\) is undefined. Now suppose that we set \(c_k = \overline{c}_k\) for all \(k = 2, \ldots, k' - 1\) but still have \(\tau_{k'}(\overline{c}_{k'}) < 0\). In this case, we increase \(c_1\), which is unbounded, and in particular raise \(t_1\) to \(I\) while fixing \(p_1\) at 0. By the same argument, there must be an allocation \(c_1\) smaller than \((I, 0)\) and the induced allocation \(c'_{k'}\) such that \(\tau_{k'}(\overline{c}_{k'}) = 0\). Since \(C_{k'+1}\) is well defined whenever the allo-
Proof. We claim that for any only one type of agents, we must again have $(a^\ell, t^\ell)$ of substitution than any lower type $i < \iota$. Consider some agent $n$. Lemma 3.

Because the adjustment procedure can always return an allocation path to the main algorithm that allows the algorithm to proceed to the next round, the algorithm will eventually stop (the number of rounds cannot exceed $I$). In each round of this construction, the allocations are chosen in such a way that they satisfy Bayes’ rule, local incentive compatibility, and the weak pairwise-matching condition (which guarantees the D1 refinement is met). Equilibrium existence is therefore proven if we can show that the allocation path obtained at the end of the main algorithm satisfies global incentive compatibility.

In any equilibrium, there can only be a finite number of allocations that are chosen on the equilibrium path. We thus denote by $\{(a^\ell, t^\ell)\}_{\ell=1}^L$ the set of equilibrium allocations where $a^{\ell+1} > a^\ell$ and $t^{\ell+1} > t^\ell$ for all $\ell = 1, \ldots, L - 1$; throughout the proof, we assume $L \geq 3$ because there is no distinction between local and global incentive compatibility when $L < 3$. We say that local incentive compatibility holds for agent $n \in Q(a_i)$ if he weakly prefers $(a^\ell, t^\ell)$ to $(a^{\min(\ell+1, L)}_L, t^{\min(\ell+1, L)})$ and $(a^{\max(\ell-1, 1)}_L, t^{\max(\ell-1, 1)})$. Our algorithm ensures that local incentive compatibility holds for all agents.

For each $\ell$, use $i^\ell$ and $i^L$ to represent respectively the lowest and highest type choosing $(a^\ell, t^\ell)$. The following fact is useful toward establishing the main existence result.

**Lemma 3.** Consider some agent $n'$ such that $\iota(n') \leq i^L$ and his equilibrium allocation $(a^{\ell'}, t^{\ell'})$. At any point on $\phi^\ast(a; \iota(n'))$ for $a < a^{\ell'}$, type $\iota(n')$ has a lower marginal rate of substitution than any lower type $i < \iota(n')$.

**Proof.** We claim that for any $n'$ such that $\iota(n') \leq i^L$, $a^{\ell'} < D(t^{\ell'}; \iota(n'), \iota(n') - 1)$, i.e., the corresponding allocation $(a^{\ell'}, t^{\ell'})$ must locate to the left of $D(t^{\ell'}; \iota(n'), \iota(n') - 1)$. There are three cases. First, if $Q(a^{\ell'})$ is a pooling set and $\ell' < L$, $Q(a^{\ell'})$ is disconnected. Then, by the weak pairwise-matching condition, $(a^{\ell'}, t^{\ell'})$ must locate to the left of $D(t^{\ell'}; \iota(n'), \iota(n') - 1)$. Second, if $Q(a^{\ell'})$ is a pooling set and $\ell' = L$, $Q(a^{\ell'})$ must be an interval. Since the lowest type who chooses $(a^L, t^L)$ must have the highest marginal rate of substitution, $(a^L, t^L)$ must locate to the left of $D(t^L; i^L, i^L - 1)$. Finally, if $(a^{\ell'}, t^{\ell'})$ is fully separating and $Q(a^{\ell'})$ contains only one type of agents, we must again have $(a^{\ell'}, t^{\ell'})$ to the left of $D(t^{\ell'}; \iota(n'), \iota(n') - 1)$, for otherwise incentive compatibility cannot be satisfied.

27
Since any indifference curve can cross a dividing line once, the equilibrium indifference curve of type \( i(n') \) cannot cross \( D(\cdot; (n'), (n')-1) \) at any point to the left of \( a'' \). This means \( a < D(\phi^*(a; i(n')); (n'), (n')-1) \) for all \( a < a'' \). Then, by Assumption 2, we have \( a < D(\phi^*(a; i(n')); (n'), i) \) for all \( a < a'' \) and \( i < i(n') \), which proves the lemma.

**Proposition 4.** There always exists a D1 equilibrium.

**Proof.** We first show that local incentive compatibility implies global incentive compatibility for all types \( i \leq i_L \). Consider some agent \( n' \) such that \( i(n') \leq i_L \) and his equilibrium allocation \( (a^L, t^L) \). Suppose there exists some \( \ell'' > \ell' + 1 \) such that agent \( n' \) strictly prefers \((a'^{\ell''}, t^{\ell''})\) to \((a^L, t^L)\); if there are multiple allocations which agent \( n' \) strictly prefer to \((a^L, t^L)\), we let \( \ell'' \) be the smallest one among them. Local incentive compatibility suggests that there is some agent \( n'' > n' \) with \( i(n'') > i(n') \) who weakly prefers \((a''^{\ell''-1}, t^{\ell''-1})\) to \((a'^{\ell''}, t^{\ell''})\). This means that \( \phi^*(\cdot; i(n')) \) must cross \( \phi^*(\cdot; i(n'')) \) from above at some \( a < a'' \) where as defined above, \( \phi^*(\cdot; i) \) is the indifference curve of type \( i \) that passes through his equilibrium allocation. This contradicts Lemma 3 because type \( i(n') \) must have a higher marginal rate of substitution than type \( i(n'') \) at any point on \( \phi^*(a; i(n')) \) for \( a < a'' \). Similarly, suppose there exists some \( \ell'' < \ell' - 1 \) such that agent \( n' \) strictly prefers \((a'^{\ell''}, t^{\ell''})\) to \((a^L, t^L)\); if there are multiple allocations which agent \( n' \) strictly prefers to \((a^L, t^L)\), we let \( \ell'' \) be the largest one among them. Then, by local incentive compatibility, there is some agent \( n'' < n' \) with \( i(n'') < i(n') \) who weakly prefers \((a''^{\ell''+1}, t^{\ell''+1})\) to \((a'^{\ell''}, t^{\ell''})\). This means that \( \phi^*(\cdot; i(n')) \) must cross \( \phi^*(\cdot; i(n'')) \) from below at some \( a < a_{\ell''+1} \). This is again a contradiction because type \( i(n') \) must have a lower marginal rate of substitution than type \( i(n'') \) at any point on \( \phi^*(a; i(n')) \) for \( a < a_{\ell''+1} \).

If \( Q(a_L) \) is a pooling set, there exists agent \( n' \) such that \( \overline{i} \geq i(n') > i_L \). By the weak pairwise-matching condition, for any such \( n' \), we have \( MRS(a^L, t^L, i_L) \geq MRS(a^L, t^L, i(n')) \). This means \( \phi^*(a; i_L) < \phi^*(a; i(n')) \) for all \( a < a^L \). Since type \( i_L \) prefers \((a^L, t^L)\) to any other allocation, so does type \( i(n') \).

