COMPUTER SIMULATIONS OF PARTICIPATORY PLANNING[1] (4516)

The “model” of a participatory economy has been presented and discussed in several books and a number of journal articles over the past thirty years. One objection critics have raised is that the participatory annual planning procedure proposed to replace both central planning and markets may be impractical because it would require worker and consumer councils and federations to engage in too many rounds, or “iterations” of proposals and revisions. Critics have pointed out that while we may have proved that the iterative planning procedure will eventually converge to a feasible plan which is efficient under assumptions that are less restrictive than those necessary to prove that a market economy will reach an efficient equilibrium;[2] if this takes too many iterations it might be impractical because it simply takes up too much of participants’ time and energy. We can now report on what computer simulations of the annual participatory planning procedure suggest about its practicality.[3] We do not claim our results are yet definitive, and are in the process of carrying out more simulation “experiments.” However, even though the results reported here are preliminary, we believe they suggest that concerns about the practicality of our annual participatory planning procedure appear to be over exaggerated.

Platforms

We wrote an early version for our simulations in the computer programming language Netlogo -- “a multiagent programmable modeling environment.” For our purposes, the “environment” being modeled was the annual participatory planning procedure, and the worker councils and consumer councils were the agents in the system. Netlogo has a number of advantages. However, for all its strengths

---

[1] The authors thank the Systems Science Program at Portland State University under the leadership of Wayne Wakeland for supporting this research. We also thank Colm Massey and the Institute for Solidarity Economics in Oxford, England, who provided financial support without which early work on developing a Netlogo prototype of the planning procedure would not have been possible.

[2] See chapter 7 in Hahnel 2021 for the most recent proof of this proposition.

[3] See chapter 9 in Hahnel 2021 for a more extensive discussion of our simulation results.

Netlogo has some key limitations. Most importantly Netlogo would not easily scale up the number of goods nor the number of worker and consumer councils to approximate the size, scale, and complexity of a real economy. For these and other reasons we adapted the code for Netlogo into that of a general-purpose programming language, Clojure.[4] Unlike Netlogo, Clojure is able to leverage the Java API and hence utilize the Java ecosystem -- all the code written in and for Java developed and tested over more than twenty years, that is at its disposal. Clojure has other advantages as well, but most importantly for our purposes it allowed us to increase the scale of the economy to approximate the size of real national economies. Clojure also has a variant called Clojurescript, which compiles Clojure code into Javascript for the web.[5] Thus, Clojure code can easily and robustly compile for use and deployment publicly on a web server or locally with a web browser. In fact, much of our later work exploring the participatory planning procedure came courtesy of a Clojurescript app.

The simulations reported here were for an economy with 30,000 consumer councils, 30,000 worker councils, 100 different private consumption goods, 100 different intermediate goods, 100 different public goods, 100 different capital goods, 100 different categories of labor, and 100 different non-produced inputs from the natural environment. Assuming 1000 people in a consumer council, 30,000 CCs represents a population of 30 million people – which is three times larger than the populations of Sweden, Austria, Portugal, or Cuba; approximately the same size as the populations of Venezuela, Peru, Poland, Canada, or Australia; and roughly half the population of the United Kingdom, France, Italy, or Thailand. In all cases we simulated what might happen during annual participatory planning forty times in order to get enough data points to draw statistically meaningful conclusions.[6]

---

[4] https://www.clojure.org.

[5] https://www.clojurescript.org.

[6] The Clojure and Clojurescript source code for the algorithm is available at https://github.com/msszczep/pequod-cljs. The data for the experiments can be downloaded as gzip compressed files from http://www.szcz.org/depexperiments. A copy of the original Netlogo code and model is available at https://github.com/msszczep/pequod2.

The Algorithm

In our Clojure program we load in a set of worker councils (WCs) and a set of consumer councils (CCs) as key-value pairs, which Clojure refers to as a “map.”

Each Worker Council produces a single good and decides how much output to supply and how much of every input to request by solving an optimization problem in which it minimizes the disutility of its members’ from their work “effort,” while being rewarded for producing outputs whose social value exceeds the cost to society of using the inputs needed to produce them. Each WC has a Cobb Douglas production function, $z = q\mathbf{x}^{\mathbf{a}}\mathbf{r}^{\mathbf{b}}\mathbf{l}^{\mathbf{c}}e^{d}$, where output, z, is a function of some number of intermediate goods, **x**, natural resources, **r**, kinds of labor, **l** -- all chosen randomly from lists of each category -- and of the “effort” level of its members, e. Every WC requests a vector of intermediate inputs, a vector of raw materials from the natural environment, and a vector of hours of different kinds of labor, and decides how much effort to exert to produce its output so as to maximize its members’ wellbeing, WB_{WC}, which is a positive function of their income and a negative function of their disutility from work:

