Institute AIFB
Universität Karlsruhe (TH)

Time Series Properties from an Artificial Stock Market with a Walrasian Auctioneer

Thomas Stümpert, Detlef Seese, Malte Sunderkötter

Overview

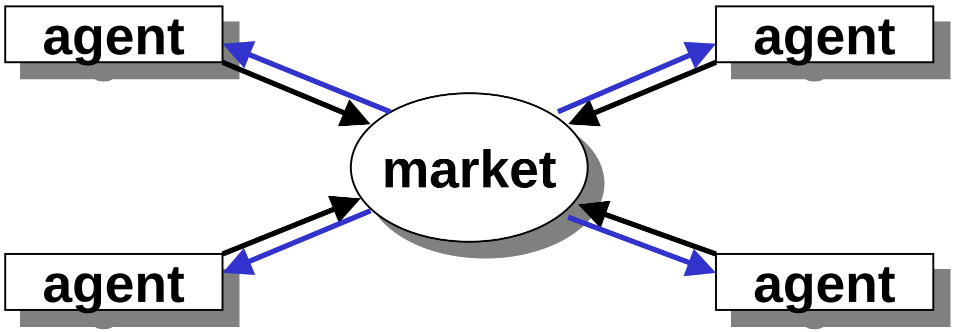
- Artificial Stock Markets: why?
- Lessons learned from SF-ASM

- Walrasian adaptive simulation market (WASIM)
  - The model
  - Causality between wealth of agents and market prices

- Empirical results
  - Wealth concentrations and monopolies lead to price fluctuations
  - Taxation stabilizes market prices

Artificial Stock Markets

Interaction of economic agents generates the return distribution



Objectives

- Unterstand how agents learn on markets
- Interdependency of mikrostructure and makrolevel
- Model should be simple

→ Artificial agent-based stock market

Lessons learned from SF-ASM (1)

Agents adapt their strategy slowly to changes (slow genetic algorithm):

Trading volume is low, no crashes and bubbles, no technical trading

→ **Efficient Market Hypothesis**

Agents adapt their strategy fast to changes (fast genetic algorithm):

Deviation from intrinsic value, bubbles and crashes arise, technical trading and trading volume more than 3 times higher

→ **Market sentiment / Behavioural Finance**

Lessons learned from SF-ASM (2)

Market design in SF-ASM (Ehrentreich 2002):

1. Non bit-neutral mutation operator in SF-ASM leads to technical trading and causes phases of over- and undervaluation

2. No technical trading with bit-neutral mutation operator

3. Market price moves on average parallel to the intrinsic value

Intrinsic asset price in SF-ASM and WASIM

Calculation of the intrinsic asset price for a risk-neutral investor

$$d_t = \bar{d} + \rho \cdot \left(d_{t-1} - \bar{d}\right) + \varepsilon_t \qquad \frac{r}{d_t / p_t} = 1 \Rightarrow p_t = p_t^*$$

$$E_t\left[d_{t+1}\right] = E_t\left[\bar{d} + \rho\left(d_t - \bar{d}\right) + \varepsilon_{t+1}\right] = \bar{d} + \rho d_t$$

$$E_t\left[d_{t+2}\right] = E_t\left[\bar{d} + \rho\left(d_{t+1} - \bar{d}\right) + \varepsilon_{t+2}\right]$$

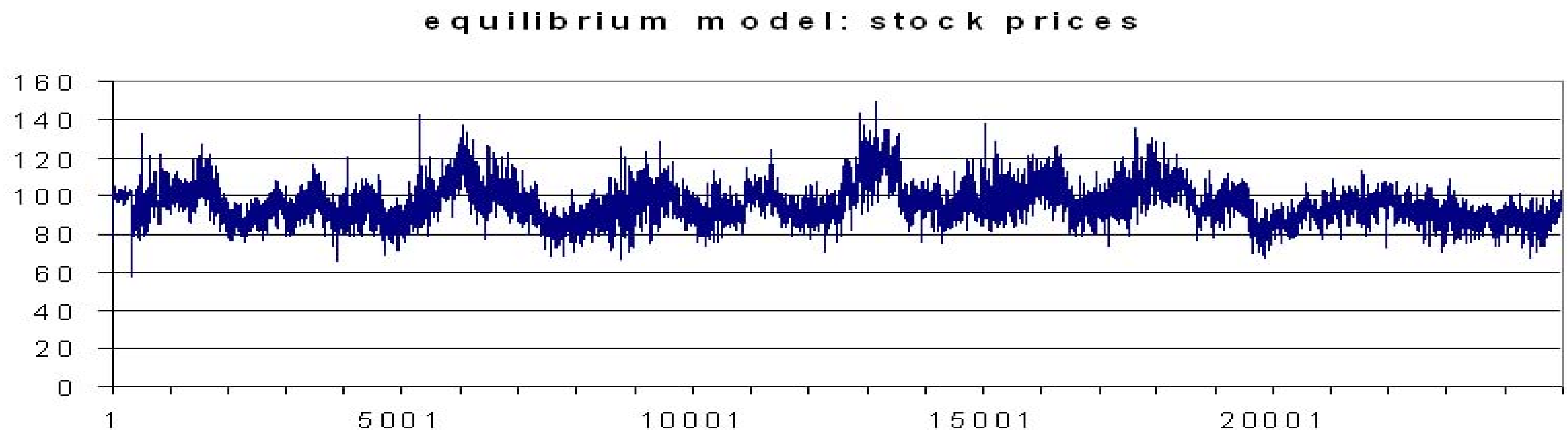
$$= E_t\left[\bar{d} + \rho^2 d_t + \rho\varepsilon_{t+1} + \varepsilon_{t+2}\right] = \bar{d} + \rho^2 d_t$$