Finally, consider agent \( n' \) such that \( i(n') > i_L \). If there is some agent \( n'' \in Q(a^L) \) such that \( i(n'') < i(n') \) and \( MRS(a^L, t^L, i(n'')) = MRS(a^L, t^L, i(n'')) \), then \( \phi^*(a; i(n'')) \geq \phi^*(a; i(n')) \) for all \( a \) by the double-crossing property. Since agent \( n'' \) prefers \((a^L, t^L)\) to any other allocation, so does agent \( n' \). If there are no such agents who have the same marginal

---

13 To see this, observe that \( Q(a''^{\ell''-1}) \) cannot contain any type \( i(n') \) agent because otherwise agent \( n' \) must be indifferent between \((a^{\ell''}, t^{\ell''})\) and \((a''^{\ell''-1}, t^{\ell''-1})\). If this is the case, local incentive compatibility implies that agent \( n' \) weakly prefers \((a^{\ell''}, t^{\ell''})\) to \((a''^{\ell''}, t^{\ell''})\).
rate of substitution, by the weak pairwise-matching condition, there must be agent \( n'' \in Q(a'') \) such that \( \iota(n') > \iota(n'') > \iota' \) and \( \text{MRS}(a', t', \iota(n'')) < \text{MRS}(a', t', \iota(n')) < \text{MRS}(a', t', \iota') \). This means that \( \phi^*(a; \iota(n')) > \phi^*(a; \iota(n'')) \) for all \( a > a' \) and \( \phi^*(a; \iota(n')) > \phi^*(a; \iota'') \) for all \( a < a' \), i.e., \( \phi^*(\cdot; \iota(n')) \) stays above the lower envelope of \( \phi^*(\cdot; \iota(n'')) \) and \( \phi^*(\cdot; \iota'') \).

Since both types \( \iota(n'') \) and \( \iota' \) prefer \( (a', t') \) to any other allocation, so does type \( \iota(n') \). This proves that local incentive compatibility implies global incentive compatibility for all agents.

We remark that except in the case of \( I = 2 \) described by Proposition 3, equilibrium is typically not unique. There are generally many equilibria because the candidate set specified for our main algorithm is only a subset of all the possible allocations that satisfy local incentive compatibility and the weak pairwise-matching condition. To illustrate this possibility, define

\[
\text{t}_{\text{max}}(a, t) := \min \left\{ \arg \max_{i=1, \ldots, I} \text{MRS}(a, t, i) \right\},
\]

\[
\text{t}_{\text{min}}(a, t) := \min \left\{ \arg \min_{i=1, \ldots, I} \text{MRS}(a, t, i) \right\},
\]

for a given allocation \((a, t)\),\(^{14}\) and consider a full pooling equilibrium in which all types choose \((a_p, \mathbb{E}[i])\). This constitutes a perfect Bayesian equilibrium under some off-equilibrium beliefs if \( a_p \) satisfies

\[
u(a_p, \mathbb{E}[i], 1) \geq u(0, 1, 1). \tag{2}\]

Moreover, this pooling equilibrium satisfies D1 if

\[
\mathbb{E}[i] \geq \max \left\{ \text{t}_{\text{max}}(a_p, \mathbb{E}[i]), \text{t}_{\text{min}}(a_p, \mathbb{E}[i]) \right\}. \tag{3}\]

It is easy to see that for any given \( \mathbb{E}[i] \), there is a range of \( a_p \) that satisfies (2). This means that we may have a continuum of pooling equilibria if there is a range of \( a_p \) that also satisfies (3). This is generally the case if there are three or more types. Suppose, for instance, that there are three types with \( \mathbb{E}[i] \geq 2 \). In this case, (3) is satisfied if \( \text{t}_{\text{max}}(a_p, \mathbb{E}[i]) = 1 \) and \( \text{t}_{\text{min}}(a_p, \mathbb{E}[i]) = 2 \), which hold for any allocation that is bounded between \( D(\cdot; 3, 2) \) and \( D(\cdot; 3, 1) \); i.e., any \((a_p, \mathbb{E}[i])\) such that \( D(\mathbb{E}[i]; 3, 2) \leq a_p \leq D(\mathbb{E}[i]; 3, 1) \).

\(^{14}\) When types are discrete, \( \arg \max_{i=1, \ldots, I} \text{MRS}(a, t, i) \) and \( \arg \min_{i=1, \ldots, I} \text{MRS}(a, t, i) \) may not be singleton. In this case, we always pick the smaller type because D1 has no bite and we can assign an arbitrary off-equilibrium belief.
As such, if there is some action \( a_p \) such that (2) holds with strict inequality and also

\[
D(\mathbb{E}[i]; 3, 2) < a_p < D(\mathbb{E}[i]; 3, 1),
\]

both (2) and (3) continue to hold for any \( a \) in a small neighborhood of \( a_p \). If there are only two types, on the other hand, \( \mathbb{E}[i] \in (1, 2) \) and (3) can be satisfied if and only if \( t_{\max}(a_p, \mathbb{E}[i]) = t_{\min}(a_p, \mathbb{E}[i]) = 1 \), i.e., the allocation \( (a_p, \mathbb{E}[i]) \) must be exactly on the dividing line as stated in Proposition 3. Since the dividing line is a function of \( t \), there is a unique value of \( a_p \) that can satisfy this, thereby eliminating any possibility of multiple equilibria.

6. Convergence to the Continuous-Type Model

Given our results, a natural question to ask is whether an equilibrium of our discrete-type model (hereafter, discrete-type equilibrium) can approximate that of the continuous-type model of Chen et al. (2022) (continuous-type equilibrium) as the number of types becomes sufficiently large. To shed light on this issue, we now consider a sequence of discrete-type models with a payoff function satisfying Assumptions 1–3 of Chen et al. (2022).\(^{15}\) Each discrete-type model in the sequence is characterized by \( G^v := (\Theta_v, f_v) \), \( v = 1, 2, \ldots \), where \( \Theta_v := \{\theta_1^v, \ldots, \theta_{I(v)}^v\} \) is the type space and \( f_v \in \Delta(\Theta_v) \) is the density function. The limit of this sequence is a continuous-type model characterized by \( G := (\Theta, f) \) where \( \Theta := [\underline{\theta}, \bar{\theta}] \) and \( f \in \Delta(\Theta) \). Throughout this section, we abuse notation and denote the allocation of type \( \theta \) by \( (a^v(\theta), t^v(\theta)) \) to be consistent with the continuous-type model.\(^{16}\) In what follows, we eliminate superscript \( v \) from \( \theta^v \) and simply write \( \Theta^v := \{\theta_1, \ldots, \theta_{I(v)}\} \) for brevity. We also assume \( \Theta^v \subset \Theta \) and let \( \theta_0 := \underline{\theta} \) and \( \theta_{I(v)+1} := \bar{\theta} \) for notational convenience.

Chen et al. (2022) show that any continuous-type equilibrium must be Low types Separate High types Pairwise-Pool (LSHPP). To ensure that an arbitrary continuous-type equilibrium can be approximated by a discrete-type equilibrium when types are dense enough, we require each discrete-type equilibrium in the convergent sequence to have this property as well. The discrete-type counterpart of this property can be stated as follows.

**Definition 3.** For a given \( G^v \), the sender's strategy is LSHPP if there is some \( \hat{\theta}^v \in \Theta^v \) such that:

(a) Each type \( \theta_i < \hat{\theta}^v \) chooses a fully separating action;

\(^{15}\)These assumptions are identical to Assumptions 1–3 of this paper, except that Assumption 3 is now defined over the continuous type space and \( D(\cdot; \theta', \theta'') \) is assumed to be continuous in \( \theta' \) and in \( \theta'' \).

\(^{16}\)As we note below, throughout this section, we focus on LSHPP equilibria, which implies that no types randomize over different actions.
(b) There exist $\theta^v_s \in \Theta^v$ and $p^v : [\hat{\theta}^v, \theta^v_s] \rightarrow [\theta^v_s, \theta_{I(v)}]$ such that: (i) $p^v(\cdot)$ is strictly decreasing on $[\hat{\theta}^v, \theta^v_s]$ with $p^v(\hat{\theta}^v) = \theta_{I(v)}$; (ii) $a^v(\cdot)$ is strictly increasing on $[\hat{\theta}^v, \theta^v_s]$ with $a^v(\theta_i) = a^v(p(\theta_i))$ for all $\theta_i \in [\hat{\theta}^v, \theta^v_s]$.