$$\text{Maximize } WB_{WC}\,(\mathbf{x}, \mathbf{r}, \mathbf{l}, e) = \{p_z q\mathbf{x}^{\mathbf{a}}\mathbf{r}^{\mathbf{b}}\mathbf{l}^{\mathbf{c}}e^{d} - [\mathbf{p}_x\mathbf{x} + \mathbf{p}_r\mathbf{r} + \mathbf{p}_l\mathbf{l}]\} - e^{f}$$

When members of a WC exert more effort there are two effects: (1) It increases output, and therefore the first term in their wellbeing function in braces which is an estimate of the size of the net social benefit they create, and therefore the average income WC members will be awarded. (2) It also increases the second term in their wellbeing function because it increases their disutility from greater effort expended in work. What makes WCs different from one another in our simulations is that they have different production functions, i.e. different values for q, **a**, **b**, **c**, and d, and they have different preferences for income vs. disutility from work, f. We give WCs different production functions by randomly assigning them different values for q, **a**, **b**, **c**, and d where the sum of the exponents range from .80 to .95, so that initially all WCs have decreasing returns to scale. We give WCs different tradeoffs between income and disutility from work by randomly assigning them different values for f between 3 and 4.

Each Consumer Council maximizes a Cobb Douglas wellbeing function, WB_{CC}, whose arguments are the different private goods it consumes, $\mathbf{y_1}$, and the different public goods it consumes, $\mathbf{y_2}$. CCs maximize their wellbeing subject to the constraint that the social cost of their consumption must be equal to their income, I_{CC}, which is determined by the average effort ratings and allowances of their members.[7]

$$\text{Maximize } z = WB_{CC}(\mathbf{y_1}, \mathbf{y_2}) = \mathbf{y_1}^{\boldsymbol{\alpha}}\mathbf{y_2}^{\boldsymbol{\beta}} \text{ subject to } \mathbf{p_{y1}}\mathbf{y_1} + \mathbf{p_{y2}}\mathbf{y_2} = I_{CC}$$

We randomly assign different CCs values for the 100 exponents for the different private consumption goods, $\boldsymbol{\alpha}$, and the 100 exponents for the different public consumption goods, $\boldsymbol{\beta}$, where each exponent varies between 0.025 and 0.05 so the sum of the exponents never exceeds 1. In a real world version of a participatory economy consumer council income, I_{CC}, would be determined by the average effort ratings and allowances of its members. However, in our simulations we simply assume that all CCs have the same number of members, the same average effort rating and allowances, and therefore the same income.[8]

After the WCs and CCs for a given experiment have been loaded into memory, along with the current vector of "indicative prices" for all goods, natural resources, and categories of labor, the algorithm proceeds as follows:

1. For each CC: Solve its optimization problem to update its demands for private goods and its demands for public goods based on the latest indicative prices.
2. For each WC: Solve its optimization problem to update its output level, effort level, and demands for all inputs based on the latest indicative prices.

---

[7] The social benefits of WCs producing a public good are calculated by multiplying the amount they propose to supply times the full indicative price for the public good in any round of the planning procedure. However, since all CCs will consume the public good "collectively," each CC is charged only for its proportionate share of the indicative price the WCs which produce the public good are credited for.

[8] We use the letter z to represent the output produced by any WC, and assume each WC produces only one such output. Therefore, the vector $\mathbf{z}$ contains *all* produced goods -- which include all intermediate goods, $\mathbf{x}$, all private consumption goods, $\mathbf{y_1}$, and all public goods, $\mathbf{y_2}$.

3. Calculate the new aggregate excess demands for all goods, natural resources, and categories of labor, and apply the price adjustment formula being tested to update all indicative prices.
4. Increase the iteration counter by one.
5. Check all excess demands to see if they fall within the specified threshold. If the excess demand for *every* good is within the threshold, stop. If not, return to step one above and repeat.

Practicality: How Many Iterations Will it Take?

Sensible people do not want to spend endless time submitting, revising, and resubmitting proposals for councils where they work, and neighborhood councils where they live. If all this can be done in a reasonably expeditious way, that is well and good. But what if it cannot? Three different considerations bear on how many iterations will be required to reach a feasible annual plan.

1. How far do we have to go to reduce excess demands and supplies before we can stop and launch a plan for the year? Or, put differently, what is a reasonable *threshold* for excess demands and supplies?
2. How efficient is our price adjustment rule?
3. What initial price vector is it reasonable to use in simulations of annual planning?

