Asset pricing with short-selling constraints and many belief types: Three fast solution algorithms

Michael Hatcher*, September 2025

Abstract

Short-selling is common in financial markets but is also strictly regulated. When short selling is *banned*, heterogeneous beliefs determine which investors take long positions and which have constrained positions of *zero* in equilibrium. Solving such models is computationally intensive. We set out *three* algorithms suited to solving models with very large numbers of investor types – such as millions – quickly on a standard laptop or desktop computer. The fastest algorithm combines price iterations with a divide and conquer approach. As an application we study the impact of a short-selling ban on price dynamics and wealth distribution in a market of many investor types in evolutionary competition, and we observe that both can be affected substantially.

1 Introduction

Short selling is widespread in financial markets but is widely regulated by policymakers. Whereas a long position can be thought of as a bet that asset prices will increase, short-selling – a negative position – allows investors to make bets on falling asset prices.[1] It has been argued that such bets exacerbate financial market volatility. A common policy response has been to restrict or ban short selling: many countries introduced short-selling bans after sharp declines in asset prices in the 2007-9 Financial Crisis, and similar bans were reintroduced in some European economies during the 2011-12 sovereign debt crisis and the Covid-19 outbreak (Siciliano and Ventoruzzo, 2020). It is thus important that researchers be able to solve asset pricing models with short-selling constraints in an efficient manner.

In this paper we present and compare *three* algorithms designed to speed up the solution of heterogeneous-belief asset pricing models with short-selling constraints. We focus in particular on the case of *very large* numbers of heterogeneous beliefs – such as millions – which describes well the population of investors in real-world asset markets, such as stock

*Department of Economics, University of Southampton, m.c.hatcher@soton.ac.uk.

[1] When investors take a 'short' position, they borrow and immediately sell a financial asset before repurchasing and returning the asset to the lender, thereby closing their position.

exchanges, but comes with the problem of *computational bottlenecks* due to the high dimension of the beliefs vector: there are many asset demands and thus many cases to check in order to find a solution. Our main contribution is to show how price iterations and divide and conquer can be used to improve computation times with very large numbers of beliefs.

Our analysis uses the heterogeneous-beliefs asset pricing model (Brock and Hommes, 1998) that allows arbitrarily many behavioural belief types whose population shares may be determined endogenously by *evolutionary competition*; note that several similar dynamic models are essentially nested in this framework.[2] Our main departure relative to these models is to introduce short-selling constraints that stop investors from taking *negative* positions in the risky asset. When there are many investors with heterogeneous beliefs, this results in a problem that is computationally demanding and therefore time-consuming to solve.

The model with many belief types has been studied by Brock et al. (2005), who permit short-selling by investors. When investors face short-selling constraints – as in our version of the model – the market-clearing price and demands depend on *belief dispersion* across types. Since demand functions are *piecewise-linear*, the market-clearing price depends on how many types are short-selling constrained, but the number of constrained types in turn depends on the price – a difficult fixed point problem. For a market with a large number of investor types, it is *computationally intensive* to solve for the price and demands, and only recently have analytical results been provided which ease the process (see Hatcher, 2024b).

In this paper, we investigate how to best exploit these analytical results for computation in models with very large numbers of heterogeneous beliefs. Tasks include simulating time series of asset prices, the numerical investigation of price attractors, and the study of wealth distribution across investor types. We consider a short-selling *ban* and study the performance of three different solution algorithms which are well suited to solving models with many beliefs types, such as millions. Of these algorithms, one is a benchmark algorithm, while the other two algorithms are novel and utilize a divide-and-conquer approach which is aimed at improving solution times. We also consider the robustness of these algorithms, that is, 'out of the box' performance when their parameters are not optimized for a particular problem.

1.1 Related literature

We are not the first to consider asset pricing with short-selling constraints in the Brock and Hommes (1997) modelling framework. In an early paper, Anufriev and Tuinstra (2013) consider a short-selling cost (or tax) and present an algorithm where beliefs are *sorted* by optimism: they consider both the analytical and numerical solution of models with a *small* number of belief types who face short-selling costs. In such a model, a short-selling *ban* – as studied in this paper – arises when the short-selling tax increases without bound, making negative positions prohibitively costly. Hatcher (2024a) presents analytical results and an improved algorithm for that problem, which makes numerical analysis with large numbers

[2]For example, see LeBaron et al. (1999), Westerhoff (2004),Panchenko et al. (2013) or Hatcher and Hellmann (2025). For surveys of the literature, see Hommes (2006) or Hatcher and Hellmann (2024).

of belief types somewhat tractable. However, since the number of cases to be checked is quadratic in the number of belief types (after sorting by optimism), modelling very large numbers of investor types such hundreds of thousands or millions remains problematic.

These problems are avoided in the case of a short-selling *ban* (as in this paper). This benchmark case has some empirical realism given that policymakers have often responded to falling asset prices by banning short-selling; in addition, there are cases of markets such as Hong Kong and Korea where short-selling has been banned as a standard practice. For the case of a short-selling ban, Hatcher (2024b) derives analytical results and shows that sorting belief types by optimism implies that the number of cases increases *linearly* with the number of types, thereby making this case somewhat more computationally tractable.

A short-selling ban has also been studied by Dercole and Radi (2020), albeit that they study a few belief types and do not present a solution algorithm, instead considering stability and price volatility and finding some support for an uptick rule that bans short-selling. Our paper is part of a growing literature studying heterogeneous beliefs, asset prices and regulatory policies in financial markets (see Westerhoff, 2016). It is known that differences in beliefs combined with short-selling constraints can lead to price bubbles (see e.g. Scheinkman and Xiong, 2003), but such regulations could also aid market stability as noted.

1.2 Findings and robustness

Our analysis of solution algorithms indicates that the benchmark algorithm relies heavily on a good initial guess on the number of short-selling constrained types. As a result, the benchmark algorithm is relatively fast to find a solution if algorithm parameters have been tuned to improve performance, but otherwise it does quite poorly, implying weak 'out of the box' performance. Both of the divide and conquer algorithms do somewhat better on this score, especially the classic divide and conquer algorithm that uses binary search.

Hence, an important finding is the *robustness* of a classic divide and conquer approach that uses binary search: this algorithm is consistently fast and requires minimal inputs to be specified by the user. On the other hand, if tuning of parameters is allowed, the fastest algorithm among the three we study is a modified benchmark algorithm that combines 'price iterations' that yield an initial guess on the number of constrained types with a 'mini' divide and conquer approach that allows backtracking. Solution times are improved relative to the benchmark algorithm in Hatcher (2024b) when the parameters of both algorithms are optimized, and the improvements in computation time are non-trivial for very long simulations or the construction of attractors (which requires many different simulations). As expected, reductions in computation time are larger the larger the number of belief types.

Our policy application finds that a short-selling ban may reduce price volatility on the downside, by preventing very large drops in price below the fundamental price; however, a short-selling ban can also result in *over*-pricing. At the same time, a short-selling ban reduces wealth inequality across investor types. Although our results are shown in a particular type of asset pricing model, we discuss generalizability below to help guide the reader.

2 Model

Consider a finite set of myopic, risk-averse investor types $\mathcal{H} = \{h_1, ..., h_H\}$. At every discrete date $t \geq 1$, each type $h \in \mathcal{H}$ chooses a portfolio of a risky asset $z_{t,h}$ and a riskless bond with return $\tilde{r} > 0$ to maximize *mean-variance* utility over future wealth, given a risk-aversion parameter $a > 0$. The risky asset has current price p_t, future price p_{t+1}, and it pays stochastic dividends d_{t+1}, which are exogenous. Investors form subjective expectations of the future price and future dividends of the risky asset, as shown below. The underlying model follows Brock and Hommes (1998), except that the risky asset is in positive net supply $\overline{Z} > 0$ and short-selling ($z_{t,h} < 0$) is ruled out by non-negativity constraints: $z_{t,h} \geq 0$ for all t and h.

2.1 Asset demand

We denote the subjective expectation of type h at date t by $\tilde{E}_{t,h}[.]$, and the subjective variance is $\tilde{V}_{t,h}[.]$. The portfolio choice of type $h \in \mathcal{H}$ at date t solves the problem:[3]

$$\max_{z_{t,h}} \tilde{E}_{t,h}[w_{t+1,h}] - \frac{a}{2}\tilde{V}_{t,h}[w_{t+1,h}] \qquad \text{s.t.} \qquad z_{t,h} \geq 0 \tag{1}$$

where $w_{t+1,h} = (p_{t+1} + d_{t+1})z_{t,h} + (1+\tilde{r})(w_{t,h} - p_t z_{t,h})$ is future wealth, $w_{t,h} - p_t z_{t,h}$ is holdings of the risk-free asset, and $\tilde{V}_{t,h}[w_{t+1,h}] = \sigma^2 z_{t,h}^2$, with $\sigma^2 > 0$ and weight $a/2 > 0$.

Mean-variance utility as in (1) has theoretical underpinnings in the form of quadratic utility or an assumption that investors subjectively perceive returns as normally-distributed. Such an assumption is a common and tractable simplification in the literature.

Given short-selling constraints, the date t demand of each investor type is:

$$z_{t,h} = \begin{cases} \dfrac{\tilde{E}_{t,h}\left[p_{t+1}\right] + \tilde{E}_{t,h}\left[d_{t+1}\right] - (1+\tilde{r})p_t}{a\sigma^2} & \text{if } p_t \leq \dfrac{\tilde{E}_{t,h}\left[p_{t+1}\right] + \tilde{E}_{t,h}\left[d_{t+1}\right]}{1+\tilde{r}} \\ 0 & \text{if } p_t > \dfrac{\tilde{E}_{t,h}\left[p_{t+1}\right] + \tilde{E}_{t,h}\left[d_{t+1}\right]}{1+\tilde{r}} \end{cases}. \tag{2}$$

If the price p_t is small enough, type h's short-selling constraint is slack and their demand decreases with the price; see Brock and Hommes (1998), where short-selling constraints are absent. However, if the price is high enough to make the expected excess return of type h *negative*, the short-selling constraint will bind on type h and their position will be *zero* (though they would prefer to short-sell). Dividends follow $d_t = \overline{d} + \epsilon_t$, where $\overline{d} > 0$ and ϵ_t is IID mean zero with fixed variance. We assume $\tilde{E}_{t,h}\left[d_{t+1}\right] = \overline{d}$ for all t and h (common dividend beliefs); there is no loss of generality as this assumption can easily be relaxed.[4]

Equation (2) and common expected dividends imply that the short-selling constraint is more likely to bind on type h the more *pessimistic* is their price expectation $\tilde{E}_{t,h}\left[p_{t+1}\right]$. Hence, an investor's position in the risky asset will depend on their optimism and the level

[3]Further details of the portfolio problem are provided in the *Supplementary Appendix* of Hatcher (2024b).

[4]See the definition of $f_{t,h}$ in (4), which potentially permits $\tilde{E}_{t,h}\left[d_{t+1}\right]$ to vary over time and across types.

of the market price as compared to their 'participation price' $\frac{\tilde{E}_{t,h}[p_{t+1}]+\tilde{E}_{t,h}[d_{t+1}]}{1+\tilde{r}}$; see (2).

2.2 Price beliefs

We consider generic price beliefs which are *boundedly-rational* and can depend linearly on the current price p_t via a common coefficient $\overline{c}$ which is assumed to be non-negative.

Assumption 1 *All price beliefs are of the form:*

$$\tilde{E}_{t,h}[p_{t+1}] = \overline{c}p_t + \tilde{f}_{t,h} \tag{3}$$

where $\overline{c} \in [0, 1+\tilde{r})$ and $\tilde{f}_{t,h} \in \mathbb{R}$ is a generic forecast that cannot depend on current price p_t.

Assumption 1 allows a wide variety of boundedly-rational beliefs. The coefficient $\overline{c}$ allows linear dependence of price expectations on the current price; for example, investors may extrapolate on top of the current price or use the information of the current price with some weight (see e.g. LeBaron et al., 1999; Barberis et al., 2018). We assume $\overline{c} \geq 0$ to allow the case of no dependence on the current price (i.e. $\overline{c} = 0$) and we assume $\overline{c} < 1+\tilde{r}$ to ensure that individual demands are *decreasing* in the current price p_t; for reference, see (4) below.

The forecast $\tilde{f}_{t,h}$ (which can differ across types and over time) permits a potentially non-linear response to *past* prices, such as non-linear trend-following rules. Also, $\tilde{f}_{t,h}$ may contain type-specific 'fixed effects', be subject to random disturbances, or be influenced by social networks as in Panchenko et al. (2013) or Hatcher and Hellmann (2025). Assumption 1 in Brock and Hommes (1998) is nested by (3) when $\overline{c} = 0$ and $\tilde{f}_{t,h} = E_t[p^*_{t+1}] + g_h(x_{t-1}, ..., x_{t-L_h})$, where $g_h : \mathbb{R}^{L_h} \to \mathbb{R}$ is a function that can differ across types, L_h is the lag of belief type h, and $x_t := p_t - p^*_t$ is the price deviation from the fundamental asset price p^*_t.

For convenience, let $f_{t,h} := \tilde{f}_{t,h} + \tilde{E}_{t,h}[d_{t+1}] - a\sigma^2\overline{Z}$ and $r := \tilde{r} - \overline{c}$. Given $\tilde{E}_{t,h}[d_{t+1}] = \overline{d}$, we have $f_{t,h} = \tilde{f}_{t,h} + \overline{d} - a\sigma^2\overline{Z}$ and the demands in (2) can be written as follows:

$$z_{t,h} = \begin{cases} \dfrac{f_{t,h} - (1+r)p_t + a\sigma^2\overline{Z}}{a\sigma^2} & \text{if } p_t \leq \dfrac{f_{t,h} + a\sigma^2\overline{Z}}{1+r} \\ 0 & \text{if } p_t > \dfrac{f_{t,h} + a\sigma^2\overline{Z}}{1+r}. \end{cases} \tag{4}$$

Writing demands in terms of $f_{t,h}$ as in (4) is convenient because the latter does *not* depend on the current price p_t and allows the addition of the term $a\sigma^2\overline{Z}$ in the numerator, which simplifies the algebra of the price solution. As noted, this specification allows expected dividends to potentially differ across types and over time (since $\tilde{E}_{t,h}[d_{t+1}]$ is subsumed in $f_{t,h}$) and it also allows price beliefs to depend (linearly) on the current price via the parameter $r := \tilde{r} - \overline{c}$. Writing demands as in (4) is also consistent with a 'deviation from fundamentals' representation of investors' demands, as commonly used in the literature.[5]

[5]Given our assumptions, the fundamental price is $p^*_t = \overline{p} := (\overline{d} - a\sigma^2\overline{Z})/\tilde{r}$, so $f_{t,h} = \tilde{f}_{t,h} + \tilde{r}\overline{p}$ (see (4)). Thus, $f_{t,h} - (1+r)p_t = \hat{E}_{t,h}[x_{t+1}] - (1+r)x_t$, where $r = \tilde{r} - \overline{c}$, $x_t := p_t - \overline{p}$, and $\hat{E}_{t,h}[x_{t+1}] := \tilde{f}_{t,h} - (1-\overline{c})\overline{p}$.