An equilibrium is an LSHPP equilibrium if the sender’s strategy is LSHPP. Observe that each LSHPP equilibrium is characterized by $x^v := (a^v, t^v, p^v)$, where $p^v(\cdot)$ indicates how types are paired together. The existence of the matching function $p^v(\cdot)$ is an essential property of LSHPP equilibria, which implies that the mapping above the gap type $\hat{\theta}^v$ is bijective, and $Q(a(\theta_i))$ is a pooling set for all $\theta_i \in [\hat{\theta}^v, \theta^v_s)$. Let $\Omega(G^v)$ be the set of all LSHPP equilibria for a given discrete-type model $G^v$ with its element $x^v \in \Omega(G^v)$. Below, we consider a pair $(G^v, x^v)$ of a discrete-type model $G^v$ and an LSHPP equilibrium $x^v \in \Omega(G^v)$. Similarly, let $\Omega(G)$ be the set of all LSHPP equilibria for a given continuous-type model $G$ with its element $x := (a, t, p) \in \Omega(G)$. Theorem 2 of Chen et al. (2022) ensures that $\Omega(G)$ is always nonempty.

We say that a sequence of discrete functions $g^v : \Theta^v \rightarrow \mathbb{R}$ converges to a continuous function $g : \Theta \rightarrow \mathbb{R}$ if (i) $\lim_{v \rightarrow \infty} \max_{i \in \{1, \ldots, I(v)+1\}} |\theta_i - \theta_{I(v)+1}| = 0$; and (ii) $\lim_{v \rightarrow \infty} \max_{i \in \{1, \ldots, I(v)\}} |g^v(\theta_i) - g(\theta_i)| = 0$. The following result shows that there exists a sequence of discrete-type equilibria converging to a continuous-type equilibrium.

**Proposition 5.** Fix a continuous-type model $G := (\Theta, f)$ and an LSHPP equilibrium $x \in \Omega(G)$. For any given $(G, x)$, there is a sequence $\{(G^v, x^v)\}$, with $x^v \in \Omega(G^v)$ for each $v = 1, 2, \ldots$, that converges to $(G, x)$, i.e., $f^v \rightarrow f$ and $x^v \rightarrow x$ as $v \rightarrow \infty$.

The proof of Proposition 5 is provided in Appendix B. In the proof, the convergent sequence of equilibria $x^v$ is constructed by carefully picking the discrete type space $\Theta^v$ so that $\theta_i$ and $p(\theta_i)$ both belong to $\Theta^v$. The equilibrium $x^v$ so obtained is LSHPP by construction. However it is also possible to pick a different sequence of equilibria $\tilde{x}^v$ which converges to the continuous-type equilibrium $x$, where each $\tilde{x}^v$ along this sequence is a discrete-type equilibrium but is not an LSHPP equilibrium, i.e., an equilibrium where there are some types above $\hat{\theta}^v$ that are fully separated and hence $p^v(\cdot)$ cannot be well defined. In Appendix C, we provide an example to make this point. This example suggests that not all discrete-type models can approximate a continuous-type model in terms of the pattern of pooling, no matter how dense the type distribution becomes.

31
7. Conclusion

In this paper we obtain a characterization of signaling equilibrium under double-crossing preferences that holds for an arbitrary number of discrete types, and based on this characterization, provide a recipe that allows an explicit construction of equilibrium, complementing our earlier work for the case of continuous types. We establish the existence of a D1 equilibrium for any discrete-type distribution via this construction. We also highlight the connection between discrete-type and continuous-type models by studying convergence properties, which allows us to illuminate robust predictions of signaling under double-crossing preferences.
Appendix A: The Set of Candidate Allocations

In this appendix, we first provide a more detailed account of how we construct the candidate set $C_k$ when $n_k \in B$ and how we pin down the corresponding action $a_k = \alpha(t_k)$ from $c_k = (t_k, q_k) \in C_k$. At the end of the appendix, we also provide a self-contained summary of the process under various contingencies.

Since case (i) (where there is a separating equilibrium for types between $\iota(n_k)$ and $j$) is fully described in the main text, we here focus on case (ii) (where it is not feasible to construct a separating equilibrium for types between $\iota(n_k)$ and $j$). In this case, the set of candidate allocations often traces the indifference curve of some relevant type that passes through his reservation allocation. To obtain the corresponding action $a_k = \alpha(t_k)$ from the choice of $t_k$, it is thus convenient to define an indifference curve as a function of reputation $t$: define $\psi(\cdot, i; \tilde{a}_k, \tilde{t}_k)$, or $\psi_k(\cdot, i)$ for short, to be the indifference curve of type $i$ that passes through some reservation allocation $(\tilde{a}_k, \tilde{t}_k)$ (i.e., $\psi_k(\cdot, i)$ is the inverse of the indifference curve $\phi(\cdot, u(\tilde{a}_k, \tilde{t}_k, i), i)$ introduced in the text). Also, as in the main text, define $(\hat{a}_k(j, i), \hat{t}_k(j, i))$ such that

$$u(\hat{a}_k(j, i), \hat{t}_k(j, i), i) = u(\tilde{a}_k, \tilde{t}_k, i),$$

$$\hat{a}_k(j, i) = D(\hat{t}_k(j, i); j, i),$$

for a given reservation allocation $(\tilde{a}_k, \tilde{t}_k)$. It is important to note that the reservation allocation $(\tilde{a}_k, \tilde{t}_k)$ depends on $c_{k-1}$ and $q_k$; as such, both $\psi_k(\cdot, i)$ and $(\tilde{a}_k, \tilde{t}_k)$ depend on them as well, although we do not explicitly indicate this dependence for brevity.

We have already shown that when $n_k \in B$, the candidate set $C_k$ is given by

$$q_k \in [n_k, F_{i_k}^{-}] \text{ and } t_k \begin{cases} = \hat{t}_k(j, \iota(q_k)) & \text{if } q_k \notin B, \\ \in [\max\{\hat{t}_k(j, \iota(q_k)), \hat{t}_k(j, \iota(q_k) - 1)\}, \hat{t}_k(j, \iota(q_k) - 1)) & \text{if } q_k \in B, \end{cases}$$

where we let $\hat{t}(j, 0) = \infty$. In each round of our algorithm, we need to search for a tentative allocation over this two-dimensional space $C_k$. Below, we will describe how we pin down the corresponding action $a_k = \alpha(t_k)$ from $c_k = (t_k, q_k) \in C_k$.

First, if $q_k \notin B$, some type $\iota(q_k)$ agents choose a separating action. We can then uniquely pin down $t_k = \hat{t}_k(j, \iota(q_k))$ with $(\tilde{a}_k, \tilde{t}_k) = (s_k^r(\iota(q_k)), \iota(q_k)))$. This is the easy case to deal with, because we can also uniquely pin down

$$a_k = \alpha(\hat{t}_k(j, \iota(q_k))) = \hat{a}_k(j, \iota(q_k)).$$
Second, if \( q_k \in B \) and \( \iota(q_k) > 1 \), type \( \iota(q_k) \) separates from type \( \iota(q_k) - 1 \) and pool with type \( j \). Both types \( \iota(q_k) - 1 \) and \( \iota(q_k) \) are relevant, and any candidate allocation must be bounded between \( \psi_k(\cdot, \iota(q_k) - 1) \) and \( \psi_k(\cdot, \iota(q_k)) \) with the reservation allocation given by (1). Moreover, to pool with type \( j \), the weak pairwise-matching condition requires that any candidate allocation must be bounded between \( D(\cdot; j, \iota(q_k)) \) and \( D(\cdot; j, \iota(q_k) - 1) \). The feasible range of \( t_k \) is therefore

\[
t_k \in \left[ \max \{ \hat{t}_k(j, \iota(q_k)), \bar{t}_k \}, \hat{t}_k(j, \iota(q_k) - 1) \right],
\]

where \( \hat{t}_k \geq \hat{t}_k(j, \iota(q_k)) \) only if \( q_k = F_{t_k} \). For \( t_k \) in this range, if the reservation allocation is to the left of \( D(\cdot; j, \iota(q_k)) \), we first move up along the dividing line \( D(\cdot; j, \iota(q_k)) \) until it hits \( \psi_k(\cdot, \iota(p_k) - 1) \), and then trace this indifference curve; if the reservation allocation is to the right of \( D(\cdot; j, \iota(q_k)) \), we simply trace \( \psi_k(\cdot, \iota(q_k) - 1) \). To sum up,

\[
\alpha_k(\cdot) = \max \{ D(\cdot; j, \iota(q_k)), \psi_k(\cdot, \iota(q_k) - 1) \}.
\]