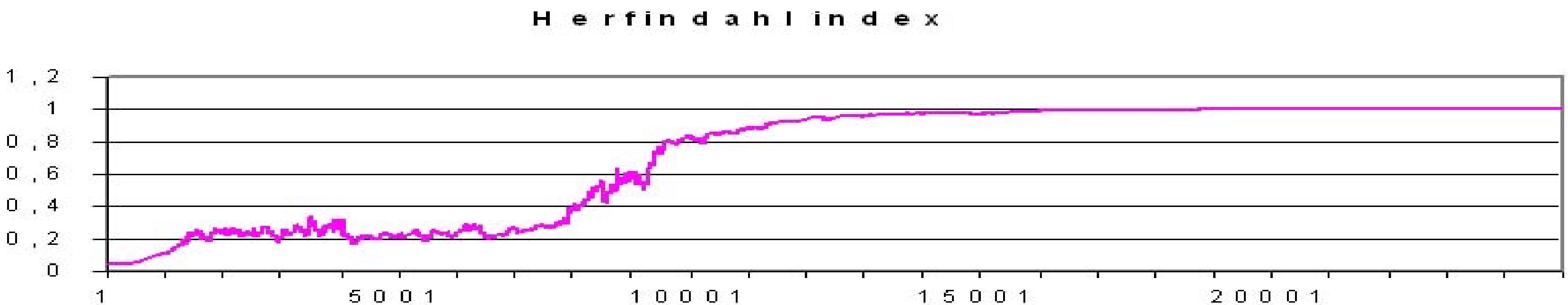
$$E_t\left[d_{t+k}\right] = \ldots \qquad = \bar{d} + \rho^k d_t \xrightarrow{k \to \infty} \bar{d}$$

Lessons learned from SF-ASM (3)

Limitations (LeBaron 2002):
The underlying equilibrium model does not connect the price function of the stock to the wealth of the agents

Every agent has the same impact on the price setting





Lessons learned from SF-ASM (4)

Santa Fe artificial stock market introduces trading restrictions:

- Each agent is allowed to trade max. 10 shares per period
- Short selling is forbidden

➔ Equilibrium model with deviation from equilibrium

Wealth of each agent has no influence on buy/sell orders

Overview: SF-ASM and WASIM

Santa Fe Artificial Stock Market (SF-ASM):

- Equilibrium model PLUS trading restrictions
- Wealth of agents does not influence the market price
- Mutation operator generalizes the forecasting rules

WASIM:

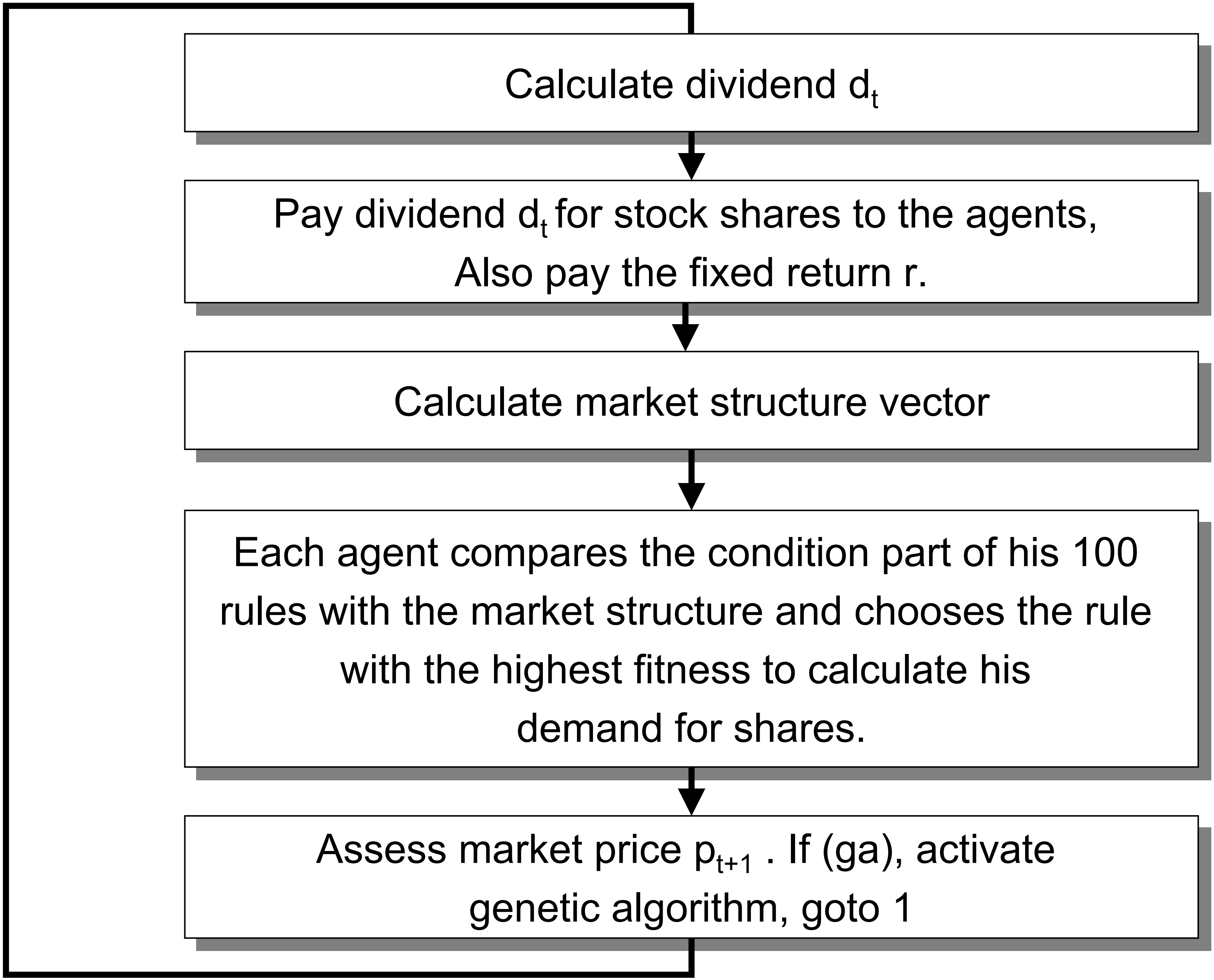
- Iteration towards equilibrium even if trading restrictions are used
- Causality between wealth of agents and market prices
- Bit-neutral mutation and fast genetic algorithm

WASIM – Walrasian adaptive simulation market

- Variant of the Santa Fe artificial stock market with an additional Walrasian auctioneer

- N=25 agents invest in a
  - risky stock with price p_t and dividend d_t
  - risik-free asset with fixed return r

- Each agent maximizes his one-period return:
  - Agent chooses the rule with the highest fitness from 100 forecasting rules for his buy/sell order
  - Agent uses a genetic algorithm to produce new rules.


Calculate dividend d_t
Pay dividend d_t for stock shares to the agents,
Also pay the fixed return r.
Calculate market structure vector
Each agent compares the condition part of his 100 rules with the market structure and chooses the rule with the highest fitness to calculate his demand for shares.
Assess market price p_{t+1} . If (ga), activate genetic algorithm, goto 1

Market structure vector (1/2)

Calculate d_t: $$d_t = \bar{d} + \rho(d_{t-1} - \bar{d}) + \varepsilon_t$$ is a AR(1)-process

Calculate markt structure vector:

$b_1 = (p_t*r/d_t>1/4)$
$b_2 = (p_t*r/d_t>1/2)$
$b_3 = (p_t*r/d_t>3/4)$
$b_4 = (p_t*r/d_t>7/8)$
$b_5 = (p_t*r/d_t>1)$
$b_6 = (p_t*r/d_t>9/8)$
$b_7 = (p_t>MA_5)$
$b_8 = (p_t>MA_{10})$
$b_9 = (p_t>MA_{100})$
$b_{10} = (p_t>MA_{500})$
$b_{11} = 1$
$b_{12} = 0$

Market structure vector (2/2)

Bits **1-6** of the market structure vector:

„current price*****return**/**dividend **>** 0.25, 0.5, 0.75, 0.875, 1.0, 1.125"

Undervaluation / overvaluation → **fundamental bits**

Bits **7-10**:

„current price **>** 5, 10, 100, 500 moving average"

price trends → **technical trading bits**

Bits **11** and **12** are „on" (1) resp. „off" (0): → **control / test bits**

Agents and demand functions

3. Activation of a forecasting rule of agent *i*:

$PR_{t,i,j}$ = $(m_1,m_2,m_3,\ldots,m_{12})$	$(a_{t,i,j}, b_{t,i,j})$	$(f_{t,i,j})$
(condition part)	**(forecasting part)**	**(fitness)**

$$\boldsymbol{m}_i \in \{0,1,\#\}$$

$$\boldsymbol{a} \in [0.7, 1.2]$$

$$b \in [-10, 19]$$

$$f_{t,i,j} = M - e^2_{t,i,j} - Cs$$

$$e^2_{t,i,j} = (1-\theta)e^2_{t-1,i,j} + \theta\left[(p_{t+1} + d_{t+1}) - E_{t,i,j}(p_{t+1} + d_{t+1})\right]^2$$

Forecast of agent *i*:

$$\hat{E}_{i,t}(p_{t+1} + d_{t+1}) = a_j(p_t + d_t) + b_j$$

Agents and demand functions

Each agent has 100 forecasting rules

Each agent compares the condition part of his rules with the current market structure

- 0 = condition is true
- 1 = condition is false
- # = don‘t care

A forecasting rule is activated if all non-# symbols match the market structure

Agents and demand functions

Agents use the parameters a and b to forecast the future market price

Forecasting is a linear combination of prices and dividends (e.g. $E(p_{t+1}+d_{t+1}) = a\,(p_t+d_t) + b$)

From all active rules the agent chooses the rule with the highest fitness for his forecast

Then he calculates his demand for shares $x_{i,t}$

Agents and demand functions

Demand of agent *i*:

$$x_{i,t} = \frac{\hat{E}_{i,t}(p_{t+1} + d_{t+1}) - (1+r)p_t}{\gamma \cdot \hat{\sigma}^2_{p+d,i}}$$

market clearing / equilibrium model

$$\sum_{i=1}^{N} x_{i,t} = N$$

Agents and rule improvement

Parameter adaptation:

At the beginning of a period, the new price and dividend is announced. Each agent updates the precision of its forecasting rules

Genetic algorithm:

A genetic algorithm is activated every 250 periods to produce new trading rules

Walrasian auctioneer

The auctioneer executes the following algorithm for each agent i and iterates the calculation of the new price towards the equilibrium price up to a given resolution ε:

- Calculate the demand for shares x_{it} of agent i
- Calculate the number of shares to be traded, $\Delta x_{it} = (x_{it} - x_{i,t-1})$
- If $\Delta x_{it} < -max_{trade}$ then $\Delta x_{it} = -max_{trade}$
- If $\Delta x_{it} p > C_{it}$ then $\Delta x_{it} = C_{it} / p$
- If $\Delta x_{it} > max_{trade}$ then $\Delta x_{it} = max_{trade}$
- Calculate $\Delta p = p * (supply\text{-}demand) * f / (N*100)$

$C_{it} = W_{i,t-1} - x_{i,t-1} * p_{t-1}$

freely available cash of agent i

Genetic algorithm (1/4)

Every agent uses a genetic algorithm to improve his forecasting rules and to adapt them to market behaviour:

On average every 250 periods the 20 rules with the lowest fitness are replaced

by recombination (p=0.1) or

by mutation (p=0.9)

Genetic algorithm (2/4)

Mutation

Select 1 rule to mutate with tournament selection
- Forecasting part:
  - Choose new a and b (p=0.2)
  - Add small numbers to the old values of a and b
- Condition part:
  - Mark each bit independently with p=0.03
  - Change each marked bit according to transition probabilities

Genetic algorithm (3/4)

Transition probabilities for mutation:

SF-ASM:

$P(0 \Rightarrow 0)=0$	$P(0 \Rightarrow 1)=1/3$	$P(0 \Rightarrow \#)=2/3$
$P(1 \Rightarrow 0)=1/3$	$P(1 \Rightarrow 1)=0$	$P(1 \Rightarrow \#)=2/3$
$P(\# \Rightarrow 0)=1/3$	$P(\# \Rightarrow 1)=1/3$	$P(\# \Rightarrow \#)=1/3$

WASIM:

$P(0 \Rightarrow 0)=0$	$P(0 \Rightarrow 1)=1/3$	$P(0 \Rightarrow \#)=2/3$
$P(1 \Rightarrow 0)=1/3$	$P(1 \Rightarrow 1)=0$	$P(1 \Rightarrow \#)=2/3$
$P(\# \Rightarrow 0)=1/3$	$P(\# \Rightarrow 1)=2/3$	$P(\# \Rightarrow \#)=0$

Keeps the fractions of *0*, *1* and # constant at 1/3

Genetic algorithm (4/4)

Recombination

Select two parent rules with tournament selection

- Forecasting part:
  - Select randomly from:
    - Choose the forecast part of one parent rule
    - Combine the two old forecasting parts
    - Calculate the new forecasting part using weighted averages of the parent forecasting parts
- Condition part:
  - Uniform crossover