Threshold

A feasible plan means there is *no* excess demand for anything. However, in the real world a plan to eliminate excess supplies and demands *completely* is not required. After all, what we are talking about is having a comprehensive production/consumption plan for a year ready to go before the year begins. But as soon as the year begins we will discover that something has changed, and therefore adjustments will have to be made. So at some point continuing to do additional rounds of the planning procedure to eliminate every last drop of excess demand for every final, intermediate, and capital good, and for every category of labor and every input from the natural environment before we stop, is not worth the extra time and energy it would take. Once we had settled on a price adjustment rule, as explained below, we experimented to see how different thresholds affect the number of iterations required.

Table 1: Results for a 5% threshold

Table 1 lists the number of iterations it took in 40 different "experiments," or trial runs, to reach a 5% threshold from an initial price vector where all prices were arbitrarily set equal to 700.

Exp.	#I	Exp.	#I	Exp.	#I	Exp.	#I
1	12	**11**	12	**21**	12	**31**	11
2	12	**12**	11	**22**	12	**32**	11
3	12	**13**	12	**23**	12	**33**	12
4	12	**14**	12	**24**	12	**34**	12
5	12	**15**	12	**25**	12	**35**	12
6	12	**16**	12	**26**	12	**36**	11
7	12	**17**	12	**27**	12	**37**	13
8	11	**18**	12	**28**	12	**38**	12
9	11	**19**	12	**29**	11	**39**	12
10	13	**20**	12	**30**	11	**40**	12

As readers can see, beginning from an arbitrary initial price vector, the most iterations it took was 13 and the least it took was 11, and on average it took 11.85 iterations until the excess demand or supply for every good was 5% or less. When we continued the same 40 experiments to reach a 3% threshold or less the most iterations it took was 22, the least it took was 18, and on average it took 19.2

iterations for excess demand or supply for every good to be reduced to 3% or less, which is 7.35 more iterations than the 11.85 iterations it took on average to reach the 5% threshold. We chose the 5% threshold for most of the experiments reported on in the remainder of this article.[9]

Price Adjustment Rule

Our annual participatory planning procedure requires the Iteration Facilitation Board (IFB) to adjust estimates of the opportunity costs of using different categories of labor, different inputs from the natural environment, and different capital goods, and the social costs of producing different final goods, intermediate goods, and capital goods in each round of the planning procedure to reduce excess demands and supplies. We explored different price adjustment rules searching for a rule that was efficient, i.e. a rule that minimized the number of iterations required. After experimenting with different functional forms we settled on $w = v\{k - u^v\}$, where w is the percentage change we make in the price of a good for the next iteration in the planning procedure, %ΔP, and v is the percentage excess supply for the good in the iteration just completed, %ExS. However, for any $v > .25$ we substituted .25 for v where it first appears in the price adjustment formula, but not where it appears as an exponent. After experimentation with different values for k and u we settled on $k = 1.05$ and $u = 0.5$ as the values which seem to reduce the number of iterations required to reach our threshold as well as any others. In sum, all results we report on here are for the price adjustment rule: $w = v\{1.05 - (0.5)^v\}$, with the *proviso* mentioned above when v exceeds .25.

Initial Prices

In our experiments with different thresholds we began with every price set to 700. However, since an arbitrary initial price vector, or one that is chosen randomly, will most likely be very different from the final price vector that reduces all excess

---

[9] To be clear, what a 5% threshold means is we continue until the excess demand for *each and every* produced good, category of labor, and type of natural resource is less than 5%. Which means for the results we report, the excess demand for *most* things in the plans we settle on and would use to start a year is less than 5%.

supplies and demands to less than 5%, such a starting point will obviously require more iterations than starting from an initial price vector which is closer to where we end up. *Moreover, when real world annual planning begins a better candidate for initial price vector for any year is obvious. Namely, the final indicative price vector from the previous year.* Considering that changes in technologies, consumer preferences, and supplies of natural resources, labor, and capital goods will only change so much from year to year, any real world version of annual participatory planning would be far better off beginning with last year's final price vector than an arbitrary price vector, or one generated randomly. The question was how to model this in our simulations to see how much it is likely to reduce the number of iterations required.

We can specify technologies, preferences, and supplies of labor, natural resources, and capital goods, and run our simulation program to find the indicative price vector that yields a feasible plan, and take that price vector as last year's final price vector. The problem, however, is how to simulate *how much* economic conditions might reasonably change in the subsequent year. Fortunately there are two different ways in which simulation experiments can shed light on this question, which provide two, not just one window into how much beginning with last year's final price vector might reduce the number of iterations required in the real world.