Finally, if \( q_k = 0 \), the set of candidate allocations is unbounded; observe that this can happen only in round 1. Any allocation on \( D(\cdot; I, 1) \) can be a candidate allocation as long as it gives type 1 a payoff at least as large as the reservation allocation \( (\bar{a}_1, \bar{t}_1) = (0, 1) \). The feasible range of \( t_1 \) is therefore \( t_1 \in \left[ \max \{ \hat{t}_1(I, 1), 1 \}, \infty \right) \) with

\[
\alpha_1(\cdot) = \max \{ D(\cdot; I, 1), 0 \}.
\]

Applying the above rules, we summarize how we determine candidate allocations under various contingencies as follows.

1. When \( \hat{t}_k(j, \iota(n_k)) \leq t_{k-1} \), i.e., the dividing line of \( j \) and \( \iota(n_k) \) is to the left of \( (a_{k-1}, t_{k-1}) \), \( q_k = n_k \), and the allocation for round \( k \) is set the same as \( (a_{k-1}, t_{k-1}) \).

2. When \( n_k \notin B \), the candidate set \( C_k \) is a singleton with \( q_k = n_k \), and the allocation \( (a_k, t_k) \) is \( (\hat{a}_k(j, i), \hat{t}_k(j, i)) \) (with reservation allocation \( (\bar{a}_k, \bar{t}_k) = (a_{k-1}, t_{k-1}) \)).

3. When \( n_k \in B \) and \( t_k \geq \iota(n_k) \), type \( \iota(n_k) \) agents cannot take a separating action and must pool with type \( j \). Therefore, \( q_k = n_k \in B \), and candidate allocations are as shown in Figure 3(a).

4. When \( n_k \in B \) and \( t_k < \iota(n_k) \), this is the only case where \( q_k \) in the candidate set can assume multiple values. Figure 3(b) illustrates candidate allocations when \( i_k^* = \)
Figure 3. Panel (a) depicts the case where \( n_k \in B \) and \( t_k \geq \iota(n_k) \). Panel (b) depicts the case where \( n_k \in B \) and \( t_k < \iota(n_k) \).

\( \iota(n_k) + 1 \) under this particular contingency. Note that the segment between points \( A \) and \( B \) corresponds to \( q_k = F_{i_k^* + 1} \), point \( B \) corresponds to \( q_k \in (F_{i_k^*}^l, F_{i_k^* + 1}^l) \), the segment between points \( B \) and \( D \) corresponds to \( q_k = F_{i_k^*} \), point \( D \) corresponds to \( q_k \in (F_{i_k^* - 1}^l, F_{i_k^*}^l) \), and the segment above \( D \) corresponds to \( q_k = F_{i_k^* - 1} \). If \( i_k^* = 1 \), \( \psi(\cdot; i_k^* - 1) \) and \( D(\cdot; j, i_k^* - 1) \) do not exist. In this case, point \( C \) corresponds to \( q_k \in (F_{i_k^* - 1}^l, F_{i_k^*}^l) \), and the extended segment from \( C \) to \( D(I; j, i_k^* - 1) \) along \( D(\cdot; j, i_k^*) \) corresponds \( q_k = F_{i_k^* - 1} \). Additionally, notice that in this figure, we assume that \( s_k^*(i_k^* + 1) < D(i_k^* + 1; j, i_k^*) \). If \( s_k^*(i_k^* + 1) \geq D(i_k^* + 1; j, i_k^*) \), the segment between \( A \) and \( B \) is not included, and the process starts from point \( B \).

Appendix B: Proof of Proposition 5

We first construct a tentative type space \( \hat{\Theta}^v \), using \( p(\cdot) \) of the continuous-type model. Chen et al. (2022) show that in any LSHPP equilibrium, there can be two types of pooling—atomless pooling where \( a(\cdot) \) is strictly increasing and mass pooling where \( a(\cdot) \) is constant—and \( a(\cdot) \) is differentiable almost everywhere (except at points where the equilibrium switches between atomless pooling and mass pooling). Given some equilibrium \( x \) of the continuous-type model \( G \), define

\[
\mathcal{D} := \{ \theta \in (\hat{\Theta}, \Theta) : a(\cdot) \text{ is differentiable at } \theta \},
\]
and $D := [\hat{\theta}, \theta_\ast] \setminus \Theta$, where $\hat{\theta}, \theta_\ast \in D$ by definition. Also, define

$$AP := \{ \theta \in \Theta : a'(\theta) > 0 \}, \quad MP := \{ \theta \in \Theta : a'(\theta) = 0 \}.\$$

Assign all types in $D$ to $\hat{\Theta}$. Also, for each $\epsilon := 1/\nu$, add $\theta_i \in (\underbar{\theta}, \theta_\ast)$ and each respective paired type $p(\theta_i)$ for $\theta_i \in (\hat{\theta}, \theta_\ast)$ to $\hat{\Theta}$ such that $|\theta_i - \theta_{i-1}| < \epsilon$. Note that this type space is only tentative as we will remove some of the types around $\theta_\ast$.

Given this tentative type space, we next describe how to determine $a^\nu(\cdot)$, $t^\nu(\cdot)$ and $f^\nu(\cdot)$ for each $\theta_i \in \hat{\Theta}$. For $\theta_i < \hat{\theta}$, $a^\nu(\theta_i)$ and $t^\nu(\theta_i)$ are determined by the least-cost separating solution. For $\theta_i \geq \hat{\theta}$, given $a^\nu(\theta_{i-1})$ and $t^\nu(\theta_{i-1})$, if $\theta_i \in AP \cup D$, we determine $a^\nu(\theta_i)$ and $t^\nu(\theta_i)$ by solving

$$MRS(a^\nu(\theta_i), t^\nu(\theta_i), \theta_i) = MRS(a^\nu(\theta_i), t^\nu(\theta_i), p^\nu(\theta_i)), \quad u(a^\nu(\theta_i), t^\nu(\theta_i), \theta_i) = u(a^\nu(\theta_{i-1}), t^\nu(\theta_{i-1}), \theta_{i-1}).$$

The first condition requires two paired types in atomless pooling to have the same marginal rate of substitution; the second condition is the constraint to prevent type $\theta_{i-1}$ to deviate upward. If $\theta_i \in MP$, let $a^\nu(\theta_i) = a^\nu(\theta_{i-1})$ and $t^\nu(\theta_i) = t^\nu(\theta_{i-1})$. If the derived $t^\nu(\theta_i)$ is weakly larger than $\theta_i$ or smaller than $\theta_i$, we then stop the procedure and remove $\Theta^\nu := \{\theta_i, \ldots, \theta_\ast, \ldots, p(\theta_i)\}$ from $\hat{\Theta}$. The type space of $G^\nu$ is given by $\Theta^\nu := \hat{\Theta} \setminus \Theta^\nu$. Note that $\Theta^\nu$ always includes $\theta_\ast$ for any $\nu$. Let $\theta^\nu_\ast := \max\{\theta_i \in \Theta^\nu : \theta_i < \theta_\ast\}$.