2.3 Population shares

We allow the population shares $n_{t,h}$ of investor types to be endogenous and time-varying, but as in Hatcher (2024b) we rule out dependence on the current price p_t (Assumption 2).

Assumption 2 *We consider population shares of the form $n_{t,h} = \hat{n}_h(\mathbf{n}_{t-1}, \mathbf{u}_{t-1})$ where $\hat{n}_h$ is a real function such that $\sum_{h \in \mathcal{H}} n_{t,h} = 1$, $n_{t,h} \in (0,1)\ \forall t, h$ and $\mathbf{n}_{t-1}$ ($\mathbf{u}_{t-1}$) is the vector of past population shares (resp. past fitness levels). In particular, we rule out dependence of $n_{t,h}$ on the current asset price p_t (though not dependence on lagged prices p_{t-1}, p_{t-2} etc.).*

Assumption 2 is quite general. For example, the population shares may be given by *evolutionary competition*. Following Brock and Hommes (1997), a popular approach is a discrete choice logistic model $n_{t+1,h} = \frac{\exp(\beta U_{t,h})}{\sum_{h \in \mathcal{H}} \exp(\beta U_{t,h})}$, where the intensity of choice $\beta \in [0, \infty)$ determines how fast agents switch to better-performing predictors. Various fitness measures $U_{t,h}$ are used in the literature, including realized profits (Brock and Hommes, 1998) and forecast accuracy (e.g. Ap Gwilym, 2010). Assumption 2 rules out the 'extreme' population shares of 0 or 1, but it is straightforward to relax this assumption; see Hatcher (2024b).

Having past fitness levels $\mathbf{u}_{t-1}$ in the function $\hat{n}_h$ allows evolutionary competition mechanisms as above, while the inclusion of past population shares $\mathbf{n}_{t-1}$ allows for asynchronous updating (see e.g. Hommes, 2013, Ch. 5). It would not impose any extra burden to allow the function $\hat{n}_h$ to be time-varying or to include additional endogenous variables within the vector $\mathbf{u}_{t-1}$; however, as stated in Assumption 2, we rule out dependence of population shares on the *current* price p_t (or future values of the price). Fixed population shares $n_{t,h} = 1/H$ is relevant for agent-based or social network models where types are *individuals*, while exogenous time-varying shares are used in herding models (see Kirman, 1991, 1993).

2.4 Benchmark results: price and demands

The asset market clears when $\sum_{h \in \mathcal{H}} n_{t,h} z_{t,h} = \overline{Z}$ subject to (4) and Assumptions 1 and 2. Given positive outside supply $\overline{Z} > 0$, there is a unique market-clearing price p_t and we have the following benchmark result that plays an important role in our solution algorithms.

Proposition 1 (Hatcher (2024b)) *Let p_t be the market-clearing price at date t, let $n_{t,h}$ be the population share of type h at date t, and let $\mathcal{B}_t \subseteq \mathcal{H}$ ($\mathcal{S}_t := \mathcal{H} \setminus \mathcal{B}_t$) be the set of unconstrained types (constrained types) at date t.*

(i) If $\sum_{h \in \mathcal{H}} n_{t,h}(f_{t,h} - \min_{h \in \mathcal{H}}\{f_{t,h}\}) \leq a\sigma^2 \overline{Z}$, then no type is short-selling constrained at date t ($\mathcal{B}_t^* = \mathcal{H}$, $\mathcal{S}_t^* = \emptyset$) and the market-clearing price is

$$p_t = \frac{\sum_{h \in \mathcal{H}} n_{t,h} f_{t,h}}{1+r} := p_t^* \tag{5}$$

with demands $z_{t,h} = (a\sigma^2)^{-1}(f_{t,h} + a\sigma^2 \overline{Z} - (1+r)p_t) \geq 0\ \forall h \in \mathcal{H}$.

(ii) If $\sum_{h\in\mathcal{H}} n_{t,h}(f_{t,h} - \min_{h\in\mathcal{H}}\{f_{t,h}\}) > a\sigma^2\overline{Z}$*, at least one type is short-selling constrained and* $\exists$ *unique non-empty sets* $\mathcal{B}_t^* \subset \mathcal{H}$, $\mathcal{S}_t^*$ *such that* $\sum_{h\in\mathcal{B}_t^*} n_{t,h}(f_{t,h} - \min_{h\in\mathcal{B}_t^*}\{f_{t,h}\}) \leq a\sigma^2\overline{Z} < \sum_{h\in\mathcal{B}_t^*} n_{t,h}(f_{t,h} - \max_{h\in\mathcal{S}_t^*}\{f_{t,h}\})$*, and the price and demands are given by*

$$p_t = \frac{\sum_{h\in\mathcal{B}_t^*} n_{t,h} f_{t,h} - (1 - \sum_{h\in\mathcal{B}_t^*} n_{t,h}) a\sigma^2\overline{Z}}{(1+r)\sum_{h\in\mathcal{B}_t^*} n_{t,h}} > p_t^* \tag{6}$$

and $z_{t,h} = (a\sigma^2)^{-1}(f_{t,h} + a\sigma^2\overline{Z} - (1+r)p_t) \geq 0 \ \forall h \in \mathcal{B}_t^*,\ z_{t,h} = 0 \ \forall h \in \mathcal{S}_t^*$.

Proposition 1 gives the market-clearing price and demands for an arbitrarily large set of belief types whose population shares may be endogenously determined. Since the results apply at any date $t \in \mathbb{N}_+$, we can find a solution for $t = 1, 2, ...$, starting from period 1. If *no* types are short-selling constrained, the asset price depends on *all price beliefs* as in (5); however, when short-selling constraints *bind*, only the beliefs of the unconstrained types matter for price determination; see (6). Intuitively, the market price depends on the demands (hence beliefs) of 'buyers' who participate in the market by taking a long position.

Part (i) of Proposition 1 gives a condition that determines whether short-selling constraints are slack for all types. If *belief dispersion*, $\sum_{h\in\mathcal{H}} n_{t,h}(f_{t,h} - \min_{h\in\mathcal{H}}\{f_{t,h}\})$, is small enough, all constraints are slack and the price is p_t^* in (5): the standard solution absent short-selling constraints (Brock and Hommes, 1998). Short-selling constraints will bind for one or more types *belief dispersion* is large enough; see Proposition 1 Part (ii). Then, at least one type – and at most $H-1$ types – will be short-selling constrained at the market-clearing price. Intuitively, those types with relatively low valuations of the asset (low $f_{t,h}$) will want to short sell if the price is 'biased' upwards relative to their valuation by more optimistic types; hence the 'pessimists' will be short-selling constrained at the equilibrium price. The sets of unconstrained and short-selling constrained types $\mathcal{B}_t^*$, $\mathcal{S}_t^*$ are determined by 'cut-off' conditions that depend on belief dispersion (see directly above (6)). Including the case where short-selling constraints are slack for all types, there are $2^H - 1$ candidates for $\mathcal{B}_t^*$, $\mathcal{S}_t^*$; hence, finding a solution is computationally intensive for large numbers of belief types H.

We now give a simple two-type example, following Hatcher (2024b).

Example 1 *Consider two types* h_1, h_2 *with beliefs* f_{t,h_1}, f_{t,h_2} *that satisfy Assumption 1 with* $f_{t,h} = \tilde{f}_{t,h} + \overline{d} - a\sigma^2\overline{Z}$*, as in (4), for* $h = h_1, h_2$*. The population shares of each type* n_{t,h_1} *and* $n_{t,h_2} = 1 - n_{t,h_1}$*. By Proposition 1, if* $\sum_{h\in\{h_1,h_2\}} n_{t,h}(f_{t,h} - \min\{f_{t,h_1}, f_{t,h_2}\}) \leq a\sigma^2\overline{Z}$ *neither type is short-selling constrained, and* $p_t = \frac{1}{1+r}\sum_{h\in\{h_1,h_2\}} n_{t,h} f_{t,h}$ *by (5), where* $r = \tilde{r} - \overline{c}$*. If the above condition is not met, then either* $f_{t,h_1} - f_{t,h_2} > a\sigma^2\overline{Z}/n_{t,h_1}$ *(*h_2 *is short-selling constrained) or* $f_{t,h_2} - f_{t,h_1} > a\sigma^2\overline{Z}/n_{t,h_2}$ *(*h_1 *is short-selling constrained). In the former case,* $\mathcal{B}_t^* = \{h_1\}$, $\mathcal{S}_t^* = \{h_2\}$*, and* $p_t = [(1+r)n_{t,h_1}]^{-1}(n_{t,h_1} f_{t,h_1} - (1 - n_{t,h_1})a\sigma^2\overline{Z})$*, with demands* $z_{t,h_1} = \overline{Z}/n_{t,h_1}$, $z_{t,h_2} = 0$ *by (6). In the latter case,* $\mathcal{B}_t^* = \{h_2\}$, $\mathcal{S}_t^* = \{h_1\}$*, so* $p_t = [(1+r)n_{t,h_2}]^{-1}(n_{t,h_2} f_{t,h_2} - (1 - n_{t,h_2})a\sigma^2\overline{Z})$ *and* $z_{t,h_1} = 0$, $z_{t,h_2} = \overline{Z}/n_{t,h_2}$ *by (6).*

Example 1 is a simple case nested by Proposition 1. It shows that ordering belief types in terms of optimism at each t allows some case(s) to be *excluded* as solutions for the price and demands (e.g. the short-selling constraint cannot bind on type h_1 if they are the more optimistic type, i.e. if $f_{t,h_1} > f_{t,h_2}$). Based on this observation, Anufriev and Tuinstra (2013) suggest a solution algorithm that uses sorting. Hatcher (2024b) shows how this approach can be combined with analytical results to produce a fast algorithm with H cases to be checked, in contrast to the exponential $2^H - 1$ when types are unordered.

To follow this approach, consider an adjusted set of types $\tilde{\mathcal{H}}_t = \{1, ..., \tilde{H}_t\}$ with the property that the most optimistic type(s) in the set of investors $\mathcal{H}$ receive label $\tilde{H}_t$, the next most optimistic type(s) gets label $\tilde{H}_t - 1$, and so on, down to the least optimistic type(s) with label 1. Types with equal optimism get the *same* label, so $\tilde{H}_t \leq H$ (i.e. $|\tilde{\mathcal{H}}_t| \leq |\mathcal{H}|$). In the case of ties in terms of optimism, the period t population share of the 'group' is the sum of the population shares of the individual types. We first present a corollary that uses the set $\tilde{\mathcal{H}}_t$ to simplify the task of finding a solution; all our solution algorithms use this result.

Corollary 1 (Hatcher (2024b)) *Let $\tilde{\mathcal{H}}_t = \{1, ..., \tilde{H}_t\}$ be the set defined above, such that beliefs are ordered as $\tilde{E}_{t,1}[p_{t+1}] < \tilde{E}_{t,2}[p_{t+1}] < ... < \tilde{E}_{t,\tilde{H}_t}[p_{t+1}]$, or equivalently $f_{t,1} < f_{t,2} < ... < f_{t,\tilde{H}_t}$. Let $disp_{t,k} := \sum_{h=k+1}^{\tilde{H}_t} n_{t,h}(f_{t,h} - f_{t,k})$, where $k \in \{1, ..., \tilde{H}_t - 1\}$. Then:*

$$p_t = \begin{cases} \dfrac{\sum_{h=1}^{\tilde{H}_t} n_{t,h} f_{t,h}}{1+r} := p_t^* & \text{if } disp_{t,1} \leq a\sigma^2 \overline{Z} \\ \dfrac{\sum_{h=2}^{\tilde{H}_t} n_{t,h} f_{t,h} - n_{t,1} a\sigma^2 \overline{Z}}{(1-n_{t,1})(1+r)} := p_t^{(1)} & \text{if } disp_{t,2} \leq a\sigma^2 \overline{Z} < disp_{t,1} \\ \dfrac{\sum_{h=3}^{\tilde{H}_t} n_{t,h} f_{t,h} - (n_{t,1}+n_{t,2}) a\sigma^2 \overline{Z}}{(1-n_{t,1}-n_{t,2})(1+r)} := p_t^{(2)} & \text{if } disp_{t,3} \leq a\sigma^2 \overline{Z} < disp_{t,2} \\ \quad \vdots & \quad \vdots \\ \dfrac{n_{t,\tilde{H}_t} f_{t,\tilde{H}_t} - (\sum_{h=1}^{\tilde{H}_t - 1} n_{t,h}) a\sigma^2 \overline{Z}}{(1-\sum_{h=1}^{\tilde{H}_t - 1} n_{t,h})(1+r)} := p_t^{(\tilde{H}_t - 1)} & \text{if } disp_{t,\tilde{H}_t - 1} > a\sigma^2 \overline{Z} \end{cases} \tag{7}$$

where $p_t^{(k^*)}$ is the equilibrium price when types $1, ..., k^*$ are short-selling constrained, p_t^* is the price if short-selling constraints were absent, and $p_t^* < p_t^{(k)}$, $p_t^{(k)} > p_t^{(k-1)}$ for all $k \leq k^*$.

