Computers and Economic Democracy

UNIVERSITY
of
GLASGOW

Faraday Partnerships

Introduction

- I will be looking at the extent to which computing technology has improved the possibilities for planned economies.
- I am a reader in Computing Science at the University of Glasgow, and co-author of ‘Towards a New Socialism’

Topics of Discussion

- Plans and computers
- Value and prices under Socialism
- Wages
- Tax under Socialism
- Economic and direct democracy

Historical Background

- Immediate - the work of Prof Nove of this university and its impact in Britain
- Long term - the work of the Austrian capital theorists, particularly von Mises and Hayek
- Current relevance - application of Hayekian economics to formerly planned economies
  - Collapse of of production
  - Drastic fall in living standards and life expectancy (about 10 years down)

Plans and computers

- Starting with Von Mises, conservative economists argued that effective socialist planning was impossible because:
- No – effective cost metric in absence of market
- Complexity too great – millions of equations argument.

No Cost Metric

- Von Mises argued that without a market one could not cost things and thus had no rational basis for deciding between production alternatives.
- One exception he allowed was the use of Labour Values – we will return to this

Millions of equations

- Computers obviously change this as they can solve millions of equations
- Need to be quite precise about how many million equations and just how hard they are to solve
- This is a branch of complexity theory

Complexity

- The complexity of an algorithm is measured by the number of instructions used to compute it as the size of a problem grows.
- We will look at a simple example before going on to economic planning

Searching

- Suppose that I have a telephone directory for Quito and a phone number.
- It is clearly possible in principle to look at every number in the directory until I find who the number belonged to.
- The task would probably take several days.

Indexing

- If I have a name on the other hand, I can probably look up the phone number in less than 60 seconds.
- The complexity of looking up a name is of order *n*, or **O***n*, for a directory with *n* names in it.
- The complexity of looking up a number is of order Log(n)

Example

- Suppose I have 2 directories

1. Has 1000 entries
2. Has 1,000,000 entries

To look up a name will take 1000 times as long in the second directory, but to look up a number – given the name will only take twice as long.

I/O table

	rubber	steel	oil	zinc	cotton
rubber					
steel					
oil					
zinc					
cotton					
labour					
outputs					

Use of I/O table

- From the I/O table one can compute how much of each intermediate product required to produce each final product.
- In particular we can compute the labour content of each output.

Computability of labour content

- Suppose we have 10,000,000 different types of goods produced in an economy (Nove quotes this)
- Labour content given by the equation
- $\lambda = A\lambda + l$
- Where λ is a vector of labour contents, l a vector of direct labour inputs and A an input output matrix
- Clearly too big to invert, matrix is even too big to store in a computer containing : 10^{14} cells.

Gaussian solution impossible

products	multiplications	Seconds taken	
		uniprocessor	multiprocessor
1000	1,000,000,000	10	0.1
100,000	10^{15}	10^{7}	100,000
10,000,000	10^{21}	10^{13}	10^{11}sec=3000yrs

Simplification

- Matrix is sparse, most elements are zero
- Replace by linked list representation, we estimate the number of inputs directly used in a product is logarithmic in the size of the economy.
- Solve iteratively - use about 10 iterations,
- Complexity of order $n\text{Log}n$ in number of products. We estimate that it takes a few minutes on a modern machine.

Sparse representation

- Each production process represented by a list of pairs (input code, quantity)
- On average a process can then be represented in about 100 cells instead of 10,000,000

Iterative solution

- We only need to know labour values to about 3 significant figures.
- Initially just include direct labour inputs.
- The produce second estimate taking into account indirect inputs. Repeat this step about 10 times.
- You end up with a figure accurate to about 3 digits.

Iterative solution feasible

products	multiplications	Seconds taken	
		uniprocessor	multiprocessor
1000	150,000	0.0016	0.000016
100,000	100,000,000	1	0.01
10,000,000	$6x10^{10}$	600	6

Feedback mechanism

- We assume a real time feedback mechanism which uses sales of products along with democratically determined general goals to set net output targets for all goods. The planning computers must derive the gross outputs required to meed these net outputs.