To pin down $f^\nu(\cdot)$, let $F^\nu(\cdot)$ be the cumulative type distribution function with $F^\nu(\theta_0) = 0$ and $F^\nu(\theta_{I(\nu)+1}) = 1$. Define

$$f^\nu(\theta_i) := \begin{cases} F^\nu(\theta_i) - F^\nu(\theta_{i-1}) & \text{for } \theta_i \in [\theta_1, \theta^\nu_\ast] \cap \Theta^\nu, \\ F^\nu(\theta_{I(\nu)}) - F^\nu(\theta_i) & \text{for } \theta_i \in (\theta^\nu_\ast, \theta_{I(\nu)}) \cap \Theta^\nu. \end{cases}$$

We first obtain $\tilde{f}^\nu(\cdot)$ that is consistent with Bayes’ rule given $t^\nu(\cdot)$ and then derive $f^\nu(\cdot)$ by normalizing $\tilde{f}^\nu(\cdot)$. Specifically, for $\theta_i \leq \theta^\nu_\ast$, we fix $\tilde{f}^\nu(\theta_i)$ by the density function of the continuous-type model, i.e., $\tilde{f}^\nu(\theta_i) = f(\theta_i)$. Given this, for $\theta_i \in [\hat{\theta}, \theta^\nu_\ast]$, we choose $\tilde{f}^\nu(p(\theta_i))$ to satisfy

$$t^\nu(\theta_i) = \frac{f(\theta_i)\theta_i + \tilde{f}^\nu(p(\theta_i))\frac{p(\theta_{i-1}) - p(\theta_i)}{\theta_i - \theta_{i-1}} p(\theta_i)}{f(\theta_i) + \tilde{f}^\nu(p(\theta_i))\frac{p(\theta_{i-1}) - p(\theta_i)}{\theta_i - \theta_{i-1}}}.$$  \hspace{1cm} (4)

\footnote{In Appendix C, we show that the allocation $(a^\nu(\cdot), t^\nu(\cdot))$ of $G^\nu$ is consistently above the allocation $(a(\cdot), t(\cdot))$ of $G$. This implies that there exists some $\theta' < \theta_\ast$ such that $t^\nu(\theta') = \theta_\ast$ and hence $\Theta^\nu$ always includes $\theta_\ast$.}
Observe that the probabilities given by \( \tilde{f}^{v}(\cdot) \) do not add up to 1. To construct a density function, we normalize \( \tilde{f}^{v}(\cdot) \) to obtain

\[
f^{v}(\theta) = \frac{\tilde{f}^{v}(\theta)}{\sum_{\theta_{i} \in [\theta_{i}, \theta_{i+1}]} \tilde{f}^{v}(\theta_{i})(\theta_{i} - \theta_{i-1}) + \sum_{\theta_{i} \in (\theta_{i}, \theta_{i+1})} \tilde{f}^{v}(\theta_{i})(\theta_{i+1} - \theta_{i})}.
\]  

(5)

We now establish that the sequence \( x^{v} = (a^{v}, t^{v}, p^{v}) \) constructed above converges to \( x = (a, t, p) \) as \( v \to \infty \). In this proof, we consider a particular algorithm described in Chen et al. (2022), which leads to a particular system of differential equations, to construct an equilibrium, but the following argument holds for an arbitrary equilibrium of any continuous-type model.\(^{18}\) Since we fix an equilibrium \( p(\cdot) \), with abuse of notation, let \( x = (a, t) \) and \( x^{v} = (a^{v}, t^{v}) \). Also, let

\[
\tilde{w}(\theta, x, p) := MRS_{\theta}(a, t, \theta) - MRS_{\theta}(a, t, p)
\]

\[
= \frac{f(\theta) \theta - t}{f(p) p - t},
\]

\[
w(\theta, x, p) := \max\{\tilde{w}(\theta, x, p)I(MRS(a, t, \theta) = MRS(a, t, p)), 0\},
\]

where subscripts to \( MRS(\cdot) \) stand for partial derivatives with respect to the specified variables. Following the algorithm, taking \( p(\cdot) \) as given, \( a(\cdot) \) and \( t(\cdot) \) of the continuous-type model \( G \) follow a system of differential equations of the form \( x^{v} = W(\theta, x^{v}) \) where

\[
\begin{align*}
a^{v} &= \frac{w(\theta, x, p)}{MRS_{\theta}(a, t, p) - MRS_{\theta}(a, t, \theta) + MRS_{\theta}(a, t, \theta)MRS_{\theta}(a, t, p) - MRS_{\theta}(a, t, \theta)}, \\
t^{v} &= \frac{MRS_{\theta}(a, t, \theta)w(\theta, x, p)}{MRS_{\theta}(a, t, p) - MRS_{\theta}(a, t, \theta) + MRS_{\theta}(a, t, \theta)MRS_{\theta}(a, t, p) - MRS_{\theta}(a, t, \theta)}.
\end{align*}
\]

See Chen et al. (2022) (Section 5 and Appendix C) for more detail. Similarly, by construction, \( a^{v}(\cdot) \) and \( t^{v}(\cdot) \) follow \( x^{v} = W^{v}(\theta, x^{v}) \) where

\[
\begin{align*}
a^{v} &= \frac{w(\theta, x, v, p)}{MRS_{\theta}(a^{v}, t^{v}, p) - MRS_{\theta}(a^{v}, t^{v}, \theta) + MRS_{\theta}(a^{v}, t^{v}, \theta)MRS_{\theta}(a^{v}, t^{v}, p) - MRS_{\theta}(a^{v}, t^{v}, \theta)} \\
t^{v} &= \frac{MRS_{\theta}(a^{v}, t^{v}, \theta)w(\theta, x, v, p)}{MRS_{\theta}(a^{v}, t^{v}, p) - MRS_{\theta}(a^{v}, t^{v}, \theta) + MRS_{\theta}(a^{v}, t^{v}, \theta)MRS_{\theta}(a^{v}, t^{v}, p) - MRS_{\theta}(a^{v}, t^{v}, \theta)}
\end{align*}
\]

and \( \eta(\theta) := \max\{\theta_{i} \in \Theta^{v} : \theta_{i} < \theta\} \).\(^{19}\) Notice a slight difference between the two systems, where we have \( MRS(a^{v}, t^{v}, \eta(\theta)) \), instead of \( MRS(a^{v}, t^{v}, \theta) \), in \( W^{v} \) because for

\(^{18}\) As noted in Chen et al. (2022), each continuous-type model \( G \) may admit a continuum of equilibria because there is some leeway for determining transition points between atomless pooling and mass pooling. The algorithm developed in that paper seeks to produce atomless pooling as long as it is feasible to do so. We could alternatively have an algorithm that switches from atomless pooling to mass pooling at an earlier point where there is still a solution for atomless pooling. Our argument in the proof does not depend on the form of algorithm we employ to construct an equilibrium, and hence applies to any arbitrary equilibrium.