Corollary 1 streamlines the task of finding the market-clearing price. In Proposition 1, where beliefs are unordered, there are $2^H - 1$ cases, compared to only $\tilde{H}_t \leq H$ when beliefs are ordered as in Corollary 1. Clearly, this amounts to a substantial reduction in computational burden in models with a large number of belief types H. For example, with only 20 distinct beliefs (types) at date t, there are $2^{20} - 1 \approx 1.05$ million candidates for the sets $\mathcal{B}_t^*, \mathcal{S}_t^*$ when types are not ordered, but only 20 candidates for these sets and the market-clearing price when types are ordered by optimism, i.e. Corollary 1 for the case of $\tilde{H}_t = 20$ types.

The final part of Corollary 1 says the market price when one or more short-selling constraints are binding is higher than the (hypothetical) price p_t^* if short-selling constraints are absent, and that the price is smaller the fewer short-selling constraints k are assumed to be binding. These properties are useful to obtain an initial guess on the number of types

k^* that are short-selling constrained, and an iterative procedure which uses the property $p_t^{(k)} > p_t^{(k-1)}$ can be used to obtain an updated guess (price iterations), as discussed below.

2.5 Discussion: Generalizability

Before turning to our solution algorithms, we first take stock and clarify the reach of our methods. Thus far, we have set out a model in which short-selling is prohibited by a non-negativity constraint, such that investors cannot indulge a desire to take a negative position. We have made several specific assumptions which are essential and cannot be generalized (such as the absence of rational expectations), but also some others that can be relaxed.

One important restriction is that of linear demand, which results from the assumption of mean-variance utility. It is essential that (unconstrained) demand be linear in the market-clearing price as in (2) and (4), since this ensures a unique price solution and allows price beliefs that depend linearly on the current price p_t. Hence, our algorithms do require some specific assumptions on utility and beliefs. A second key assumption is that investors can take unrestricted positions in the risk-free asset. If this assumption were relaxed and investors faced a leverage constraint as in in't Veld (2016), then analytical results for an arbitrary number of types would no longer be tractable, which is central for our algorithm.

On the other hand, there are several assumptions that can be relaxed without little or no difficulty. We have already mentioned that dividend expectations may differ across types and over time. We have maintained the assumption of a fixed subjective (return) variance of investors (equal to σ^2), but this assumption can be relaxed to allow time-varying and potentially different variances (or risk aversion) across investor types. Similarly, investors can be allowed to have different weights on the current price in their price beliefs.[6]

Finally, it is straightforward to allow short-selling bans across multiple asset markets with endogenous participation in each as in Westerhoff (2004); in this case, we just need to solve for the price and demands in each market rather than in one but the solution procedure is the same. In a similar vein, allowing a short-selling ban that is conditional on a falling price or triggered by drops in price that exceed some threshold is straightforward since this requires only minor changes to the "if" statements in Proposition 1 and Corollary 1.[7]

It is worth emphasizing that although the number of cases after sorting is linear in the number of types – see Corollary 1 – this will still impose a formidable computational hurdle for extremely large numbers of types. Hence, one limitation even of an algorithm that uses sorting is that it will not be practical to tackle all problems, although very large numbers of beliefs can be considered without computation times becoming too large. Our comparison of three algorithms below reports computation times as the number of types is increased.

[6]For more detail on these two extensions, see Hatcher (2024b, Sec. 5 and Supp. Appendix). The key point is that these extensions only require the user to change the price equations and the 'cutoff conditions' that determine the set of constrained types and the set of unconstrained types: the process of the searching for sets that meet those conditions (using our algorithms) is essentially unchanged.

[7]The case of multiple asset markets with short-selling bans is discussed in Section 5.1 of Hatcher (2024b) and a conditional short-selling ban in Section 3.3. See also the Supplementary Appendix of that paper.

3 Algorithms

We now present our three algorithms and discuss their main similarities and differences.

3.1 Benchmark algorithm with price iterations

We first consider a benchmark algorithm from Hatcher (2024b) that draws on the analytical results presented above in Corollary 1 and uses a 'price iterations' approach. Although the algorithm itself is not new, its performance and the best use of price iterations has not been examined in previous work, and some speed improvements are implemented here.[8]

Algorithm 1

1. Construct the set $\tilde{\mathcal{H}}_t$ by ordering beliefs as $f_{t,1} < f_{t,2} < ... < f_{t,\tilde{H}_t}$ and find the associated population shares $n_{t,h}$ of types $h = 1, ..., \tilde{H}_t$.

2. Compute $disp_{t,1} = \sum_{h=2}^{\tilde{H}_t} n_{t,h}(f_{t,h} - f_{t,1})$. If $disp_{t,1} \leq a\sigma^2\overline{Z}$, accept $p_t = p_t^*$ as the date t price, where p_t^* is the price if short-selling constraints were absent (see Corollary 1), compute demands and proceed to period $t+1$. Otherwise, move to Step 3.

3. Set $p_t^{guess} = p_t^*$ and find the largest k such that $z_{t,k}^{guess} = \frac{f_{t,k} + a\sigma^2\overline{Z} - (1+r)p_t^{guess}}{a\sigma^2} < 0$, and denote this value $\underline{k}$. Starting from $k = \underline{k}$, check if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$; if not, use *price iterations* to update the guess $\underline{k}$.[9] Starting from initial value $k = \underline{k}$, set $k = k_{prev} + 1$ until a k^* is found such that $disp_{t,k^*+1} \leq a\sigma^2\overline{Z} < disp_{t,k^*}$.

4. Accept k^* as the number of short-selling constrained types, such that the price is $p_t = p_t^{(k^*)} := \frac{\sum_{h=k^*+1}^{\tilde{H}_t} n_{t,h}f_{t,h} - \left[\sum_{h=1}^{k^*} n_{t,h}\right] a\sigma^2\overline{Z}}{(1+r)\sum_{h=k^*+1}^{\tilde{H}_t} n_{t,h}}$, compute demands and go to period $t+1$.

Algorithm 1 has three desirable properties. First, if the condition on belief dispersion in Step 2 is met, it is known right away that all short-selling constraints are slack at date t and no computation time is wasted checking cases where one of more types have binding short-selling constraints. Second, if the condition in Step 2 is not met, then using the unconstrained solution p_t^* as a guess yields a lower bound $\underline{k} > 0$ on the number of short-selling constrained types k^*. Note that $\underline{k}$ is a lower bound for k^* since $p_t^{(k)} > p_t^*$ for all $k \leq k^*$ (see Corollary 1); that is, binding short-selling constraints *raise* the asset price relative to the counterfactual of no short-sale constraints. Therefore, if types $1, ..., k$ would like to short-sell at price p_t^*, they *must* also be short-selling constrained at price $p_t^{(k^*)} > p_t^*$ (see (4)), implying that $k^* \geq \underline{k}$. Third, since $\underline{k} \leq k^*$, the algorithm will converge to k^* when started from a valid initial guess.

[8] Code is vectorized where possible and we have endeavoured to eliminate any redundant computations.

[9] Given $\underline{k}$, update the price to $p_t^{guess} = \left(\sum_{h=\underline{k}+1}^{\tilde{H}_t} n_{t,h}f_{t,h} - \left[\sum_{h=1}^{\underline{k}} n_{t,h}\right] a\sigma^2\overline{Z}\right) / [(1+r)\sum_{h=\underline{k}+1}^{\tilde{H}_t} n_{t,h}]$ based on Corollary 1. Find the largest k such that $z_{t,k}^{guess} = \frac{f_{t,k} + a\sigma^2\overline{Z} - (1+r)p_t^{guess}}{a\sigma^2} < 0$, update $\underline{k}$ to this value, and update p_t^{guess}. Repeat for a fixed number of iterations or until there is no change in $\underline{k}$.

A visual representation of Algorithm 1 is given in Figure 1. Starting from the initial guess $\underline{k}$ on the number of short-selling constrained types, the algorithm checks progressively larger number of constrained types in a sequential manner, by adding 1 at each update, until a solution is found (Step 3). Such an approach is guaranteed to find a solution but is not optimized for speed: it could take many updates of the number of short-selling constrained types until the equilibrium number k^* is reached, especially when $\underline{k}$ is not a good initial guess for k^*, such that many updates are needed until a solution is found.

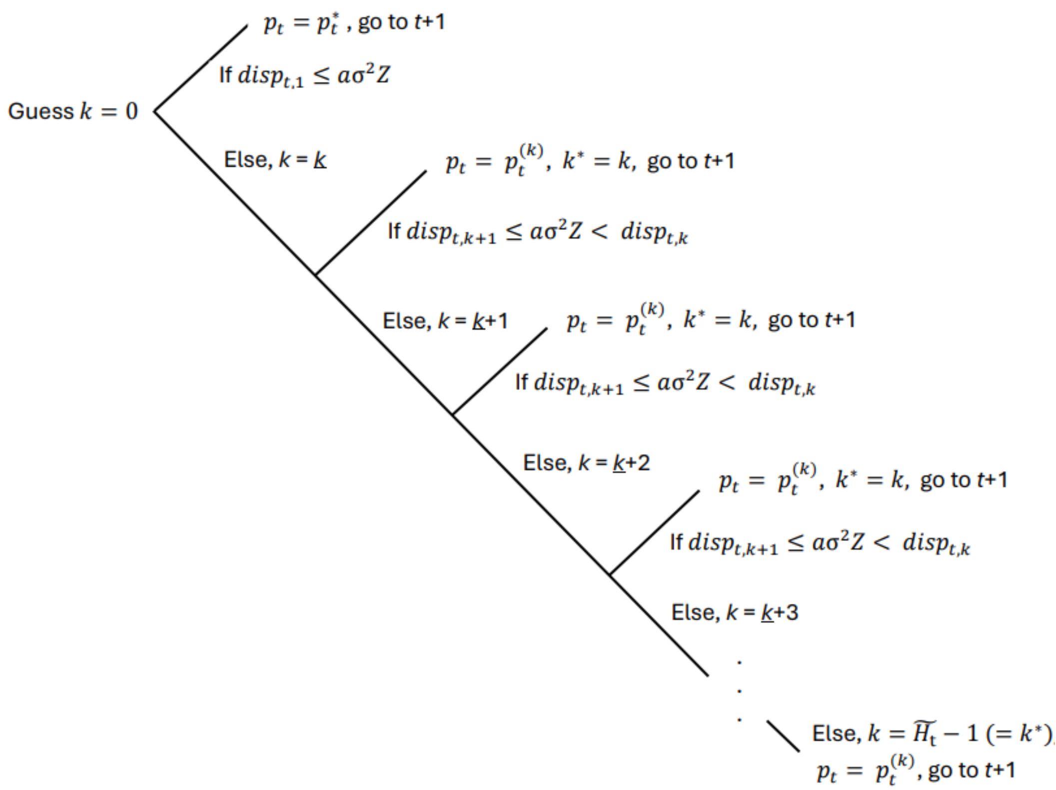

Figure 1: Sequential solution approach in Algorithm 1. The algorithm moves south-east as the number of short-selling constrained types, k, is increased and terminates when $k = k^*$.

To improve the initial guess $\underline{k}$, price iterations are used (Algorithm 1, Step 3). Specifically, we can repeatedly replace p_t^{guess} with an updated price $p_t^{(k)}$ based on the current guess $k = \underline{k}$ in Step 3 and then generate an updated value of k that equals the number of negative (unconstrained) demands at this new price. Here, we use $p_t^{(1)} < ... < p_t^{(k^*-1)} < p_t^{(k^*)}$ (Corollary 1) to find the number of constrained types k^* faster and improve solution times.[10] By varying the number of 'price iterations' systematically as below, we can ascertain whether many such iterations improve computation speed or whether several iterations are sufficient.

[10]Note that the iterations will not overshoot k^* because the guessed price will remain below the market-clearing price. If $k' = k_{prev}$ or if k^* is reached, the iterations are terminated early using a 'break' command.

3.2 Divide and conquer I

We now consider a variant of Algorithm 1 that adds an updating step which is inspired by the divide-and-conquer method. In particular, when searching for the number of short-selling constrained types k^*, a general step size $k_{step} > 1$ is used at each update rather than moving from one value k to the next k' in steps of 1 as in Algorithm 1. Without loss of generality, we assume k_{step} is an even number.[11] The algorithm follows Algorithm 1 in all other respects.

Algorithm 2

1. Implement Steps 1 and 2 of Algorithm 1 (see above); if $k^* = 0$ go straight to Step 4.

2. Set $p_t^{guess} = p_t^*$ and find the largest k such that $z_{t,k}^{guess} = \frac{f_{t,k}+a\sigma^2\overline{Z}-(1+r)p_t^{guess}}{a\sigma^2} < 0$, say $\underline{k}$. Starting from $k = \underline{k}$, check if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$; if not, use *price iterations* to update the guess $\underline{k}$.[12] Starting from $k = \underline{k}$, check if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$; if not, set $k = k_{prev} + k_{step}$ as the new guess, where $k_{step} > 1$ is an even number.

3. If $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$, then $k^* = k$ and go to $t + 1$. If $disp_{t,k}, disp_{t,k+1} > a\sigma^2\overline{Z}$, update k to $k = k_{prev}+k_{step}$ (undershoot);[13] else if $disp_{t,k}, disp_{t,k+1} \leq a\sigma^2\overline{Z}$ (overshoot), use backtracking and consider $k_{prev} + 2, k_{prev} + 4, \ldots, k_{prev} + k_{step} - 2$ in turn, and let $\tilde{k}_{step} = 2, 4, \ldots, k_{step} - 2$. At each $\tilde{k}_{step}$, if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$, then $k^* = k$ and go to period $t + 1$; else if $disp_{t,k} \leq a\sigma^2\overline{Z}$, then $k^* = k - 1$ and go to period $t + 1$; else update $\tilde{k}_{step}$ by 2 and repeat these steps until k^* is found and then go to $t + 1$.

4. Use k^* to compute price and demands and go to period $t + 1$ (Algorithm 1, Step 4).

Algorithm 2 starts in the same way as Algorithm 1 (see Step 1 above) but considers a different updating of the (guessed) number of short-selling constrained types k by employing a fixed even step size (Step 2), rather than an update of 1 as the baseline algorithm in Algorithm 1. The potential advantage of such an approach is speed: if k^* is somewhat larger than $\underline{k}$, then the equilibrium number of short-selling constrained types k^* will be found much faster using an approach that 'jumps' toward k^* rather than increasing k by 1 at each update.