Model we propose

Drawn on the principles of Robert Owen, the founder of New Lanark

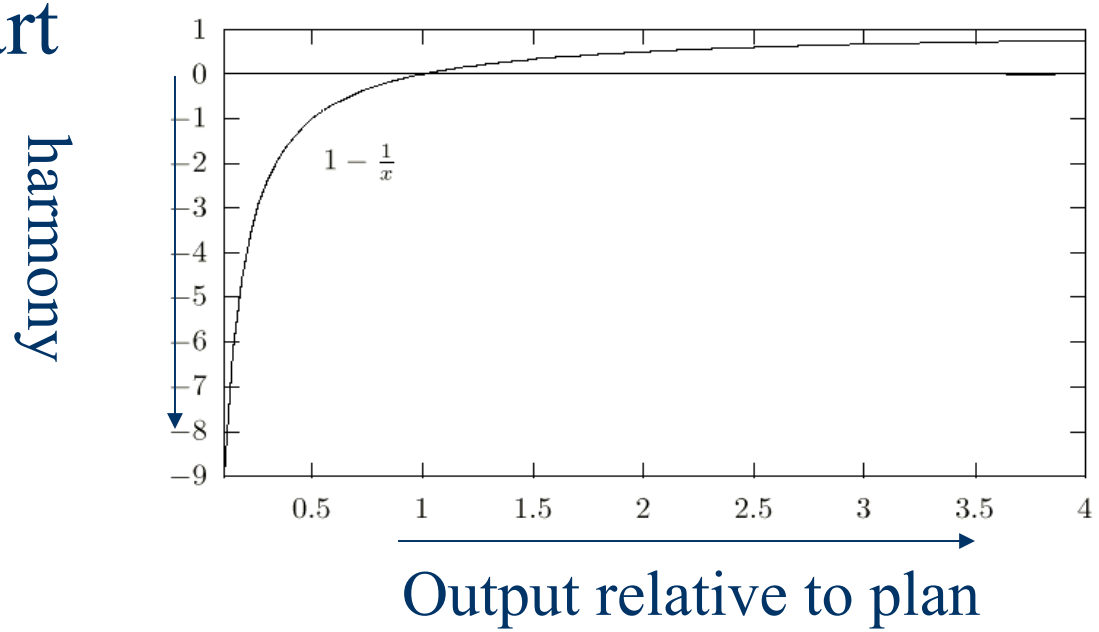
- Industry publicly owned and planned in physical units.
- Employees paid in labour tokens, 1 per hour.
- Goods priced in labour tokens proportional to the labour required to make them. (some discounting possible)

Market clearing prices used for finished goods

- If stocks of unsold goods grow – then reduce selling price
- If stocks fall – then increase selling price
- If price above labour value - then increase output
- If price below labour value – then reduce output

Maximise Harmony function

- Where x=output/target output
- Algorithm based on the Harmony function of Rumelhart



Outline of algorithm

- Algorithm again feasible in log-linear time.
- Based on iterative adjustment of allocation of stocks between different production activities guided by the derivative of the harmony function for each industry.

Why computers better than markets

- The market can be viewed as computing engine - this is explicit in Hayek.
- Cycle time is slow, measured in months or years.
- Arrives at answer by physically adjusting production up or down.
- Constantly tends to overshoot in an unstable way.
- Human costs to these adjustments- poverty and unemployment

Computers are faster

- Computers can predict where an ideal market economy would get to if it ever had the chance.
- Production can then be adjusted directly to this target.
- Cycle time for computation is in the order of hours not years or months.

Computers and democratic control

- We propose system of online electronic voting on key issues like the proportion of national income to be allocated to health, eduction, research etc.
- This done in terms of the fraction of the working week in labour units that is to go on it.
- Taxes automatically adjusted to the democratic vote on social labour allocation.

Payment

- Payment assumed to be 1 hour per hour worked minus taxes.
- No differentials for different grades of labour.
- Enterprises charged more by the state for skilled labour since this costs more to educate.
- Prevent accumulation of human capital but ensures efficient use of scarce labour.

No subsidies for essentials

- Soviet system subsidised bread, housing etc from profits of state industry.
- Wages underestimated the value of labour power
- This generated no incentive to economise on labour and as such led to inefficiency.
- Paying full value of labour makes subsidies unnecessary.

Incentives

- Would there still be an incentive to aquire skills
- Yes – because skilled work is more interesting and enjoyable than unskilled work even aside from payment questions.
- Equal pay is fundamentally democratic.

References

- Towards A New Socialism, Cockshott and Cottrell, Spokesman, available from Amazon, pdf version from my web page.
- A number of related papers from my web pages.
- http://reality.gn.apc.org
- http://www.gn.apc.org/Reality