Changing Exponents in Production and Wellbeing Functions: In our simulations we use Cobb Douglas production functions. Therefore the most natural way to model changes in technology is to change the exponents for different inputs in those production functions. We can model an *overall* improvement in technology by increasing the sum of the exponents in any production function. And we can model *variations* in how technologies change for different WCs and different industries by increasing individual exponents in a production function to different degrees.

In our simulated economy output is the product of a number of inputs, each with an exponent, where the number of inputs is not the same for each WC, all exponents are between zero and one, and for any individual WC the sum of its exponents is between .80 and .95. We can simulate *uneven* technological *progress* among WCs by increasing their exponents by different amounts chosen randomly. The problem was how to change the exponents to simulate the amount of overall technological progress which typically occurs from one year to the next.

We tried two very different modifications: First we added one of the following amounts to each exponent in every WC production function [.0000, .0001, .0002, .0003, .0004] chosen randomly. Next we added one of the following amounts to each exponent in every production function [.000, .001, .002, .003, .004] again, chosen randomly. To see which simulation of "technological progress" was more in line with historical increases in economic productivity we calculated the percentage increase in real GDP in both scenarios. The first, smaller adjustments in exponents in our Cobb Douglas production functions, yielded increases in real GDP in the range of 0.1% to 1.1%. The second, larger adjustments in exponents yielded increases in real GDP between 2.178% and 2.659%, with an average of 2.446%.

The results we report here are for the *larger* increases in exponents, which yield *larger* annual increases in real GDP than has normally occurred historically over the past forty years in most countries. Moreover, historical increases in real GDP are due not only to improvements in technologies, but also to increases in the supply of labor and other inputs. So, if anything, our simulation of technological change with the larger increases in exponents *over exaggerates* how much conditions of production typically change in most years in most countries. In which case, however many iterations it takes us to reach the 5% threshold should *overestimate* what would most likely be required.

We also use Cobb Douglas wellbeing functions for consumer councils in our simulations. Therefore we can model changes in consumer preferences by changing the exponents on the private and public goods in those wellbeing functions. In this case we simply increase some exponents and decrease other exponents to simulate changes in preferences. We added one of the following amounts to each exponent in a CC utility function [-.002, -.001, .000, +.001, +.002] chosen randomly.

Table 2: Changes in technologies and preferences

Exp.	#I	**GDP**	Exp.	#I	**GDP**	Exp.	#I	**GDP**	Exp.	#I	**GDP**
1	6	2.6%	**11**	7	2.577%	**21**	5	2.326%	**31**	7	2.211%
2	7	2.549%	**12**	5	2.554%	**22**	5	2.263%	**32**	7	2.534%
3	7	2.528%	**13**	7	2.609%	**23**	6	2.275%	**33**	7	2.373%
4	7	2.271%	**14**	7	2.609%	**24**	8	2.6%	**34**	6	2.21%
5	7	2.32%	**15**	5	2.218%	**25**	7	2.236%	**35**	8	2.264%
6	6	2.534%	**16**	5	2.567%	**26**	5	2.62%	**36**	7	2.558%
7	7	2.628%	**17**	7	2.551%	**27**	7	2.201%	**37**	5	2.659%
8	7	2.571%	**18**	7	2.227%	**28**	7	2.597%	**38**	6	2.239%
9	7	2.282%	**19**	7	2.282%	**29**	7	2.58%	**39**	5	2.603%
10	7	2.603%	**20**	7	2.649%	**30**	7	2.178%	**40**	7	2.587%

Table 2 lists the number of iterations it took to reach the 5% threshold when starting from the final prices from the previous year after making the changes to exponents in WC production functions and CC wellbeing functions described above in 40 different trial runs, or “experiments.” It also lists the annual rate of growth in real GDP in each experiment. As readers can see, beginning from the price vector from the previous year, it never took more than 8 iterations or less than 5, and on average it took only 6.575 iterations to reduce excess demand or supply for every good to 5% or less. The rate of growth of real GDP simulated ranged from a low of 2.178% to a high of 2.659%, and the average rate of growth of real GDP was 2.446%.

Tracking When Different Thresholds are Achieved: However, there is a second way to see how much beginning from last year's prices may reduce the number of iterations required. We were displaying in a Clojurescript app the excess supplies and demands for goods grouped into five categories which we could see after each iteration: (1) the excess demands and supplies for the private consumption goods were in one box, (2) the excess demands and supplies for the public consumption goods were in a second box, (3) the excess demands and supplies for the intermediate goods were in a third box, (4) the excess demands and supplies for the different categories of labor were in a fourth box, and (5) the excess demands and supplies for the different inputs from the natural environment were in a fifth box.