With such a 'multi-step' approach there is the possibility of overshooting k^* and hence a backtracking option is needed. We use backtracking with a step size of 2 (Step 3), so Algorithm 2 is guaranteed to find k^*. In particular, if $\tilde{k}_{step}$ in Step 3 overshoots k^* (case $disp_{t,k} \leq a\sigma^2\overline{Z}$), then $k^* = k_{prev} + \tilde{k}_{step} - 1$; if there is no overshoot, either $k^* = k_{prev} + \tilde{k}_{step}$ or a further update of $\tilde{k}_{step}$ is needed and the same rules will apply at the new $\tilde{k}_{step}$. Once the number of short-selling constrained types k^* is found, the price and demands can be computed and we can then proceed to the next period $t + 1$ (Step 4, as in Algorithm 1).

[11] There is no loss of generality because if k_{step} is odd this simply implies a remainder of 1 after all pairs have been dealt with and hence a case that can be checked individually (i.e. on its own) as in Algorithm 1.

[12] See Footnote 8 for a description of how the price iterations part of the algorithm works.

[13] Note $disp_{t,k} := \sum_{h=k+1}^{\tilde{H}_t} n_{t,h}(f_{t,h} - f_{t,k})$ falls as k rises, so $disp_{t,k}, disp_{t,k+1} > a\sigma^2\overline{Z}$ if $disp_{t,k+1} > a\sigma^2\overline{Z}$.

The above description makes clear that Algorithm 2 differs from Algorithm 1 only in the step size that is used, i.e. k_{step} (see Steps 2 and 3). Like Algorithm 1, it will converge to a solution if implemented correctly. Intuitively, when k^* is much larger than $\underline{k}$, the algorithm will avoid checking many initial guesses on the number of constrained types one-by-one; instead, it will check only half such cases if $k_{step} = 2$, a quarter of such cases if $k_{step} = 4$, and so on. A trade-off arises, however, because as k_{step} is increased above 2, backtracking may be required, which carries its own computational cost that will need to be offset against any potential speed gains coming from initially moving faster towards k^* by 'ascending' several steps at each update rather than going 'step by step'. We investigate the trade-off below.

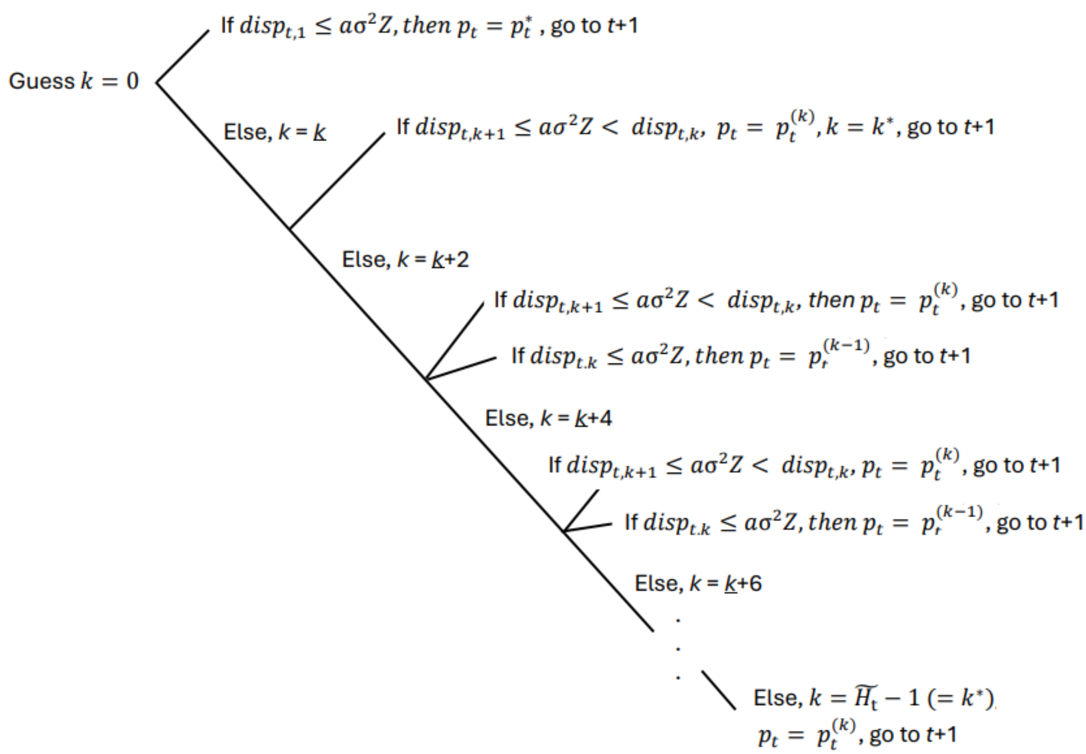

Figure 2: Divide and conquer I when $k_{step} = 2$. The algorithm moves south-east as the number of short-selling constrained types, k, is increased and terminates when $k = k^*$.

Algorithm 2 is illustrated in Figure 2 for $k_{step} = 2$, such that backtracking is not required, to ease the exposition. We see that Algorithm 2 uses fewer updating steps than Algorithm 1 and uses double 'if' statements at each update to check if $k = k^*$. Notice that the latter approach uses the fact that belief dispersion $disp_{t,k}$ must *decrease* as k is increased; therefore, if $k > k^*$ after an update of 2 from k $(< k^*)$, it follows that $k^* = k - 1$ and hence an explicit backtrack to $k - 1$ types is not needed, which should improve solution times.[14]

[14]Recall that $disp_{t,k} := \sum_{h=k+1}^{\tilde{H}_t} n_{t,h}(f_{t,h} - f_{t,k})$, where $k \in \{1, ..., \tilde{H}_t - 1\}$; see Corollary 1. Both the summand $n_{t,h}(f_{t,h} - f_{t,k})$ and the number of terms in the sum fall as k is increased. Therefore, if

3.3 Divide and conquer II

Lastly, we consider a classic divide-and-conquer approach that uses a binary search algorithm. As with Algorithm 2 above, this approach to solving the short-selling constraints problem has not been considered in the previous literature.[15] Divide and conquer algorithms break a larger problem into smaller subproblems which are each solved separately. One classic application of divide and conquer is using binary search to find an item in an ordered list by discarding half of the remaining items at each step. Since the time complexity of binary search is logarithmic, i.e. $O(\log n)$, it is an ideal search procedure when faced with a large number of sorted items; e.g. 20 probes is sufficient to search among a million such items.[16]

In the application at hand, the items are beliefs and we know where each item is in the sorted list because we have re-labelled investor beliefs according to their ranking in terms of optimism from least to most optimistic; see Corollary 1 and the preceding discussion. However, although we know how to find a belief with a particular position in the sorted list, we do not know whether this corresponds to the equilibrium number of short-selling constrained types k^*, until we check belief dispersion at this number of types to see if the inequality $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$ holds, as required; if not, we must try a different value of k, say k'. Hence, search is needed and the problem at hand is analogous to a standard problem solved using binary search, as shown in the following algorithm.

Algorithm 3

1. Implement Steps 1 and 2 of Algorithm 1 (see above); if $k^* = 0$ go straight to Step 4.

2. Set $p_t^{guess} = p_t^*$ and find the largest k such that $z_{t,k}^{guess} = \frac{f_{t,k}+a\sigma^2\overline{Z}-(1+r)p_t^{guess}}{a\sigma^2} < 0$, and denote this value $\underline{k}$. Starting from $k = \underline{k}$, check if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$; if not, set $k_l = \underline{k}$, $k_u = \tilde{H}_t - 1$ and $k = \text{round}\left(k_l + \frac{k_u-k_l}{2}\right)$ as initial values.[17]

3. If $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$, number of short-selling constrained types is $k^* = k$, go to Step 4; else if $disp_{t,k+1}, disp_{t,k} > a\sigma^2\overline{Z}$, hold k_l at $k_l = k_{prev}$ and update k to $k = \text{ceil}\left(k_l + \frac{k_u-k_l}{2}\right)$, and if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$, then $k^* = k$, go to Step 4; otherwise, hold k_u at $k_u = k_{prev}$ and update k to $k = \text{floor}\left(k_l + \frac{k_u-k_l}{2}\right)$, and if $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$, then $k^* = k$, go to Step 4. Repeat until k^* is found.

4. Use k^* to compute price and demands and go to period $t + 1$ (Algorithm 1, Step 4).

Algorithm 3 shows how binary search starts with the initial guess for the number of short-selling constrained types, $\underline{k}$, and an upper bound equal to the maximum possible number

$disp_{t,k+1}, disp_{t,k} > a\sigma^2\overline{Z}$ then k is too high, while if $disp_{t,k+1}, disp_{t,k} \leq a\sigma^2\overline{Z}$ then k is too low.

[15] See Anufriev and Tuinstra (2013), Hatcher (2024a) and Hatcher (2024b) for previous algorithms.

[16] For an introduction to divide and conquer and binary search see, for example, Louridas (2020).

[17] Here, round(.) returns the number in brackets to the nearest integer. Note $k_l + \frac{k_u-k_l}{2} = \frac{k_l+k_u}{2}$ but the latter should not be used due to the problem of overflow ($k_l + k_u$ may exceed the largest allowable number).

of short-selling constrained types, $\tilde{H}_t - 1$. Hence, we can construct an initial set of beliefs under consideration – i.e. $\{\underline{k}, \underline{k}+1, \ldots, \tilde{H}_t - 1\}$. From this set a new guess k is computed as an average of the minimum and maximum (Step 2). The updated value of k is checked to see if it is the equilibrium number of short-selling constrained types, k^* (in which case we can go to Step 4); if not, we check whether the current k undershoots or overshoots the equilibrium value, k^*, and we use that information in the next update (Step 3).[18] A visual representation of the search procedure used in Algorithm 3 is provided in Figure 3.

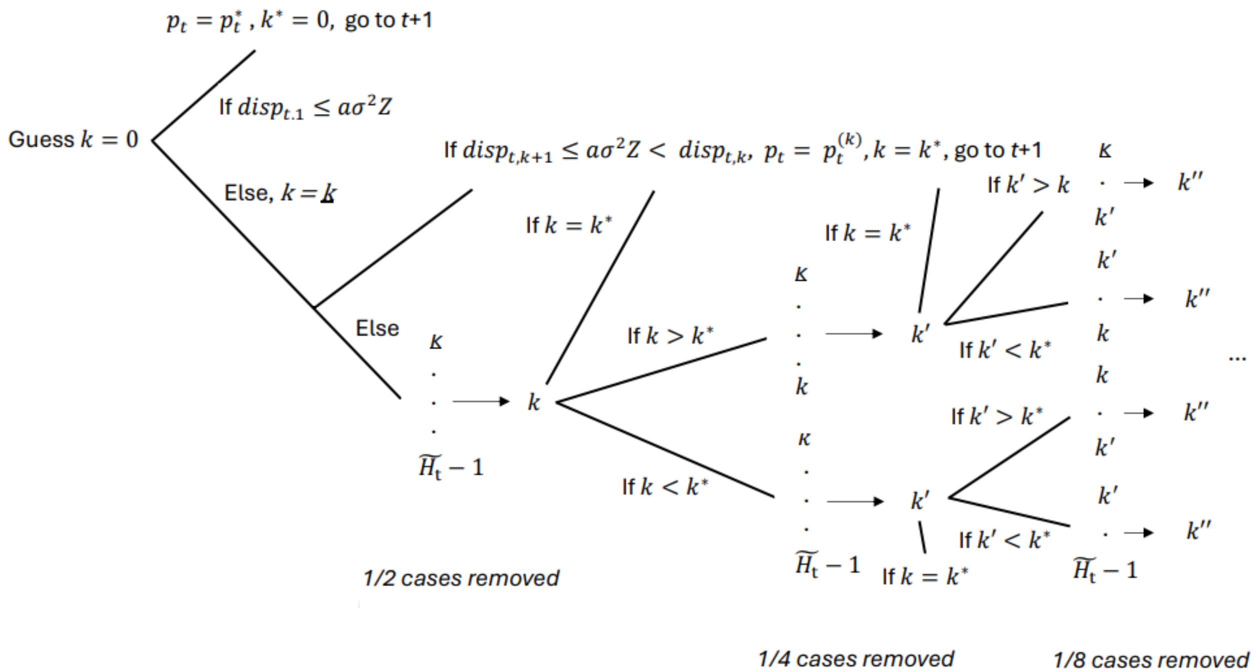

Figure 3: Algorithm 3: Divide and conquer II – binary search algorithm. The algorithm moves right as the number of short-selling constrained types k is updated. Half the remaining cases are eliminated at each step, and the algorithm terminates when $k = k^*$.

In the case of an undershoot, k is increased by taking the current value of k as the new minimum and keeping the maximum unchanged; similarly, in the case that k overshoots (i.e. $k > k^*$) the value of k is decreased by taking the current k as the new maximum and keeping the minimum unchanged. In this way, the algorithm discards 'useless' beliefs at each update and thereby 'zones in' on the equilibrium number of short-selling constrained types, k^*. The process of updating k used in Step 3 will move toward k^*, and this divide and conquer approach can be expected to find k^* quite efficiently even if the number of beliefs is very large (as noted, 20 probes are sufficient for binary search with a million items).

The visual depiction of Algorithm 3 – see Figure 3 – shows that half of the existing

[18] Recall that $disp_{t,k} := \sum_{h=k+1}^{\tilde{H}_t} n_{t,h}(f_{t,h} - f_{t,k})$ (where $k \in \{1, ..., \tilde{H}_t - 1\}$) decreases as k is increased.

cases are eliminated at each step by checking whether the current k is smaller or larger than the equilibrium number of short-selling constrained types, k^*. Recall that although k^* is not known until the algorithm terminates, the above comparison can nevertheless be made because $disp_{t,k}, disp_{t,k+1}$ are informative about whether the current guess on the number of short-selling constrained types, k, is above or below the equilibrium value, k^*. If $k = k^*$, then the algorithm is terminated using a 'break' command. The latter is tested after each update of k by checking whether $disp_{t,k+1} \leq a\sigma^2\overline{Z} < disp_{t,k}$: if so, then $k = k^*$; if not then k is updated and the problem is divided into further subproblems until k^* is found.

4 Example and Application

We now consider a numerical example. We use a version of the Brock and Hommes (1998) model with many types and a short-selling ban, such that investors who want to short-sell ($z_{t,h} < 0$) are prevented from doing so by non-negativity constraint: $z_{t,h} \geq 0$ for all t and h.