Herfindahl indices (1/2)

Ownership ratio:

$$\rho_{ownership}(t) = \frac{\sum_{i=1}^{N} x_{it}^2 - N}{N^2 - N}$$

Wealth ratio:

$$\rho_{wealth}(t) = \frac{\sum_{i=1}^{N} W_{it}^2 - W_t^N}{\left(W_t^N\right)^2 - W_t^N}$$

$W_t^N = \sum_{i=1}^{N} W_{i,t}$ denotes the cumulative wealth of all agents in t

Herfindahl indices (2/2)

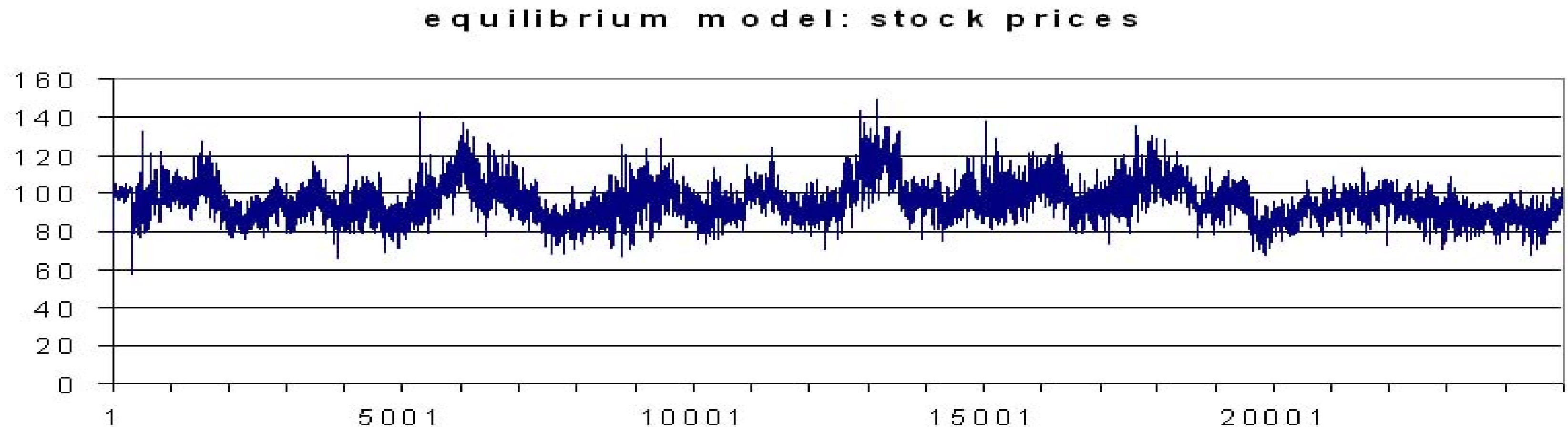
Measure for wealth distribution:

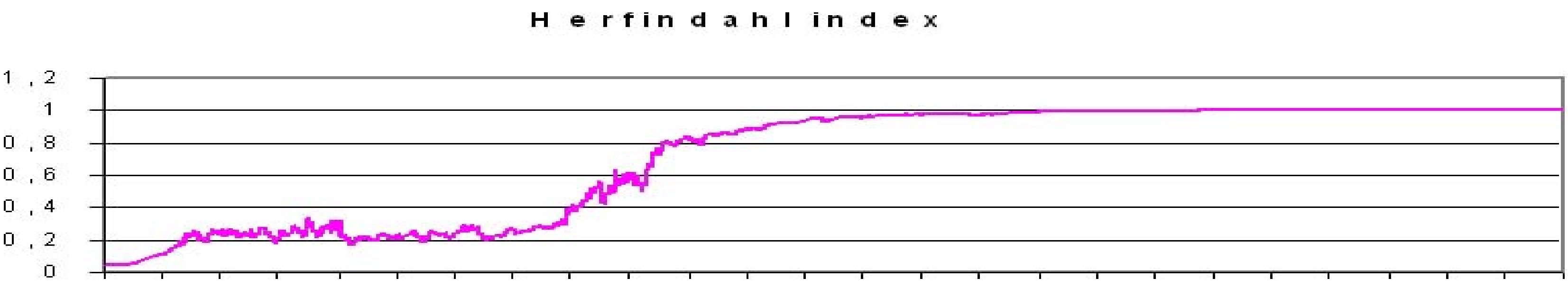
Herfindahl index:

$\rho_w \in (0,1)$

$\rho_w = 0$: Wealth is equally distributed among all agents

$\rho_w = 1$: One agent has accumulated all wealth (monopoly)