\(^{19}\) Here, we take both \( a^{v}(\cdot) \) and \( t^{v}(\cdot) \) as continuous and differentiable functions although their domains are discrete.
\( \theta_i \in AP, (a^v(\theta_{i+1}), t^v(\theta_{i+1})) \) and \((a^v(\theta_i), t^v(\theta_i))\) are connected by the indifference curve of type \( \theta_i \). Note that \( W^v(\theta, x) = W(\theta, x) \) for \( \theta \in \Theta^v \). Moreover, since we fix \( p(\theta) \), \( W(\theta, x) \) is Lipschitz-continuous in \( x \) given \( \theta \), i.e., there exists some \( L > 0 \) such that \( |W(\theta, x_2) - W(\theta, x_1)| \leq L|x_2 - x_1| \) for any \( x_1 \) and \( x_2 \). It then follows that

\[
|x(\theta_i) - x^v(\theta_i)| \leq |x(\theta_{i-1}) + (\theta_i - \theta_{i-1})x^v(\theta_{i-1}) + \delta| + |x^v(\theta_{i-1}) + (\theta_i - \theta_{i-1})x^v(\theta_{i-1}) + \delta^v|
\]

\[
\leq |x(\theta_{i-1}) - x^v(\theta_{i-1})| + \epsilon|W(\theta_{i-1}, x(\theta_{i-1}))-W^v(\theta_{i-1}, x^v(\theta_{i-1}))| + \delta + \delta^v
\]

where

\[
\delta := \max_{\theta_i \in \Theta^v} |x(\theta_i) - x(\theta_{i-1}) - (\theta_i - \theta_{i-1})x'(\theta_{i-1})|,
\]

\[
\delta^v := \max_{\theta_i \in \Theta^v} |x^v(\theta_i) - x^v(\theta_{i-1}) - (\theta_i - \theta_{i-1})x^{v'}(\theta_{i-1})|.
\]

Moreover,

\[
|x(\theta_i) - x^v(\theta_i)| \leq (1 + \epsilon L)|x(\theta_{i-1}) - x^v(\theta_{i-1})| + \delta + \delta^v
\]

\[
\leq (1 + \epsilon L)[(1 + \epsilon L)|x(\theta_{i-2}) - x^v(\theta_{i-2})| + \delta + \delta^v] + \delta + \delta^v
\]

\[
\leq (1 + \epsilon L)^{i-1}|x(\theta_0) - x^v(\theta_0)| + [1 + (1 + \epsilon L) + \cdots + (1 + \epsilon L)^{i-2}](\delta + \delta^v)
\]

\[
= (1 + \epsilon L)^{i-1}|x(\theta_0) - x^v(\theta_0)| + \frac{(1 + \epsilon L)^{i-1} - 1}{\epsilon L}(\delta + \delta^v).
\]

Note that \( |x(\theta_0) - x^v(\theta_0)| = O(\epsilon) \) and \( \delta + \delta^v = O(\epsilon^2) \). Also, \( (1 + \epsilon L)^{i-1} \) converges as \( \nu \to \infty \) (or alternatively as \( \epsilon \to 0 \)) because \( (1 + \epsilon L)^{i-1} < (1 + \epsilon L)^{(\nu)} \) and \( (1 + \epsilon L)^{(\nu)} \) converges as \( \nu \to \infty \).

To ensure that \( x^v(\cdot) \) converges to \( x(\cdot) \) for all \( \theta \in [\theta_0, \theta_s] \), we also need to show that \( \theta^v \) converges to \( \theta_s \). This holds because \( t^v(\theta_i) \) converges to \( t(\theta_i) \) and \( \theta_i \leq t^v(\theta_i) < \theta_s \) for all \( \theta < \theta_s \). Finally, the convergence of \( f^v(\cdot) \) to \( f(\cdot) \) can be established from (4) and (5).

**Appendix C: Non-LSHPP Equilibria in the Limit**

In this appendix, we construct a discrete-type equilibrium that is not LSHPP even as \( \nu \to \infty \). To this end, consider a discrete-type model \( G^v \) and an LSHPP equilibrium allocation \( x^v \), as constructed in Appendix B, such that \( \hat{\theta} > \theta_1 \) and

\[
\text{MRS}(a^v(\theta_1^v), t^v(\theta_1^v), \theta_1^v) = \text{MRS}(a^v(\theta_s^v), t^v(\theta_s^v), p(\theta_s^v)).
\]
Define $\phi^\nu(\cdot; \theta)$ as the equilibrium indifference curve of type $\theta$ in equilibrium $x^\nu$. Below, we show that for any discrete-type model in the sequence, it is always possible to construct a non-LSHPP equilibrium for an arbitrarily small $\varepsilon := 1/\nu$ by adding two more types, $\theta_{j_1}$ and $\theta_{j_2}$, in $(\nu(\theta_{s_i}^k), \theta_s)$.

To construct such an equilibrium, we first establish that $x^\nu = (a^\nu, t^\nu)$ in the convergent sequence always stays above the continuous-type equilibrium $x^\nu = (a, t)$ in the $(a, t)$-space. More precisely, let $\varepsilon > 0$, there is always a space to add $\varepsilon$ for an arbitrarily small $\phi \in A$. Define $T$ the set of actions that are chosen on the equilibrium path in equilibrium $x$. Then, since there is always a one-to-one relationship between action $a$ and reputation $t$ in any equilibrium, the equilibrium reputation can be written as $T(a)$ for each $a \in A$. Similarly, the equilibrium reputation can be written as $T^\nu(a)$ for each $a \in A^\nu$, where $A^\nu$ is the set of actions that are chosen on the equilibrium path in equilibrium $x^\nu$. Under this formulation, we show that $T^\nu(a) > T(a)$ for all $a \in A^\nu$, or alternatively, $\phi^\nu(a^\nu(\theta_i); \theta_i) > \phi^\nu(a^\nu(\theta_i); \theta_i)$ for all $\theta_i \in [\theta_1, \theta^\nu_1] \cap \Theta^\nu$. Once we establish this claim, we then show that there is always a space to add $\theta_{j_1}$ and $\theta_{j_2}$ such that the resulting equilibrium is not LSHPP for any $\varepsilon > 0$.

We show that $\phi^\nu(a^\nu(\theta_i); \theta_i) > \phi^\nu(a^\nu(\theta_i); \theta_i)$ for all $\theta_i \in [\theta_1, \theta^\nu_1] \cap \Theta^\nu$ by induction. Note that since $a^\nu(\theta_1) = 0$, $a(\theta_1) > 0$, and $t^\nu(\theta_1) = t(\theta_1) = \theta_1$, we have $T(a^\nu(\theta_1)) = T(a(\theta_1))$, which implies that

$$\phi^\nu(a^\nu(\theta_1); \theta_1) > \phi^\nu(a^\nu(\theta_1); \theta_1).$$

Now suppose $\phi^\nu(a^\nu(\theta_i); \theta_i) > \phi^\nu(a^\nu(\theta_i); \theta_i)$ and $T^\nu(a^\nu(\theta_i)) \geq T(a(\theta_i))$ for some $\theta_i > \theta_1$. If types $\theta_i$ and $\theta_{i+1}$ are pooled and $a^\nu(\theta_{i+1}) = a^\nu(\theta_i)$, then $\phi^\nu(a^\nu(\theta_{i+1}); \theta_{i+1}) > \phi^\nu(a^\nu(\theta_{i+1}); \theta_{i+1})$ and $T^\nu(a^\nu(\theta_{i+1})) \geq T(a(\theta_{i+1}))$ trivially hold. We thus focus on the case where $a^\nu(\theta_{i+1}) > a^\nu(\theta_i)$, so that types $\theta_i$ and $\theta_{i+1}$ are separated. In this case, $(a^\nu(\theta_i), t^\nu(\theta_i))$ and $(a^\nu(\theta_{i+1}), t^\nu(\theta_{i+1}))$ are connected by the equilibrium indifference curve of type $\theta_i$, whereas global incentive compatibility requires that

$$\phi^\nu(a; \theta_i) \geq T(a),$$

for all $a \in A$. Therefore, $T^\nu(\cdot)$ and $T(\cdot)$ cannot cross each other for $a \in [a^\nu(\theta_{i}), a^\nu(\theta_{i+1})]$. For $\theta_{i+1} \in [\theta_1, \theta^\nu] \cap \Theta^\nu$, $T^\nu(a^\nu(\theta_{i+1})) = T(a(\theta_{i+1})) = \theta_{i+1}$, which ensures

$$\phi^\nu(a^\nu(\theta_{i+1}); \theta_{i+1}) > \phi^\nu(a^\nu(\theta_{i+1}); \theta_{i+1}).$$

(6)