Demands of types $h \in \mathcal{H}$ are

$$z_{t,h} = \begin{cases} \frac{\tilde{E}_{t,h}\left[p_{t+1}\right] + \overline{d} - (1+\tilde{r})p_t}{a\sigma^2} & \text{if } p_t \leq \frac{\tilde{E}_{t,h}[p_{t+1}]+\overline{d}}{1+\tilde{r}} \\ 0 & \text{if } p_t > \frac{\tilde{E}_{t,h}[p_{t+1}]+\overline{d}}{1+\tilde{r}} \end{cases} \tag{8}$$

where we have assumed IID dividends $d_t = \overline{d} + \epsilon_t$ with $\tilde{E}_{t,h}[d_{t+1}] = \overline{d}\ \forall t, h$.

Following Brock and Hommes (1998), we consider linear predictors of the form:

$$\tilde{E}_{t,h}\left[p_{t+1}\right] = \overline{p} + b_h + g_h(p_{t-1} - \overline{p}), \qquad b_h \in \mathbb{R},\ g_h \geq 0. \tag{9}$$

Equation (10) is a standard specification in the literature. The intercept term consists of the fundamental price $\overline{p}$ plus 'bias' b_h in the price forecast of type h, whereas g_h is trend-following parameter of type h. Type h is a pure fundamentalist investor if $b_h = g_h = 0$, while larger values of g_h or $|b_h|$ imply, respectively, stronger trend-following and stronger forecast bias.

The fundamental price $\overline{p}$ is the unique fundamental solution under common rational expectations; see Brock and Hommes (1998). Given that the risky asset is in positive net supply $\overline{Z} > 0$, the fundamental price is $\overline{p} = (\overline{d} - a\sigma^2\overline{Z})/r$, where $r := \tilde{r}$ is the interest rate on the riskless asset. Writing (9) in deviations from the fundamental price, $x_t := p_t - \overline{p}$, gives:

$$\hat{E}_{t,h}\left[x_{t+1}\right] = b_h + g_h x_{t-1}, \qquad \text{where } \hat{E}_{t,h}\left[x_{t+1}\right] := \tilde{E}_{t,h}\left[p_{t+1}\right] - \overline{p}. \tag{10}$$

The demands in (8) can be written in terms of the price deviations x_t as

$$z_{t,h} = \begin{cases} \frac{\hat{E}_{t,h}\left[x_{t+1}\right] - (1+r)x_t + a\sigma^2\overline{Z}}{a\sigma^2} & \text{if } x_t \leq \frac{\hat{E}_{t,h}[x_{t+1}]+a\sigma^2\overline{Z}}{1+r} \\ 0 & \text{if } x_t > \frac{\hat{E}_{t,h}[x_{t+1}]+a\sigma^2\overline{Z}}{1+r}. \end{cases} \tag{11}$$

Fitness $U_{t,h}$ is a linear function of past profits net of predictor costs $C_h \geq 0$. Profits at date t are given by scaling demand $z_{t-1,h}$ by the realized excess return $R_t := p_t + d_t - (1+r)p_{t-1} = x_t - (1+r)x_{t-1} + a\sigma^2\overline{Z} + \epsilon_t$, where ϵ_t is the IID dividend shock, and we abstract from memory of past performance. For all $t \geq 1$ fitness and population shares are given by

$$U_{t,h} = R_t z_{t-1,h} - C_h, \qquad n_{t+1,h} = \frac{\exp(\beta U_{t,h})}{\sum_{h \in \mathcal{H}} \exp(\beta U_{t,h})}, \qquad \text{where } \beta \in [0, \infty). \tag{12}$$

The fitness levels $U_{t,h}$ determine the population shares $n_{t+1,h}$ of each type via a discrete-choice logistic model with intensity of choice β. The intensity of choice determines how fast agents switch to better-performing predictors. In the special case $\beta = 0$ no switching occurs; increasing β implies more switching to relatively profitable predictors. Following Brock and Hommes (1997, 1998), this *evolutionary competition* mechanism has been widely studied.

We use the same parameters as in Anufriev and Tuinstra (2013, Sec. 3.1): $\overline{Z} = 0.1$, $a\sigma^2 = 1$, $r = 0.1$, and we set $\overline{d} = 0.6$, giving a fundamental price $\overline{p} = \frac{\overline{d} - a\sigma^2\overline{Z}}{r} = 5$. Further, we assume half the types pay no cost and have a chartist-type predictor: $\hat{E}_{t,h}[x_{t+1}] = g_h x_{t-1}$ for $h = 1, \ldots, H/2$, where $g_h > 0$ is drawn from a uniform distribution with support $[1.05, 1.2]$. The remaining types $h = H/2+1, \ldots, H$ have a fundamental-type $\hat{E}_{t,h}[x_{t+1}] = b_h$, where b_h is drawn from a uniform distribution with support $[-0.1, 0.1]$, and predictor costs $C_h = 1 - |b_h|$ that depend on the 'closeness' of beliefs to a pure fundamentalist. Lastly, we set $\beta = 5$.

4.1 Baseline simulation

We first present a simulation of the asset price for an initial price $x_0 = 5$, $H = 100,000$ types, and no dividend shocks. We also track the number of short-selling constrained types in each period. The left panel of Figure 4 plots the price and the right panel the number of constrained types (those who prefer a negative position but are prevented from doing so); for comparison purposes we also plot the price when positions are unrestricted (No ban).

For the case of no ban, the asset price falls rapidly from the initial value of 5 and settles close to the fundamental price (grey line). By comparison, when a short-selling ban is enforced, the price initially increases and then oscillates before settling at a value where the risky asset is substantially overvalued relative to fundamentals (black line). In the first period $t = 1$, around 63,000 types are short-selling constrained (i.e. almost two-thirds) and this fraction initially increases sharply before reaching a peak and falling (right panel). Intuitively, given the relatively high initial price deviation of $x_0 = 5$, chartist types will tend to be more optimistic than fundamental types, such that the latter would like to short-sell along with some of the more pessimistic chartists (with relatively low trend-following parameters g_h) but are *unable* to do so. The initial increase in price reinforces the differences in beliefs and relative performance also plays a role through changes in the population shares of different types according to trading performance (evolutionary competition).

In Table I we report some results for the simulation shown in Figure 4. The simulation of

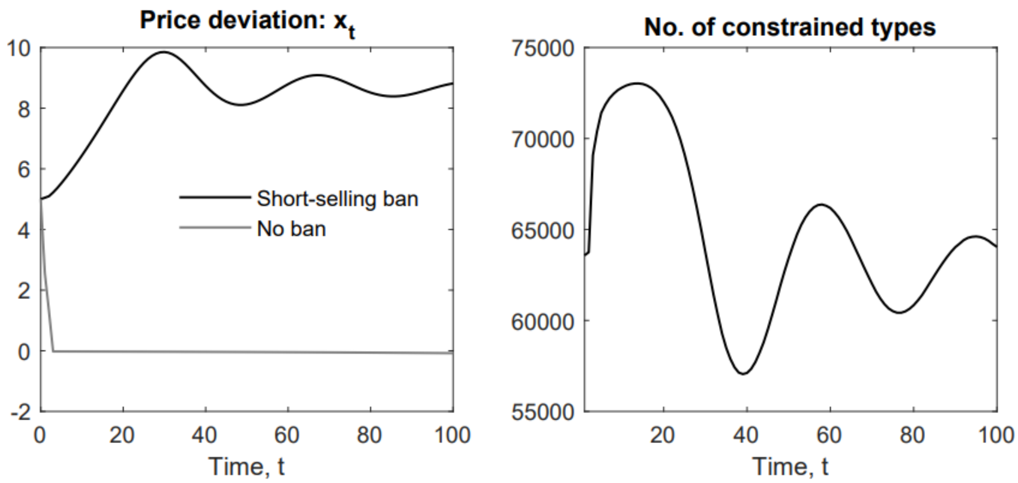

Figure 4: Asset price and number of short-selling constrained types $|\mathcal{S}_t^*|$ when $H = 100,000$: deterministic simulation of 100 periods. Price with no ban is shown for comparison (grey).

Table I: Baseline simulation: $H = 100,000$ types, $T = 100$ periods

Case	Time (s)	Bind freq.	$\max(\lvert\mathcal{S}_t\rvert), \min(\lvert\mathcal{S}_t\rvert)$	$\max(Error_t)$
No short-sell constraints	0.20	-	-	5.8e-16
Short-sell constraints	0.38	100%	73,055 57,006	5.2e-14

Notes: $\max(Error_t) := \max\{Error_1, ..., Error_T\}$, where we define the date t simulation error as $Error_t = \lvert \sum_{h\in\mathcal{H}} n_{t,h} z_{t,h} - \overline{Z}\rvert$. Demands $z_{t,h}$ depend on the computed market-clearing price.

100 periods took less than half a second (we used the baseline algorithm with price iterations here; we compare the different algorithms in the next section).[19] The maximum error across periods is essentially zero as expected and is comparable to that for the standard analytical solution $x_t = (1+r)^{-1} \sum_{h\in\mathcal{H}} n_{t,h} \hat{E}_{t,h} [x_{t+1}]$ for the case of unrestricted short-selling (see the final column); this gives us confidence that the correct solution is found when a short-selling ban is imposed.[20] As shown by Figure 4 (right panel), every period from $t = 1$ to $t = 100$ has some type(s) short-selling constrained (a bind frequency of 100%), hence preferring a negative position to zero. The maximum number of constrained types across all 100 periods was 73,055 in period 15, whereas the minimum of 57,006 was reached in period 40.

4.2 Comparing the algorithms

Table II reports simulation times for Algorithms 1–3 when the number of types H is increased from 100,000 up to 1 million, 5 million and 10 million. Dividend shocks $d_t = \overline{d} + \epsilon_t$ are *now*

[19] For the case of $H = 100,000$ types shown in Table I, the solution times are very similar for all three algorithms (around 0.4 seconds) and the computed solution is the same for all three algorithms.

[20] The small error seems to be a result of computation with a high-dimensional beliefs vector. For example, the error is not exactly zero even for the known analytical solution when short-selling constraints are absent.

introduced: ϵ_t is drawn from a truncated-normal distribution with mean zero, standard deviation $\sigma_d = 0.005$ and support $[-\overline{d}, \overline{d}]$. Simulations were run in MATLAB 2023a (Windows version) on a Dell OptiPlex Desktop with an Intel Core i5-13500 2.50 GHz processor and 16GB of RAM. We report results for three, five and seven price iterations (i.e. updates) to the initial guess $\underline{k}$ on the number of short-selling constrained types.[21] For Algorithm 2, the step size was set at $k_{step} = 2$ – i.e. double the unit step size in Algorithm 1.

Table II: Computation times in seconds (Baseline simulation, $T = 100$ periods)

		Three price iterations	
#Types H / Algorithm	1 (Benchmark)	2 (DivCon I, $k_{step} = 2$)	3 (DivCon II)
$100,000$	0.38	0.38	0.44
1 million	9.9	7.3	7.1
5 million	178.8	99.6	40.7
10 million	733.2	344.1	92.3
		Five price iterations	
#Types H / Algorithm	1 (Benchmark)	2 (DivCon I, $k_{step} = 2$)	3 (DivCon II)
$100,000$	0.41	0.40	0.47
1 million	5.1	5.0	8.0
5 million	28.1	28.0	46.0
10 million	60.4	60.2	103.2
		Seven price iterations	
#Types H / Algorithm	1 (Benchmark)	2 (DivCon I, $k_{step} = 2$)	3 (DivCon II)
$100,000$	0.40	0.40	0.60
1 million	5.1	5.1	8.0
5 million	28.1	28.1	46.4
10 million	60.6	60.8	104.1

Notes: The table reports simulation times in seconds for all three algorithms (short-selling ban) when using two, four or six price iterations (top, middle, bottom resp.) to generate initial guess $\underline{k}$.

For a relatively small number of belief types such as $H = 100,000$, all three algorithms have similar solution times of around 0.4 seconds (Table II, top rows), so there is little to choose between them. However, a *trade-off* arises when the number of belief types H is very large, such as several million. Specifically, if the number of price iterations is too low to get a good guess on the number of short-selling constrained types (top panel, 3 iterations), both Algorithms 1 and 2 perform poorly in terms of speed, while Algorithm 3 – classic divide and conquer – is much faster. Furthermore, the relative speed gain of using Algorithm 2 rather than the benchmark Algorithm 1 is substantial at more than 40% because three price iterations are insufficient to give a good initial guess $\underline{k}$ on the number of constrained types in each period, and Algorithm 2 avoids sticking to 'poor guesses' for a long while.

[21] The corresponding values of $\max(Error_t)$ across all 100 periods (omitted from Table II) are 5.1e-14 ($H = 100,000$), 2.7e-14 ($H = 1$ million), 3.3e-14 ($H = 5$ million), and 4.3e-14 ($H = 10$ million).

Increasing the number of iterations to five (middle panel) switches the ranking between the algorithms: both Algorithm 1 and Algorithm 2 are faster than Algorithm 3, with Algorithm 2 performing slightly better than Algorithm 1 because its divide and conquer approach means that fewer cases need to be checked in total and also that updating proceeds faster than one-by-one.[22] Intuitively, five iterations is enough to provide a good initial guess on the number of constrained types in this example, which greatly benefits Algorithm 1 and Algorithm 2 – which both rely on a good initial guess – while making little difference for the classic divide and conquer approach in Algorithm 3 (which becomes slightly slower).

Overall, Algorithm 3 (binary search) is quite robust in the sense of not requiring a good initial guess on the number of short-selling constrained types to produce relatively fast solution time. The fact that speed of Algorithm 3 deteriorates slightly as the number of price iterations is increased suggests this 'add-on' is not useful in this case. In stark contrast, Algorithms 1 and 2 rely on a good initial guess for fast computation times, and therefore need enough price iterations to have been specified, as seen in the middle and lower panel of Table II. Of course, what is 'enough' in terms of iterations will usually depend on the example or problem at hand, and a good guess is not assured. Therefore, a clear advantage of Algorithm 3 is its robustness and lack of reliance on a 'good' setting of user inputs.