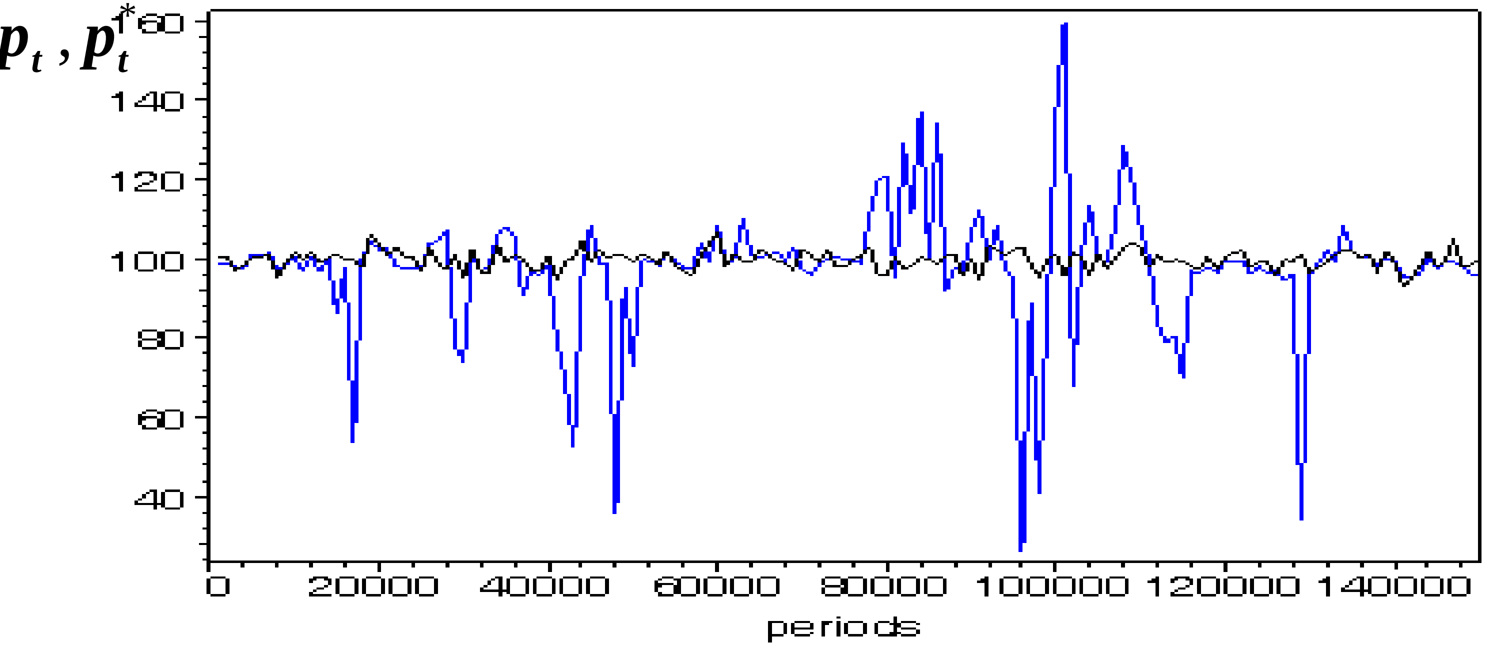
Parameter settings (1/2)

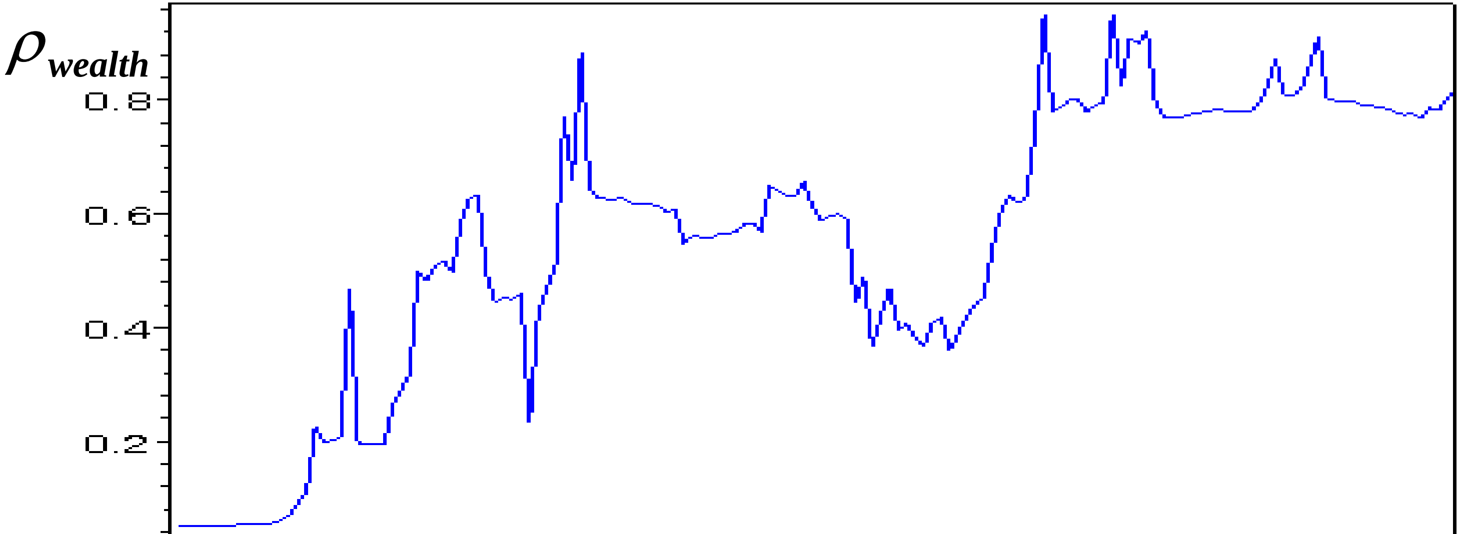
Parameter	Value	Description
N	25	Number of agents (= number of shares)
M	100	Number of forecasting rules
T	150 000	Number of periods
ε	0.0001	Walras resolution
γ	0.5	Risk aversion coefficient
p_0	100	Initial stock price
d_m	10	Mean value of dividends
r	0.10	Return of risk-free asset
σ	0.1	Initial fitness of a forecasting rule