For $\theta_{i+1} \in [\hat{\theta}, \theta^\nu_s] \cap \Theta^\nu$, type $\theta_{i+1}$ is pairwise matched with type $p(\theta_{i+1})$, and any allocation must be on $D(\cdot; \theta_{i+1}, p(\theta_{i+1}))$. Provided that $T^\nu(\cdot)$ and $T(\cdot)$ cannot cross, (6) and $T^\nu(a^\nu(\theta_{i+1})) \geq T(a(\theta_{i+1}))$ must hold.
The preceding argument proves by induction that $\phi^v(a^v(\theta_i); \theta_i) > \phi^v(a^v(\theta_i); \theta_i)$ for all $\theta_i \in [\theta_1, \theta^*] \cap \Theta^v$, which in turn implies $T^v(a) > T(a)$ for all $a \in A^v$. We now build on this fact to construct a non-LSHPP equilibrium. To this end, note that our algorithm ensures $a(\theta_v) = D(\theta_v; \theta_v, \theta_v)$ and $t(\theta_v) = \theta_v$ (see Appendix B of Chen et al. (2022)), so that $T(a)$ intersects with $D(t; \theta_v, \theta_v)$ at $(D(\theta_v; \theta_v, \theta_v), \theta_v)$ in the $(a, t)$-space. Since $T^v(a) > T(a)$ for all $a \in A^v$, this means that $a^v(\theta^*_v) < D(\theta_v; \theta_v, \theta_v)$ and $t^v(\theta^*_v) < \theta_v$. We can then always find $\theta_{j_1}$ and $\theta_{j_2}$ in $(t^v(\theta^*_v), \theta_v)$ such that

$$u(a^v(\theta_i), t(\theta_i), \theta_{i-1}) = u(a^v(\theta_{i-1}), t(\theta_{i-1}), \theta_{i-1}),$$

for $i = j_1, j_2$, both to the left of $D(\cdot; \theta_v, \theta_v)$. Since types below $\theta^*_v$ have no incentive to mimic those types, the double-crossing property ensures that types above $p(\theta^*_v)$ have no incentive to mimic either.
References


In this appendix, we provide pseudo-codes to illustrate how the “main algorithm” and the “adjustment procedure” described in the paper work to produce an equilibrium allocation given model primitives. These codes are relatively sketchy. They are not meant to be used as a real program to compute an equilibrium, but are intended to supplement the description of the text to clarify the flow of the algorithm under various contingencies.

We first introduce some auxiliary functions used by the codes:

```plaintext
Define function type(q);
/* Returns the type of agent q: if q is in [F[i], F[i+1)), then the agent is of type i. This is the “iota” function in the text. */

Define function utility(a, t, i);
/* Returns utility level of type i given action a and reputation level t. */

Define function indiffC(t, u0, i);
/* Indifference curve: returns an action a at which utility(a, t, i) = u0. This is the “phi” function in the text. */

Define function D(t, i, j);
/* Dividing line: returns an action a satisfying the condition that the marginal rates of substitution of type i and type j are equal at the point (a, t). */

Define function average(q, r, n);
/* Returns the expected type when agents in [q, r) pool with type-n agent. This is the “mu” function in Section 5.1. */

Define function separating(i_min, i_max, a0, t0);
/* If i_min >= t0, returns an I-dimensional array [a_1, a_2, ..., a_I] of actions taken by the I types in a least-cost separating solution. The initial condition a_j for j=i_min is determined by utility(a_j, j, j) = utility(a0, t0, j). For i_min < j <= i_max, the separating actions are determined recursively by utility(a_j, j, j-1) = utility(a_{j-1}, j-1, j-1). For j < i_min or j > i_max, set a_j to a large number N.
  If i_min < t0, returns an I-dimensional array of large numbers [N, N, ..., N]. */

Define function intersectionUD(n, i_qn, u0, i_u);
/* Returns reputation level t at which the dividing line between type n and type i_qn intersects with an indifference curve of type i_u that corresponds to a utility level of u0. That is, the t that satisfies indiffC(t, u0, i_u) = D(t, n, i_qn). */
```
The main program “FindEquilibrium” calls the function “allocationTypeN” recursively and append the allocation generated by this function to the output until the program terminates.

Algorithm FindEquilibrium(I, F)

Input: I (the number of types: the type space is \{1, 2, ..., I\}) and F (an array with I+1 elements \{0, f1, f2, ..., 1\} that records the partition of types).

Output: q_eqm, a_eqm, t_eqm (arrays that stores equilibrium allocation: agents between q_eqm[i] and q_eqm[i+1] play a_eqm[i] and get reputation t_eqm[i]), and existence (existence equals 1 if an equilibrium is found and -1 if not. Proposition 4 shows that the value of existence is 1).

begin
  declare global arrays q_k, r_k, a_k, and t_k;
  existence = allocationTypeN(I, 0, 0, 1)
  /* As function “allocationTypeN" is called recursively, q_k, r_k, a_k, and t_k will be filled with the allocations in each round. As shown in Proposition 4, existence = 1 as the program terminates.
   In the following, q_k, r_k, a_k, and t_k are transformed to equilibrium allocations q_eqm, a_eqm, and t_eqm, showing that agents between q_eqm[i] and q_eqm[i+1] play a_eqm[i] and get reputation t_eqm[i]. */
  let nstar be the size of array q_k;
  for i := 1 to nstar
    /* Allocations for types below n^star. */
    if (i = 1 and q_k[i] > 0) or (i > 1 and q_k[i] > r_k[i-1])
      /* This corresponds to the case when there are separating allocations in round i. */
      imax := type(q_k[i]);
      if F[imax] = q_k[i]
        /* This takes care of the case when q_k[i] is in B. */
        imax := imax - 1;
      end
      if i=1
        imin := 1;
        sstar := separating(imin, imax, 0, 1);
      else
        imin := type(r_k[i-1]);
        sstar := separating(imin, imax, a_k[i-1], t_k[i-1]);
      end
      append \{sstar[imin], sstar[imin+1], ..., sstar[imax]\} to a_temp;
      append \{imin, imin+1, ..., imax\} to t_temp;
      append \{F[imin], F[imin+1], ..., F[imax]\} to q_temp;
    end
    /* pooling allocation in round i. */
    append a_k[i] to a_temp;
    append t_k[i] to t_temp;
    append q_k[i] to q_temp;
  end
  for i := nstar down to 1
    /* Allocations for types above nstar. */
    append a_k[i] to a_temp;
    append t_k[i] to t_temp;
    append F[I-i+1] to q_temp;
end
let m be the size of q_temp;
j := 1;
for i := 1 to m
/* Combine adjacent types with the same allocations. */
    if i = 1 or t_temp[i] != t_temp[i-1]
        a_eqm[j] := a_temp[i];
t_eqm[j] := t_temp[i];
qu_eqm[j] := q_temp[i];
j := j+1;
end
end
end

Here is the main function “allocationTypeN”.