The results in Table II clearly raise the question of how sensitive the relative performance of the algorithms is to choices made by the user, such as the number of price iterations and the step size k_{step} in Algorithm 2. We consider this issue in the next section in order to shed some light on how the speed of the three solution algorithms can be improved.

4.3 Speeding up the algorithms

We now consider how the performance of each algorithm can be improved. We build on the results in the previous section which show the number of price iterations is important for computation times, especially for Algorithm 1 and Algorithm 2. We also consider the role of other inputs, such as the 'step size' in Algorithm 2 (fixed at $k_{step} = 2$ thus far) and the best approach for an initial guess $\overline{k}$ on the number of constrained types in Algorithm 3.

The improvements we consider are as follows:

1. **Algorithm 1**: number of price iterations allowed to be any positive integer in a range
2. **Algorithm 2**: number of price iterations and step size k_{step} allowed to vary
3. **Algorithm 3**: number of price iterations allowed to vary, including allowing *zero* iterations and the case of *no guess* on the number of short-selling constrained types

[22]Recall that there are two considerations here: first, updating using a step size of $k_{step} = 2$ avoids the need for backtracking because if k 'overshoots' k^*, this implies that the current guess on the number of short-selling constrained types *minus one* will yield the equilibrium number of constrained types k^*; second, updating with $k_{step} = 2$ 'ascends' two steps at a time rather than one step at a time as in Algorithm 1.

4.3.1 Algorithm 1

We start with Algorithm 1 by plotting the solution time in the baseline simulation against the number of price iterations as described in Step 3 of the algorithm; see Figure 5.

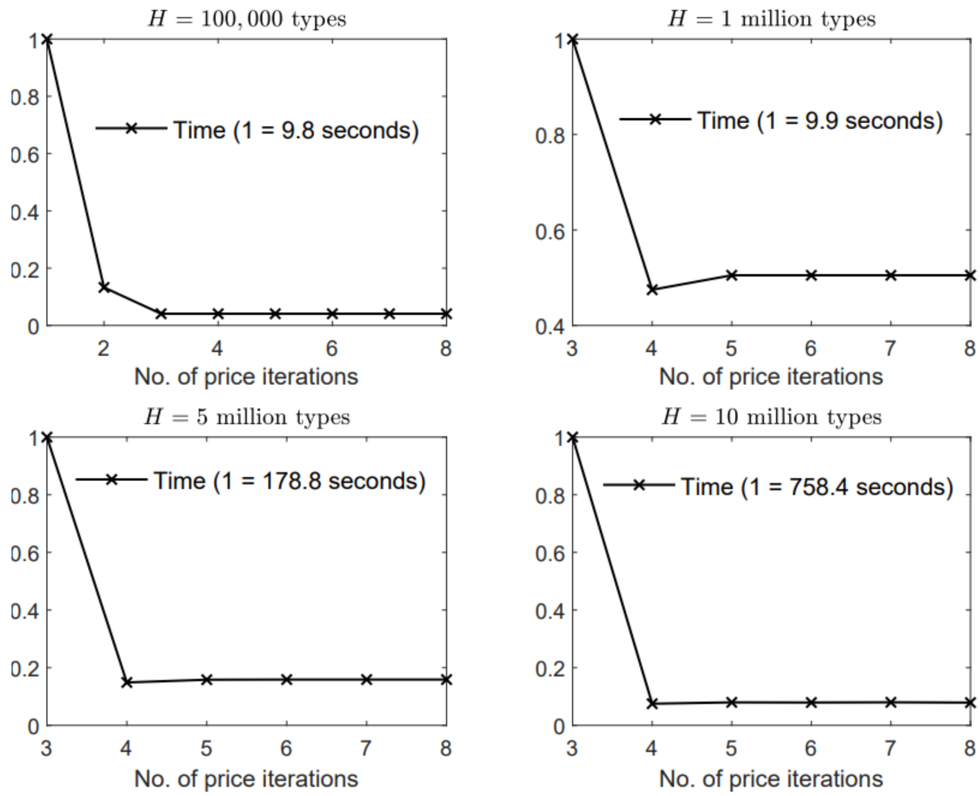

Figure 5: Algorithm 1 computation speed vs number of price iterations: various H

The key observation from Figure 5 is that computation time drops dramatically once a large enough number of price iterations is used in the algorithm. For a relatively small number of types $H = 100,000$ (top left), two price iterations are sufficient to secure almost all of the speed gain. For much larger numbers of types such as 1–10 million (other panels), three iterations are need to secure the speed gains and larger numbers of iterations beyond this do not yield any clear benefit. Note that the result that computation time is essentially unchanged at higher number of iterations make sense because (i) the updating of the price is not very computationally intensive, and (ii) price iterations are automatically terminated if there is no update on the value that was reached in the previous iteration.

4.3.2 Algorithm 2

We now consider Algorithm 2. Recall there are two important values set by the user in this case: the number of price iterations and the step size k_{step}, with the latter being a parameter in the divide and conquer part of the algorithm. We consider below the impact of price iterations and the changes in the step size separately.

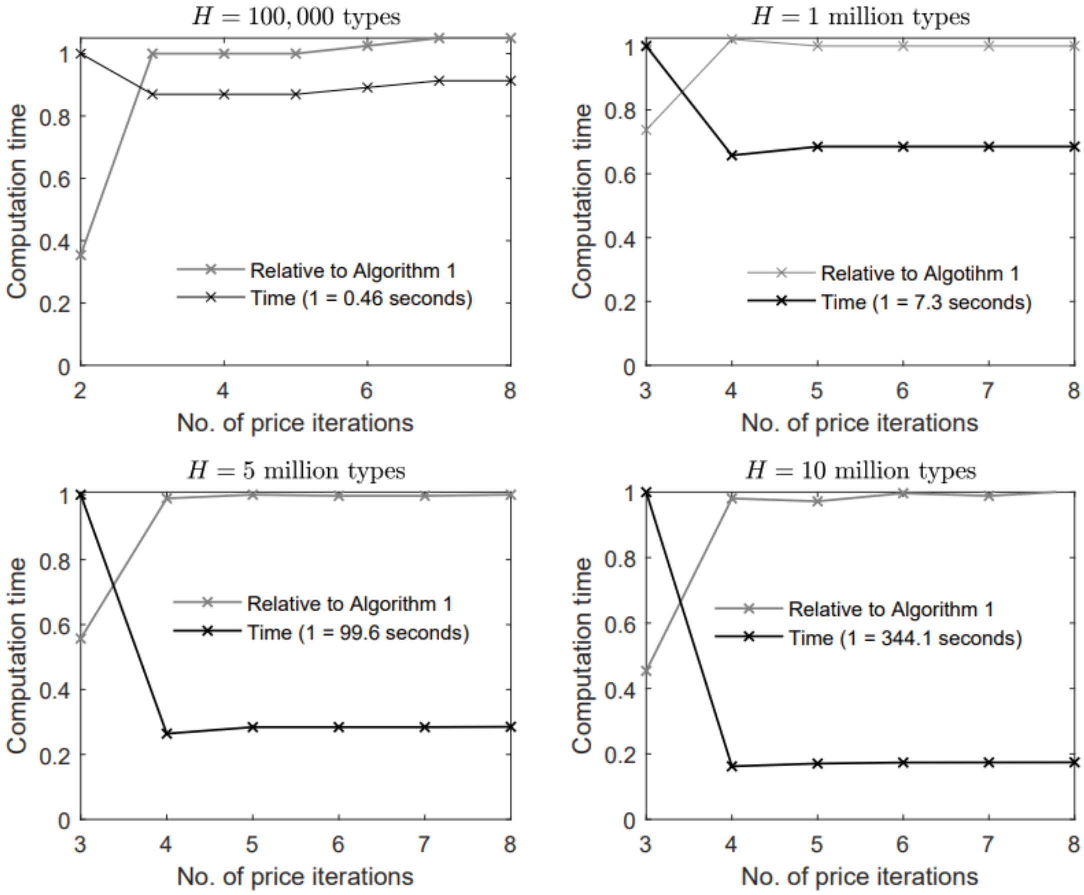

Figure 6: Algorithm 2 computation time vs number of price iterations: various H

In Figure 6 we show how increasing the number of price iterations affects computation times, and we report a relative measure of computation time that takes Algorithm 1 as the benchmark. As for Algorithm 1 we see that computation time is drastically reduced once the number of iterations is large enough, with higher numbers of iterations having relatively little effect on simulation time (black line). For the comparison with Algorithm 1, the general picture is that Algorithm 2 reduces computation times relative to Algorithm 1 (grey line). The relative gain is large for 2 iterations at more than 30% (which is consistent with Table II), but less than 6% for 3 iterations or more, although the gain is non-trivial for very high numbers of types and lies in the range of 3-6% for $H = 10$ million types. Hence, Algorithm 2 delivers substantial time advantages when the number of iteration is set too low or if a good initial guess on the number of constrained types does not emerge from price iterations. On the other hand, for a good guess, there are small but non-trivial gains in computation time when the number of belief types is very large, such as several million.

We now turn to the choice of the step size k_{step}, so far set at 2. Figure 7 shows the impact of increasing the step size in steps of 2; this is done for $H = 1, 5, 10$ million types and at each number of types both 3 price iterations (black lines) and 4 price iterations (grey lines) are considered.[23] If the number of price iterations is not sufficient to obtain a good initial guess for the number of short-selling constrained types (black), then increasing the step size above 2 leads to non-trivial gains in computation speed that range from more than 20% in the case of 1 million types (left panel) to around 70% for 10 million types (right panel). However, if the number of price iterations is large enough to obtain a good guess (grey lines), changing the step size k_{step} has little impact on computation times. Intuitively, a larger step size will speed up the search for the number of constrained types, but if a good initial guess is obtained then the search will be very fast regardless, as seen in Figure 7.

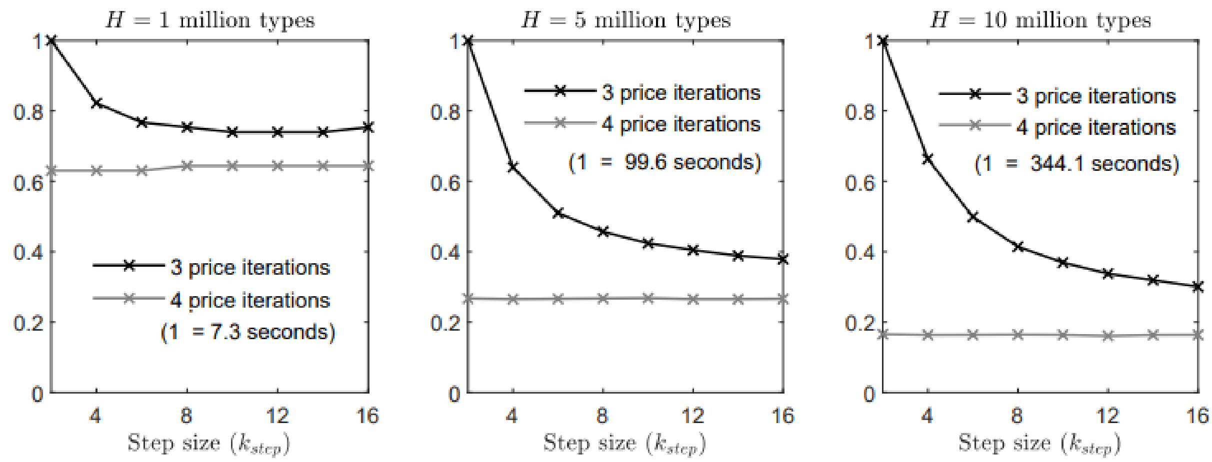

Figure 7: Algorithm 2 computation time vs step size k_{step}: various H

In summary, we have seen that Algorithm 2 reduces simulation times as expected when the number of types is very large (Figure 6) and this approach especially useful relative to Algorithm 1 if the number of price iterations is underestimated by the user (Figure 7, black), as is more likely when first encountering a particular model, or when there are changes in parameter settings. Hence, use of Algorithm 2 may help prevent excessive simulation times and is most useful relative to Algorithm 1 if little is known about the 'right' the number of price iterations (a setting whose 'sweet spot' is usually *not* known to the user ex ante).

4.3.3 Algorithm 3

Lastly, we consider Algorithm 3, the classic divide and conquer approach that uses binary search. In this case, there is *no* step size to be chosen, but price iterations may be used to obtain an initial guess $\overline{k}$ on the number of short-selling constrained types in each period.

[23] We do not plot results for the case of $H = 100,000$ types because the simulation times are essentially identical regardless of the step size and whether 3 or 4 price iterations are used.

We therefore investigate the impact of changing the number of price iterations, as well as considering the cases of *zero* iterations and ‘no guess’ on the number of short-selling constrained types (in which case we set $\underline{k} = 1$). Note that the results in Table II hint that few or no price iterations may be beneficial for Algorithm 3, unlike the other two algorithms. Intuitively, binary search does not require a good initial guess to be an effective search procedure, even in cases where the number of types H to be searched is very large.

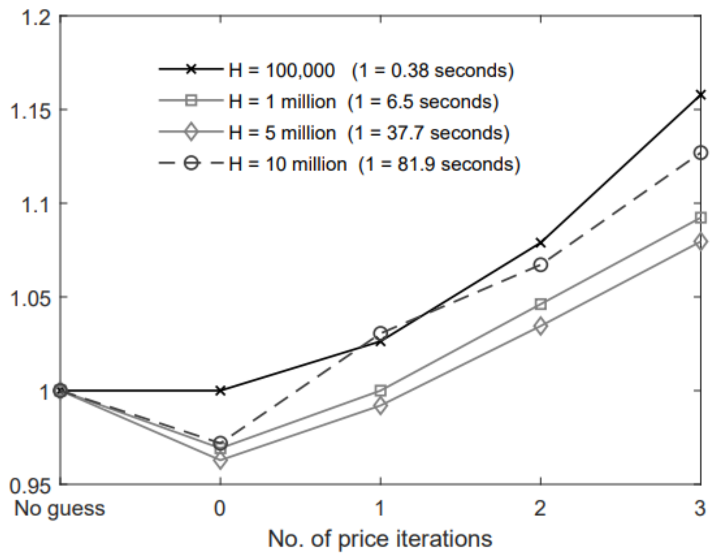

Figure 8: Algorithm 3 computation time vs number of price iterations: various H

Figure 8 shows how the solution times for Algorithm 3 vary with the number of price iterations, including the cases of (i) initial guess but *zero* price iterations, and (ii) *no* initial guess and *zero* price iterations. Differently from the other two algorithms, using multiple price iterations does not improve solution times. The fastest solution is for zero iterations in all cases in Figure 8, meaning that there is an initial guess on the number of short-selling constrained types but no subsequent updates of that guess. When the number of types is very large, the speed gains from this approach are 3-4% compared to making no guess. The gains relative to one price iteration are generally smaller, but the improvement in computation time is more than 5% when compared to numbers of iterations such as 2 or 3.