Parameter settings (2/2)

Parameter	Value	Description
W	100	Initial wealth of agent i
θ	0.1	Smoothing parameter of the fitness
max_{own}	25	Ownership restriction in shares/agent
max_{trade}	25	Trading restrictions in shares/agent
ga_{min}	200	Step width of the genetic algorithm
ga_{max}	300	Step width of the genetic algorithm

Empirical Results without taxation





Empirical Results without taxation

The Walrasian auctioneer produces phases of **over- and undervaluation** even when using the bit-neutral mutation operator.

Why? linking the **wealth of agents** to the price setting

What else?

iterating towards **equilibrium** even with active **trading restrictions**

How to avoid high fluctuations / Taxation

- Extreme wealth concencntrations lead to monopolies
  ➔ redistribute wealth

Regulation of price volatility by introduction of a Tobin tax:

Every agent has to pay a tax of 5% every 100 periods

The collected wealth is redistributed equally to all agents

Avoid high concentrations of wealth

Market stability

Taxation in WASIM

Seed-no	ρ_1^{notax}	ρ_J^{notax}	ρ_1^{tax}	ρ_J^{tax}
1	0,041	0,893	0,040	0,041
2	0,052	0,451	0,041	0,040
3	0,096	0,473	0,041	0,041
4	0,056	0,553	0,041	0,041
5	0,210	0,619	0,041	0,040
6	0,173	0,830	0,041	0,102
7	0,096	0,418	0,042	0,040
8	0,061	0,581	0,041	0,040
9	0,088	0,638	0,041	0,041
10	0,074	0,809	0,041	0,040
11	0,262	0,694	0,041	0,040
12	0,049	0,305	0,041	0,041

Phase *1*: Periods 1-7500 Phase *J*: 142501-150000

Taxation in WASIM

Average price variance per simulation run:

k	With Tax	No Tax	k	With Tax	No Tax
1	6019.01	118003.07	14	4845.68	132235.36
2	7329.48	104151.73	15	4739.02	69911.94
3	5409.49	79389.98	16	5881.87	72228.12
4	6674.24	101820.32	17	4982.83	117911.64
5	5196.16	60403.43	18	4601.88	77576.66
6	6491.39	67626.24	19	4921.87	80197.59
7	5942.82	81111.87	20	6613.29	50803.49
8	7817.09	59976.77	21	5836.15	25645.55
9	5790.44	48700.65	22	6918.05	52342.53
10	4769.49	94719.41	23	5196.16	59428.20
11	5485.68	152669.52	24	4739.02	58833.92
12	7451.38	46369.23	25	7695.19	122269.71
13	5759.96	55847.27			

$$F = \frac{\max \sigma^2_{tax}}{\min \sigma^2_{notax}} = \frac{25645.55}{7817.09} = 3.28$$

Critical value at 99% confidence level: 2.659

Taxation reduces market price volatilty

Conclusion (1/3) – In general

Investigation of the wealth of agents and the arising phenomena is shown, where each agent can change his strategy each period.