*/ The main function begins here: */

function allocationTypeN(n, r, a0, t0)
/* This function finds the agents to pair with type-n agents (agents in [F[n],
    F[n+1]]) and their equilibrium allocation, given that allocations to agents in [r,
    F[n+1]) remain to be determined, and the allocation to type-(n+1) agent is (a0,
t0).
    If a well-defined equilibrium is found, it returns 1 and stores a, t, q, r in
each round in arrays a_k, t_k, q_k, and r_k; otherwise, it returns -1. */
begin
    i_r := type(r);
    /* i_r is the type of agent r. */

    /* Construct the candidate set. The candidate set will be stored in arrays
candidate_t, candidate_q, and candidate_a. Theoretically, the three arrays should
be continuous, but in a code, they can only be discrete. In this pseudocode, t and
q are discretized with intervals dt and dq, respectively. */
    if D(t0, n, i_r) <= a0
        /* If the dividing line between type n and type i_r is on the left of (a0,t0),
        the only possible allocation for type n is (a0, t0). */
        append t0 to candidate_t;
        append r to candidate_q;
        append a0 to candidate_a;
    else
        /* Otherwise, the allocation for type n is higher than (a0, t0), and we
        construct the candidate set as follows. */
        sstar := separating(i_r, n, a0, t0);
i_q := i_r - 1;
        while i_q+1 < n and sstar[i_q+1] <= D(i_q+1, n, i_q+1)
            i_q := i_q + 1;
        end
        if i_q = n - 1
            /* This corresponds to case (i) in Section 5.2, and a well-defined
            equilibrium is found. */
            q_k[I-n+1] := F[n];
\[ r_{k[n+1]} := F[n+1]; \]
\[ t_{k[n+1]} := n; \]
\[ a_{k[n+1]} := sstar[i_q+1]; \]
\[ \text{return 1; } \]

else

/* This corresponds to case (ii) in Section 5.2. Collect all the elements by decreasing t and q sequentially as shown in Figure 2. */

\[ q := \max(F[i_q+1], r); \]
if \( sstar[i_q+1] < D(i_q+1, n, i_q) \)

\[ u0 := \text{utility}(sstar[i_q], i_q, i_q); \]
\[ t_{\text{start}} := i_q + 1; \]
\[ t_{\text{end}} := \text{intersectionUD}(n, i_q, u0, i_q); \]
\[ t := t_{\text{start}} \]
while \( t < t_{\text{end}} \)

append \( t \) to \( \text{candidate}_t \);
append \( q \) to \( \text{candidate}_q \);
append \( \text{indiffC}(t, u0, i_q) \) to \( \text{candidate}_a \);
increase \( t \) by \( dt \).
end

\[ \text{for } i := i_q \text{ down to } i_r+1 \]
\[ t := \text{intersectionUD}(n, i, \text{utility}(sstar[i], i, i), i); \]
\[ a := D(t, n, i); \]
\[ q := F[i+1]; \]
while \( q > F[i] \)

append \( t \) to \( \text{candidate}_t \);
append \( q \) to \( \text{candidate}_q \);
append \( a \) to \( \text{candidate}_a \);
decrease \( q \) by \( dq \).
end
\[ t_{\text{start}} := t; \]
\[ t_{\text{end}} := \text{intersectionUD}(n, i, \text{utility}(sstar[i], i, i-1), i-1); \]
while \( t < t_{\text{end}} \)

append \( t \) to \( \text{candidate}_t \);
append \( q \) to \( \text{candidate}_q \);
append \( D(t, n, i) \) to \( \text{candidate}_a \);
increase \( t \) by \( dt \).
end
\[ t_{\text{start}} := t_{\text{end}}; \]
\[ u0 := \text{utility}(sstar[i-1], i-1, i-1); \]
\[ t_{\text{end}} := \text{intersectionUD}(n, i-1, u0, i-1); \]
while \( t < t_{\text{end}} \)

append \( t \) to \( \text{candidate}_t \);
append \( q \) to \( \text{candidate}_q \);
append \( \text{indiffC}(t, u0, i-1) \) to \( \text{candidate}_a \);
increase \( t \) by \( dt \).
end

end

/* Below is the last part of the candidate set for \( i = i_r \). */

\[ u0 := \text{utility}(a0, t0, i_r); \]
\[ t := \text{intersectionUD}(n, i_r, u0, i_r); \]
\[ a := D(t, n, i_r); \]
while \( q > r \)

append \( t \) to \( \text{candidate}_t \);
append \( q \) to \( \text{candidate}_q \);
append \( a \) to \( \text{candidate}_a \);
decrease \( q \) by \( dq \);
if \( r > F[i_r] \)
/* If \( r \) is not in \( B \), there is only one last element to be added to the candidate set. */
append \( t \) to candidate_t;
append \( r \) to candidate_q;
append \( D(t, n, i_r) \) to candidate_a;
else if \( r = F[i_r] \)
/* If \( r \) is in \( B \), there is a continuum of elements to be added. */
if \( i_r = 1 \)
    \( t_{end} := I; \)
else
    \( u0 := \text{utility}(a0, t0, i_r-1); \)
    \( t_{end} := \text{intersectionUD}(n, i_r, u0, i_r-1); \)
end
while \( t < t_{end} \)
    append \( t \) to candidate_t;
    append \( q \) to candidate_q;
    append \( D(t, n, i_r) \) to candidate_a;
    increase \( t \) by \( dt \);
end
if \( i_r > 1 \)
    \( t_{start} := t_{end}; \)
    \( t_{end} := \text{intersectionUD}(n, i_r-1, u0, i_r-1); \)
    while \( t < t_{end} \)
        append \( t \) to candidate_t;
        append \( q \) to candidate_q;
        append \( \text{indiffC}(t, u0, i_r-1) \) to candidate_a;
        increase \( t \) by \( dt \);
    end
end
end
end

/* Find the minimum allocation: remove elements before the minimum allocation from the candidate set. The remaining set is called the eligible candidate set. */
while average(q,t,n) > t where (t,q) is the first element in (candidate_t,candidate_q)
    remove the first (t,q,a) from (candidate_t,candidate_q,candidate_a);
end
feasible := -1;
/* feasible = 1 when an equilibrium is found. */

/* Adjustment procedure: construct the equilibrium path recursively by going through the candidate set until a well-defined equilibrium is found. */
/* Suppose that in round \( k'+1 \) (when \( n=I-k' \)), \( C(k'+1) \) is undefined, which means that the eligible candidate set (the candidate set after removing all the elements before the minimum allocation) is empty. Then the following outer while loop does not start, function allocationTypeN for round \( k'+1 \) just returns -1, and the procedure goes back to round \( k' \). If the eligible candidate set only contains one element, then the while loop in round \( k' \) also ends, and the procedure goes back to the previous round. This process continues until it reaches a round with other elements in the remaining eligible candidate set, which is \( k^\wedge \). The following outlines the three cases of the "adjustment procedure" on pages 25-26. Case 1: While the procedure goes through elements in round \( k^\wedge \), the subsequent rounds are initiated by executing ‘Feasible := allocationTypeN(n-1, r_n, a, t)’.
If \( r_n \) is in \( B \) in some round \( k^+m' \), then there are multiple elements in the eligible candidate set. The while loop will automatically go through all the elements in the eligible candidate set in round \( k^+m' \), performing like resetting \( k^+ \) to \( k^+m' \).

Case 2: After the procedure exhausts all the elements in round \( k^+ \), the while loop ends, and function allocationTypeN for round \( k^+ \) returns -1. The procedure keeps going back to the previous round until it reaches a round with other elements in the remaining eligible candidate set, and \( k^+ \) is reset to this round.

Case 3: As the adjustment procedure continues, eligible candidate set will arise in round \( k'+1 \). Then the following outer while loop starts for round \( k'+1 \), and allocations for subsequent new rounds will be constructed. */

```plaintext
while feasible = -1 and (candidate_t, candidate_q, candidate_a) is not empty
    let (t, q, a) be the first element in (candidate_t, candidate_q, candidate_a);
    r_n := q;
    while average(q, r_n, n) > t
        /* Find \( r_n \), which is the minimum value satisfying average(q, r_n, n) = t. */
        increase r_n by dt;
    end
    if r_n = F[n]
        /* If \( r_n = F[n] \), a well-defined equilibrium is found. */
        Feasible := 1;
    else
        /* In this case, \( r_n < F[n] \). More rounds are necessary to construct the whole equilibrium path. Go to the next round to see if an equilibrium can be found. */
        Feasible := allocationTypeN(n-1, r_n, a, t);
        if feasible = -1
            remove (t, q, a) from (candidate_t, candidate_q, candidate_a);
        end
    end
end
if feasible = 1
    q_k[I-n+1] := q;
    r_k[I-n+1] := r;
    a_k[I-n+1] := a
    t_k[I-n+1] := t;
    /* \( q, a, \) and \( t \) are the first element in candidate_q, candidate_a, and candidate_t, respectively. */
end
return feasible;
end
```