In short, there is little or no role for price iterations when using classic divide and conquer as in Algorithm 3. As a result, achieving improved computation times does not require ‘fine tuning’ of the user in response to experience of the problem at hand; this is a clear advantage from a practical perspective as there is no reliance on choosing a suitable number of price iterations to obtain a fast solution, unlike for the other two algorithms. Indeed, the results in Table II show that for two price iterations the solution times for algorithms 1 and 2 are excessive and *much worse* than for Algorithm 3; what the results above confirm is that changing the step size in Algorithm 2 does not alter that conclusion.

4.3.4 Computation times with speed improvements

Having studied how the speed of each algorithm can be improved, we now provide an overall comparison when settings are chosen by the user according to the fastest solution for a given algorithm; here we use the 'fastest' parameter settings suggested by the results in the previous section.[24] The solution times in Table IV below can be compared to the earlier ones in Table II for the same simulation path but with suboptimal algorithm settings.

Table III: Fastest computation times in seconds (Baseline simulation, $T = 100$ periods)

#Types H / Algorithm	1 (Benchmark)	2 (DivCon I)	3 (DivCon II)
$100,000$	0.38	0.38	0.38
1 million	4.7	4.7	6.3
5 million	26.6	26.3	36.1
10 million	56.3	55.8	79.6

Notes: Solution times (seconds) when algorithm parameters are set to maximize speed.

The solution times in Table IV are faster as expected. For very large numbers of types such as $H = 5$ million or $H = 10$ million there are non-trivial improvements in computation times; for example, at $H = 5$ million types, Algorithm 3 is around 10% faster than in Table II, while for $H = 10$ million types the speed gain is around 4%. These results make sense given that the use of price iterations (as in Table II) does not improve solution times for the binary search algorithm, as indicated in Figure 8. For Algorithm 1 and Algorithm 2 there are more modest improvements in computation times of around 3–6% for $H = 5$ or 10 million types, which also makes sense since the settings of four price iterations and $k_{step} = 2$ (for Algo. 2) in Table II are not too far from those that give the fastest solution in Figs. 5–7.

As before, the classical divide and conquer approach of Algorithm 3 is slower than the other two algorithms – taking around 1.5 times as long for very large numbers of types – but as noted it is also robust in the sense of not relying on a good initial guess on the number of short-selling constrained types, in contrast to Algorithm 1 and Algorithm 2.

4.4 Application: Price dynamics and wealth distribution

We now provide an application which illustrates the above algorithms in action by simulating price dynamics and wealth distribution in the above model. Specifically, we first present some bifurcation diagrams for the asset price with respect to the intensity of choice parameter; we then consider a particular price trajectory and visualize the evolving the wealth distribution.

[24]For each algorithm we report the fastest solution from 5 attempts at each number of types H. Algorithm 1 uses four price iterations (see Figure 5); Algorithm 2 uses four price iterations and $k_{step} = 2$ (see Figure 6 and Figure 7); and Algorithm 3 is run with an initial guess but *zero* price iterations (see Figure 8).

4.4.1 Price dynamics

We start by studying price dynamics as the intensity of choice parameter β is increased, implying that investors are more willing switch their type to relatively profitable predictors. Studying the impact of the intensity of choice on price dynamics is a standard exercise in the literature, but our analysis here differs in considering a model with short-selling constraints and a very large number of heterogeneous beliefs. This exercise is a highly computationally-intensive problem because, in addition to the large number of occasionally binding constraints, the initial conditions should be varied at each parameter setting.

Relative to the previous section, we consider some small changes in the price predictors. In particular, we assume the chartist beliefs g_h are drawn from a uniform distribution on $(1.1, 1.3)$ and we consider fundamental beliefs b_h linearly-spaced on the interval $[-0.1, 0]$. These choices ensure there is a negative attractor for the price (given a negative initial value) and also allow us to keep contact with previous work in the literature.[25] Figure 9 plots the resulting numerical bifurcation diagram for initial prices $x_0 \in (-1, 0)$ and $H = 100,000$; we focus on the *deterministic skeleton* by setting $d_t = \overline{d}$ for all t in the simulations.[26]

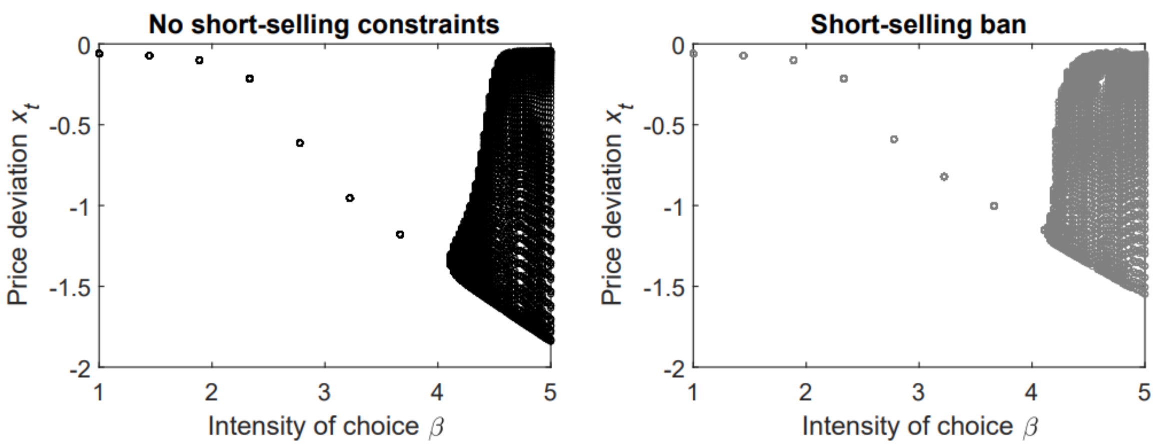

Figure 9: Numeric bifurcation diagrams: with and without short-selling constraints. For each β, we plot 160 points following a transitory of 510 periods from initial prices $x_0 \in (-1, 0)$. $H = 100,000$ types and the final 40 points of each simulation are plotted for 4 different x_0.

For sufficiently low values of the intensity of choice β, price trajectories converge to a non-fundamental steady state price. The steady-state price becomes more negative as β is increased, and for large enough β the non-fundamental steady state loses stability through a bifurcation to (quasi-)periodic dynamics. Note that such boom-bust dynamics appear

[25] Anufriev and Tuinstra (2013) consider a two-type model with $g = 1.2$ (chartist) and $b = 0$ at cost of 1 (fundamentalist). Hence, the mean chartist belief in our heterogeneous beliefs model matches theirs, while we allow modest negative bias in fundamental beliefs. They consider both positive and negative initial values and plot both price attractors, while we focus on the negative attractor only as in Hatcher (2024b).

[26] We consider 4 different initial values at each β and plot the final 40 values from each simulation.

more convincing empirically than convergence to a steady-state price.[27] For unrestricted short-selling (left panel), we see a similar attractor to the two-type case in Anufriev and Tuinstra (2013), though only a negative steady-state price is seen here, due to the pessimism of fundamentalists. With a short-selling ban (right panel), the price attractor is qualitatively similar, but the range of price fluctuations after the bifurcation point is somewhat smaller, with large negative downside values of the price being absent because negative positions cannot be taken. Hence, relative to unfettered short-selling, a ban prevents strong undervaluation of the risky asset, and is therefore potentially attractive from a policy perspective.[28]

Table IV: Computation times for bifurcation diagram in Figure 9

No constraints	1 (Benchmark)	2 (DivCon I)	3 (DivCon II)
76.1	95.0	94.8	96.5

Notes: Computation times (seconds), including plotting time for the orbit diagram.

In Table IV we report computation times for the numerical bifurcation diagrams in Figure 9. Given the long simulation horizon (550 periods) and the large number of simulations (180), the diagram takes longer than 1 minute to compute, even for the case of unrestricted short-selling. Algorithms 1 and 2 have similar computation times of just over 1.5 minutes, while Algorithm 3 is slower by around 1.5 seconds; these results make sense given that there was little to choose between the algorithms in terms of solution speed for $H = 100,000$ types in the timed simulations above (see Table I and Table II). The results in Table IV use a parallel pool in MATLAB (i.e. parfor loop) with 14 workers. Use of a dedicated graphics processing unit (GPU) would be one way to reduce computation times even further.[29]

Table V: Computation times at a typical β value

H / Algo.	No constraints	1 (Benchmark)	2 (DivCon I)	3 (DivCon II)
100,000	2.3	3.5	3.4	3.2
1 million	33.0	79.7	71.2	114.3
5 million	192.3	443.8	403.7	702.2
10 million	396.4	1042.9	925.5	1509.3

Notes: Computation times (seconds) at a typical β in a bifurcation diagram. Four x_0 values at each β and $T = 550$ periods as in Figure 9. Reported time: average from 4 different β values.

To shed light on the implications of much larger numbers of investor types, we report in Table V computation times at a 'typical' β value in the bifurcation diagram, for much larger

[27] Asset pricing models with evolutionary competition have been found to perform well empirically partly as a result of their boom-bust dynamics; see e.g. Boswijk et al. (2007) and Chiarella et al. (2014).

[28] The minimum price with a short-selling ban is around −1.56 versus −1.84 with unrestricted short-selling. Recall the intrinsic value of the risky asset is the fundamental price $\overline{p}$, so there is no mispricing for $x = 0$.

[29] For examples where GPUs speed up solution times, see Aldrich (2014) or Hatcher and Scheffel (2016).

numbers of types up to $H = 10$ million.[30] We then see the benefits of Algorithm 2: for 1–10 million types, computation times are reduced by about 10% or more relative to Algorithm 1, while relative performance of Algorithm 3 deteriorates somewhat compared to Table IV.

4.4.2 Wealth distribution

We now turn to the impact of a short-selling ban on wealth distribution. We focus on the price paths shown in Figure 4 by assigning the same beliefs as in sections 4.1–4.3, an initial price $x_0 = 5$ and dividend shocks of zero. Figure 10 plots the price with and without a short-selling ban over 100 periods, along with two summary measures of wealth inequality: the Gini coefficient and 90:10 ratio. Both measures indicate that wealth inequality across types is reduced by a short-selling ban. Wealth inequality initially rises in both cases, with the increase prolonged for a short-selling ban as the price cycles (left) 'die out' slowly.[31]

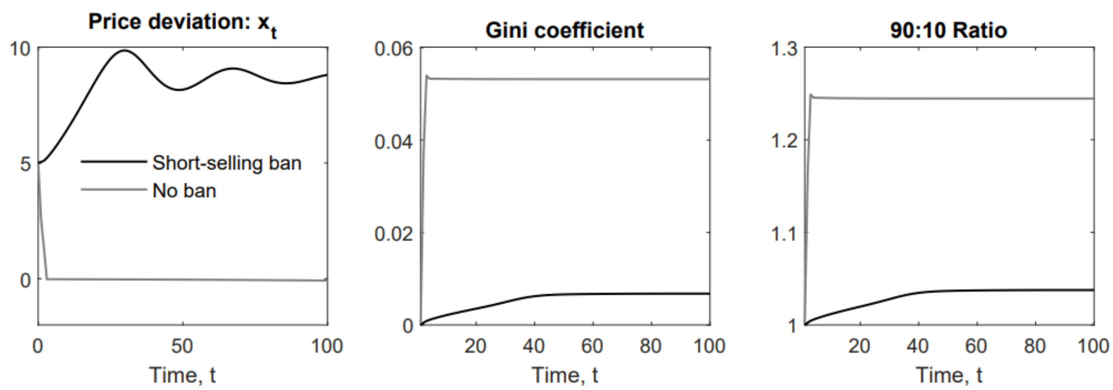

Figure 10: Asset price and two measures of wealth inequality when $H = 100,000$: deterministic simulation of 100 periods. Simulated series without a ban shown for comparison (grey).

Figure 11 plots the wealth distribution across types at two points in time: period 5 (top) and period 25 (bottom). If short-selling is unrestricted (left panels), the wealth distribution looks roughly bimodal, which makes sense given that types are heterogeneous but drawn from one of two groups. The fundamental types are more successful, giving a concentration of types at high wealth levels, while the chartist types have much lower wealth of around 80% of the maximum. In the later snapshot at $t = 25$, the lower end of the distribution looks very similar, but at the upper end there is a greater concentration of types near the highest wealth level. With a short-selling ban (right panels), wealth is highly concentrated and initially within 1% of the maximum wealth level (period 5, top right) and as time passes

[30] Again, a parfor loop was used. To obtain the time at a 'typical' β value we ran simulations for four initial conditions at four different values of β and computed the average by dividing the cumulative computation time by four. For Algorithm 2 the step size was set at $k_{step} = 2$ for the simulations reported in Table V.

[31] The initial rise in inequality makes sense since we start types from equal initial wealth in our simulation. Note that wealth evolves according to $w_{t+1,h} = (p_{t+1} + d_{t+1})z_{t,h} + (1 + r)(w_{t,h} - p_t z_{t,h})$; see (1).

the tail gets longer, although many types remain near the highest wealth level (period 25, bottom right). In short, inequality across types is strongly reduced by a short-selling ban, as this prevents a vast performance differential between fundamentalists and chartists.

Note that the wealth distribution above is not necessarily informative about inequality across *individuals*, because the latter will depend on the fractions of the population that adopt each type. Therefore, as a final exercise we set $\beta = 0$ so that each type is an *individual* investor that sticks with the same predictor in every time period; see Figure 12. In this case we see a similar pattern, but wealth inequality is more stable because 'winners' do not have a reinforcing effect on the price and returns as under evolutionary competition.