We color coded the boxes as follows: As long as the excess supply or demand for *any* of the goods in a box was still greater than 20%, the entire box was colored red. As soon as *all* excess demands and supplies in a category box fell below 20% the box color turned from red to orange. When all excess supplies and demands in a box fell below 10% the box turned from orange to yellow. When all excess supplies and demands in a box fell below 5% the box turned from yellow to green. And when all excess supplies and demands in a box fell below 3% the box turned from green to blue. We initially did this to get some idea how fast we were making progress in different phases of the convergence process during our search for a more efficient price adjustment rule. In particular we wanted to see if, when we were already close to meeting our threshold, it still required many iterations to get the excess supplies and demands for all goods under the threshold because our price rule was too aggressive at that point, and we were wasting planning time skipping back and forth from excess supplies to excess demands.

However, when we viewed our results we realized that if we believe that conditions only change by so much from year to year, what a real economy might go through would resemble *only* those iterations we were observing between when our boxes were changing from yellow to green for a 5% threshold, or from yellow to blue (for a 3% threshold.) In other words, we realized that since the IFB could always begin the annual participatory planning process with last year’s final indicative prices, no economy should have to go through the iterations that occurred when all our boxes were still red or orange. In any case, we report on the results of these experiments below in table 3. We think of this as a “second window” into *how much* beginning with last year’s final price vector might shorten the annual participatory planning process, as compared to the number of iterations it takes to reach the 5% threshold when we begin from an arbitrary initial price vector.

Table 3: From 10% excess supplies (Yellow) to 5% excess supplies (Green)

Exp.	#I	Exp.	#I	Exp.	#I	Exp.	#I
1	4	**11**	4	**21**	4	**31**	3
2	4	**12**	3	**22**	4	**32**	3
3	4	**13**	4	**23**	4	**33**	4
4	4	**14**	4	**24**	4	**34**	4
5	4	**15**	4	**25**	4	**35**	4
6	4	**16**	3	**26**	4	**36**	3
7	3	**17**	4	**27**	4	**37**	5
8	3	**18**	4	**28**	4	**38**	3
9	3	**19**	4	**29**	3	**39**	4
10	5	**20**	4	**30**	3	**40**	4

This table lists the number of iterations it took in 40 different trial runs, or “experiments,” to go from 10% excess demands (all yellow boxes), to 5% excess demands (all green boxes). On average it took only 3.77 iterations, and never took more than 5 iterations to reduce excess demands or supplies as large as 10% to excess demands or supplies to 5%.

Conclusion: A Practical Possibility?

To be frank, we were pleasantly surprised by our results. We did not expect our computer simulations to suggest that our annual participatory planning procedure was as practical as our reading of the preliminary evidence presented here suggests it may be. In particular:

- We did not expect to find a price adjustment rule as efficient as the one we have already found with very little experimentation or effort on our part. So presumably there may well be grounds for further improvement in this regard.

- When we simulated technological change by making random additions to the exponents in our WCs' Cobb-Douglas production functions, we were pleased to discover that additions to exponents which correspond to annual increases in real GDP which are, if anything, greater than historical increases on average due to technical change, only required a number of iterations to reach a feasible plan for the new year which seem quite "practical." And we were also pleasantly surprised that when we began from an arbitrary initial price vector the number of *additional* iterations required to move from excess demands in excess of 10% to excess demands less than 5%, which is also more in line with what would be required in the real world, yielded results very consistent with those findings.

- Because our early work was done in Netlogo which greatly limits the size of the economy one can study, we were uncertain how "scaling up" the size and complexity of the economy would affect convergence. However, we were pleased to discover that when we changed over to Clojure, increasing the number of WCs and the number of goods, inputs

> from nature, and kinds of labor to approximate the size of real economies, this did not appear to render participatory annual planning any less practical.

In sum, until there is further evidence; until there is a real world example of comprehensive participatory planning permitted to function for a number of years, in normal conditions and not under extreme duress; until there is a government willing to sanction a test run with real people in WCs and CCs; or until there is further evidence from more computer simulation experiments; based on the work done so far we believe our annual participatory planning procedure *can**not*** be summarily dismissed as a practical impossibility – a “bridge too far” -- as some have done.

References

Hahnel, R. 2021. *Democratic Economic Planning*. London, UK: Routledge.