WASIM produces phases of over- and undervaluation even with bit-neutral mutation operator

Redistribution of wealth avoids concentrations, i.e. Tobin tax leads to a stabilizing effect

Conclusion (2/3) – Extending SF-ASM

The Walrasian Simulation Market (WASIM):

- Builds causality between wealth of agents and price setting
- Substitutes equilibrium model of SF-ASM by a Walrasian auctioneer
- Iterates to equilibrium prices even with trading restrictions
- Uses a bit-neutral mutation operator

Conclusion (3/3) – Market design

Design objectives for articial stock markets

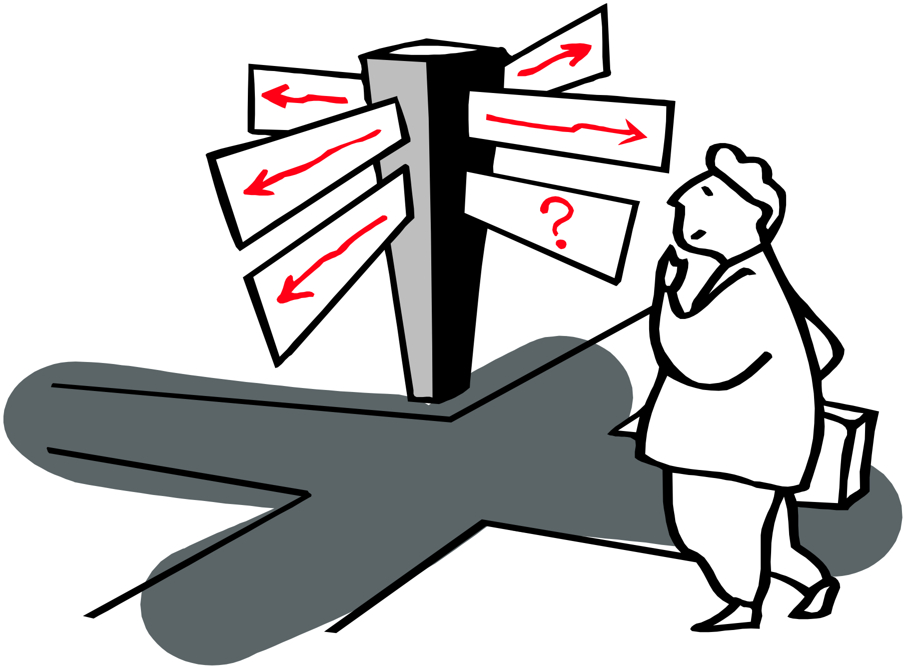
The underlying model has many parameters that influence the dynamics:

- Agents explore a fairly wide range of possible forecasting rules
  - ➔ Active rules permanently can change
- The genetic algorithm produces new rules
- The Walrasian auctioneer combines an iteration method towards equilibrium prices with the available wealth of agents

References

1. Cont, R. (2001) Empirical Properties of Asset Returns: Stylized Facts and statistical Issues, Quantitative Finance 1: 223--236.
2. Mandelbrot B (1963). The Variation of Certain Speculative Prices. Journal of Business 36: 394-419
3. LeBaron B (2002) Building the Santa Fe Artificial Stock Market. Working Paper, Brandeis University
4. Stuempert and Seese (2003) Influence of Heterogeneous Agents on Market Structure in an Artificial Stock Market. WEHIA 2003
5. Ehrentreich (2002) A Corrected Cersion of the Santa Fe Institute Artificial Stock Market Model. Complexity 2003:
6. Marks (2006) Market Design using Agent-based Models, in K.L. Judd, and L. Tesfatsion (eds.) Handbook of Computational Economics, Volume 2

Questions?



Articial Stock Markets (1/2) – Stylized facts

Financial data and observed empirical phenomena (Mandelbrot, Cont)

- High volatilty of market prices, which cannot be explained by fundamental data
- Volatility Clustering and increasing volatility causes decreasing prices
- Histogram of returns is leptokurtic (fat tails)
- Self-Similarity of market prices
- Deviation from fundamental prices over long time is possible

Taxation in WASIM

Seed-no	ρ_1^{notax}	ρ_J^{notax}	ρ_1^{tax}	ρ_J^{tax}
13	0,053	0,499	0,041	0,041
14	0,091	0,540	0,042	0,040
15	0,171	0,679	0,041	0,041
16	0,051	0,783	0,041	0,041
17	0,072	0,850	0,042	0,040
18	0,199	0,257	0,041	0,040
19	0,114	0,410	0,041	0,041
20	0,049	0,804	0,042	0,041
21	0,168	0,772	0,041	0,041
22	0,211	0,451	0,041	0,040
23	0,100	0,737	0,041	0,040
24	0,185	0,418	0,041	0,041
25	0,121	0,707	0,041	0,041

Microscopic reasons for high fluctuations

The Walrasian auctioneer establishes a relationship between stock prices and the wealth of agents

- Some agents who bought only the risky asset go bankrupt
- Bankrupt agents are replaced with randomly initialized new agents
- Good forecasting rules of the remaining agents are punished due to the crash
- Rarely used and extreme forecasting rules get a higher weight

➔ That causes agents to trade more randomly than relying on established strategies