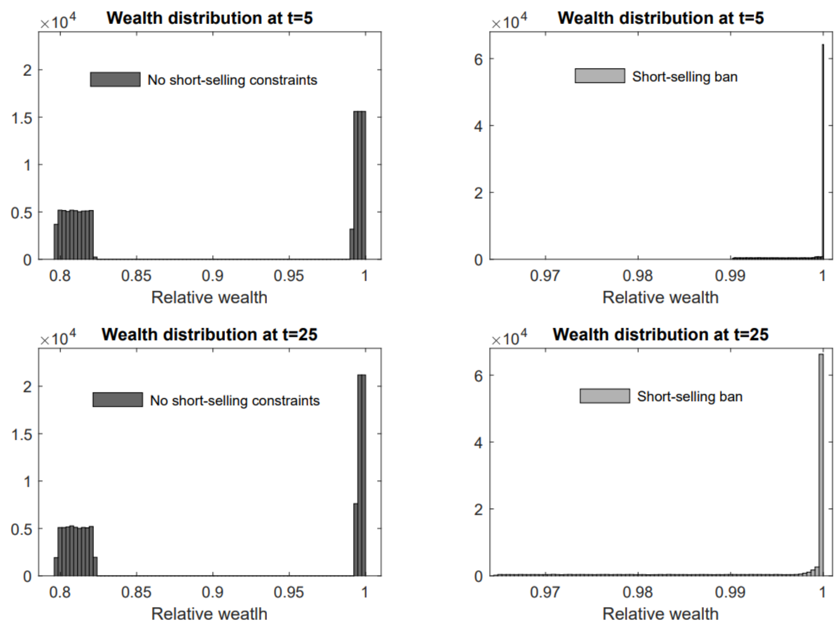

Figure 11: Wealth distribution across *types* with and without short-selling ban ($\beta = 5$)

The above results are not intended as a description of the empirical wealth distribution. The observed Gini coefficient is low by quantitative standards, though we do see a very strong concentration of wealth among relatively few types as in the data. We do not view the model presented as quantitative, but instead as a lens through which to study the impact of short-selling regulation – in the form of a short-selling ban – as compared to unrestricted short-selling. We have tried to highlight this aspect in the above examples.

4.5 Discussion

We have compared the speed of three different algorithms for solving heterogeneous-belief asset pricing models with short-selling constraints and many investor types; as our testing ground we used a model with evolutionary competition of many belief types. We also provided an application that studied price dynamics and wealth distribution using this model.

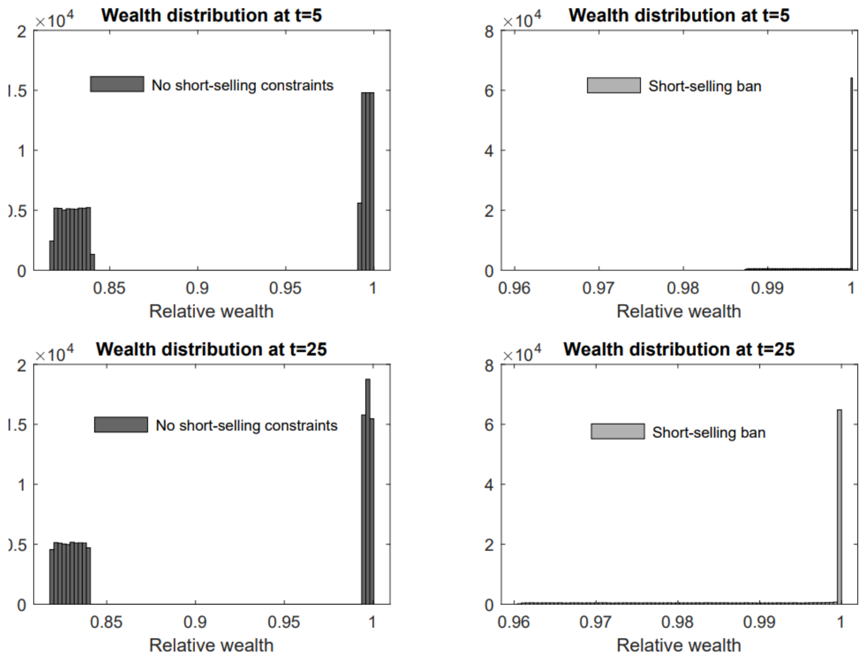

Figure 12: Wealth distribution across *individuals* with and without short-selling ban ($\beta = 0$)

A performance comparison of the algorithms showed that a *divide and conquer* approach has potential benefits relative to a benchmark algorithm that relies on a price iterations approach to obtain a good initial guess on the number of short-selling constrained types (Algorithm 1). In particular, we saw that if too few price iterations are used to obtain a good initial guess, an algorithm that builds on the benchmark by adding a divide and conquer component (Algorithm 2) can substantially improve speed, and that algorithm is in turn outperformed by a classical divide and conquer algorithm that uses a binary search approach to find the number of short-selling constrained types in equilibrium (Algorithm 3).

On the other hand, when a good initial guess can be obtained – as with a large enough number of price iterations – there is a dramatic improvement in the speed of Algorithm 1 and Algorithm 2 since a weaker search procedure for finding the number of constrained types can be compensated for if the initial guess is very good. In this case, Algorithm 1 and Algorithm 2 are somewhat faster than Algorithm 3 and can solve a model with 5 million belief types in around half a minute, with Algorithm 2 being slightly faster than Algorithm 1 due to the use of a 'mini' divide and conquer approach in the search part of the algorithm.

Thus, Algorithm 3 is ideal for solving models with little prior knowledge, while Algorithm 2 is expected to be fastest if users can 'fine tune' the algorithm settings. These results highlight the potential advantages of using both price iterations and divide and conquer approaches when solving models with short-selling constraints and many belief types.

Our application studied price dynamics and wealth distribution by constructing bifurcation diagrams for the price as the intensity of choice parameter was increased and by plotting the wealth distribution across many types. The former exercise is a standard method of analysis in the literature but is computationally very intensive (especially if short-selling constraints are present), while the latter question of wealth distribution has been less studied but is of particular relevance when considering very large populations of investors, such as the many buyers and sellers of a stock market index over a given time period.

The bifurcation analysis illustrates the rich variety of price dynamics of possible in these models and there are important qualitative differences relative to the case of no short-selling ban. Most notably, a short-selling ban avoids more extreme drops in asset prices relative to fundamentals, and so from a policy perspective helps to stabilize prices on the downside.

The computation times in the bifurcation exercise favour Algorithm 2, with speed gains of 10% or more over the benchmark Algorithm 1 for very large numbers of belief types, and it is notable that fast solution algorithms make such an exercise tractable despite the high computational intensity. On the other hand, the analysis of the wealth distribution illustrates the potential for studying wealth dispersion across large populations in asset pricing models, including the implications of regulations such as short-selling constraints for wealth inequality, which is presumably a concern for some policymakers.

5 Conclusion

In this paper we studied dynamic behavioural asset pricing models with many beliefs and short-selling constraints that rule out negative positions (i.e. a short-selling *ban*). We presented three different algorithms for fast solution of such models for a very large number of occasionally-binding constraints (one per investor type), up to 10 million. The first algorithm is a benchmark algorithm introduced in previous work but whose speed has not been studied in detail or linked to user choices in the algorithm itself. The other two algorithms employ a divide and conquer approach: one is a variation on the benchmark algorithm designed to improve its speed, while the other employs a classic divide and conquer approach via a binary search approach. All three algorithms enable fast solution times on a standard laptop or desktop computer for large numbers of investor types such as 100,000 up to 10 million.

The relative performance of the algorithms differs substantially depending on the setting of user inputs, such as the number of price iterations used to make an initial guess on the number of short-selling constrained types. Although such settings can be tuned for a particular problem once it is well understood, this knowledge is absent *ex ante*, so robustness matters as well as the fastest potential solution times. Our results indicate that the classical divide and conquer approach (Algorithm 3, binary search) is highly robust in that solution times do not vary dramatically depending on the user settings; indeed, we found this approach performs best when no price iterations are used, so there is no reliance on a good initial guess. As a result, this approach works best for problems that are not yet well understood by the user and also in cases where price iterations cannot be relied upon to always give a good initial guess for the number of short-selling constrained types.

On the other hand, we found that if the number of price iterations is large enough, then a good guess for the number of constrained types is typically obtained. In these circumstances, Algorithm 1 and Algorithm 2 outperform the classic divide and conquer approach in Algorithm 3 by giving substantially faster solution times: around a 30% reduction in solution time for our numerical example with 1–10 million investor types and stochastic simulation of 100 periods. Moreover, Algorithm 2 is faster than Algorithm 1 due to the use of divide and conquer to speed up the process of finding the equilibrium number of constrained types in each period; however, the relative reduction in computation time is not that large because having a good initial guess reduces the potential gains of a divide and conquer approach.

Future work could proceed in three directions. First, future work could try to extend the divide and conquer algorithms in the present paper to the tricker case of a short-selling *tax* as in Anufriev and Tuinstra (2013) and Hatcher (2024a), where the potential speed gains of an effective search procedure could be even larger. Second, it is an open question whether the algorithms in the present paper could be extended to heterogeneous-belief models with piecewise-linear expectation rules or strategies (e.g. price beliefs that depend on price thresholds or entry linked to such thresholds; see Tramontana et al., 2010, 2015); if so, the algorithms presented here would have wider applicability beyond short-selling constraints. Third, there is scope for work that studies in more detail the consequences of short-selling regulations in models with rich heterogeneity and very large numbers of investor types.

Funding / Competing Interests

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

The authors have no relevant financial or non-financial interests to disclose.

References

Aldrich, E. M. (2014). Gpu computing in economics. In *Handbook of computational economics*, volume 3, pages 557–598. Elsevier.

Anufriev, M. and Tuinstra, J. (2013). The impact of short-selling constraints on financial market stability in a heterogeneous agents model. *Journal of Economic Dynamics and Control*, 37(8):1523–1543.

Ap Gwilym, R. (2010). Can behavioral finance models account for historical asset prices? *Economics Letters*, 108(2):187–189.

Barberis, N., Greenwood, R., Jin, L., and Shleifer, A. (2018). Extrapolation and bubbles. *Journal of Financial Economics*, 129(2):203–227.

Boswijk, H. P., Hommes, C. H., and Manzan, S. (2007). Behavioral heterogeneity in stock prices. *Journal of Economic Dynamics and Control*, 31(6):1938–1970.

Brock, W. A. and Hommes, C. H. (1997). A rational route to randomness. *Econometrica: Journal of the Econometric Society*, pages 1059–1095.

Brock, W. A. and Hommes, C. H. (1998). Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8-9):1235–1274.

Brock, W. A., Hommes, C. H., and Wagener, F. O. (2005). Evolutionary dynamics in markets with many trader types. *Journal of Mathematical Economics*, 41(1-2):7–42.

Chiarella, C., He, X.-Z., and Zwinkels, R. C. (2014). Heterogeneous expectations in asset pricing: Empirical evidence from the s&p500. *Journal of Economic Behavior & Organization*, 105:1–16.

Dercole, F. and Radi, D. (2020). Does the "uptick rule" stabilize the stock market? insights from adaptive rational equilibrium dynamics. *Chaos, Solitons & Fractals*, 130:109426.

Hatcher, M. (2024a). Heterogeneous beliefs and short selling taxes: A note. *Journal of Economic Dynamics and Control*, 168:104970.

Hatcher, M. (2024b). Solving heterogeneous-belief asset pricing models with short-selling constraints and many agents. *Macroeconomic Dynamics*, 28(8):1715–1738.

Hatcher, M. and Hellmann, T. (2024). Communication, networks and asset price dynamics: a survey. *Journal of Economic Interaction and Coordination*, 19(1):1–58.

Hatcher, M. and Hellmann, T. (2025). Networks, beliefs, and asset prices. *Journal of Economic Dynamics and Control*, 173:105059.

Hatcher, M. C. and Scheffel, E. M. (2016). Solving the incomplete markets model in parallel using gpu computing and the krusell–smith algorithm. *Computational Economics*, 48:569–591.

Hommes, C. (2013). *Behavioral rationality and heterogeneous expectations in complex economic systems*. Cambridge University Press.

Hommes, C. H. (2006). Heterogeneous agent models in economics and finance. *Handbook of computational economics*, 2:1109–1186.

in't Veld, D. (2016). Adverse effects of leverage and short-selling constraints in a financial market model with heterogeneous agents. *Journal of Economic Dynamics and Control*, 69:45–67.

Kirman, A. (1991). Epidemics of opinion and speculative bubbles in financial markets. *Money and Financial Markets, Macmillan*.

Kirman, A. (1993). Ants, rationality, and recruitment. *The Quarterly Journal of Economics*, 108(1):137–156.

LeBaron, B., Arthur, W. B., and Palmer, R. (1999). Time series properties of an artificial stock market. *Journal of Economic Dynamics and Control*, 23(9-10):1487–1516.

Louridas, P. (2020). *Algorithms*. MIT Press.

Panchenko, V., Gerasymchuk, S., and Pavlov, O. V. (2013). Asset price dynamics with heterogeneous beliefs and local network interactions. *Journal of Economic Dynamics and Control*, 37(12):2623–2642.

Scheinkman, J. A. and Xiong, W. (2003). Overconfidence and speculative bubbles. *Journal of Political Economy*, 111(6):1183–1220.

Siciliano, G. and Ventoruzzo, M. (2020). Banning cassandra from the market? an empirical analysis of short-selling bans during the covid-19 crisis. *European Company and Financial Law Review*, 17(3-4):386–418.

Tramontana, F., Westerhoff, F., and Gardini, L. (2010). On the complicated price dynamics of a simple one-dimensional discontinuous financial market model with heterogeneous interacting traders. *Journal of Economic Behavior & Organization*, 74(3):187–205.

Tramontana, F., Westerhoff, F., and Gardini, L. (2015). A simple financial market model with chartists and fundamentalists: market entry levels and discontinuities. *Mathematics and Computers in Simulation*, 108:16–40.

Westerhoff, F. H. (2004). Multiasset market dynamics. *Macroeconomic Dynamics*, 8(5):596–616.

Westerhoff, F. H. (2016). The use of agent-based financial market models to test the effectiveness of regulatory policies. In *Agent Based Models for Economic Policy Advice*, pages 195–227. De Gruyter Oldenbourg.