Von Mises, Kantorovich and in-natura calculation

*W. Paul Cockshott**

The article reviews the idea of calculation in kind. It is argued that Kantorovich and subsequent mathematicians essentially validated the idea of in-kind calculation. This has not been evident because Kantorovich nowhere deals with the Austrian school and they for their part have ignored him. The article continues by examining improvements in linear optimisation since Kantorovich and the implications these have for economic planning. Finally it discusses the problem of deriving the plan ray in the context of markets for consumer goods.

JEL classifications: C61, B14
Keywords: planning, calculation debate, linear optimisation

1. Introduction

This paper presents a historical review and extended tutorial on the work of Kantorovich and his position with respect to the famous economic calculation debate. It focuses on Kantorovich because he is the most significant Soviet contributor to the question, and because his ideas are less well known to modern Western economists than those of the Austrian school. A Western readership is more likely to be familiar with neo-Classical or Sraffian approaches to economic calculation, so it is perhaps worth saying a little about how Kantorovich's approach will be seen to differ from these. We shall argue that in one sense Kantorovich's methods are a generalisation of those of Ricardo, but one aspect of Ricardo's work that Kantorovich shares, Ricardo's analysis of foreign trade, is not one that the modern neo-Ricardian school lays much emphasis on. In another aspect of his work though,

* University of Glasgow.

Correspondence Address:
W. Paul Cockshott, Dept. Computing Science, 18 Lilybank Gardens, Univerity of Glasgow, Glasgow, G12 8QQ.

Received 01 July 2008, accepted 10 March 2009

the use of linear production functions, Kantorovich has a certain amount in common with Sraffa, but with an important difference. Kantorovich assumes that there are multiple possible linear techniques, the optimal intensities of which have to be determined. The existence of a multiplicity of techniques, a combination of which will be used, means that the production frontier for Kantorovich is piecewise linear. This contrasts with the continuously curved production frontier assumed by the Cobb Douglas function typically used in neo-Classical work. At a micro level, at the level of detailed production planning, we know that what happens has to be strictly linear: the output of cars will be constrained by engine production in a linear rather than an exponential way. This implies that the stylised curved production frontier of neo-Classical theory is probably best seen as a conceptual approximation to a piecewise linear reality. We will start, however, by situating Kantorovich in relation to the Austrian school.

2. *What is economic calculation?*

In contemporary society the answer seems simple enough: economic calculation involves adding up costs in terms of money. By comparing money costs with money benefits one may arrive at a rational – wealth maximising – course of action.

In a famous paper (von Mises 1935) the Austrian economist von Mises argued that it was only in a market economy in which money and money prices existed, that this sort of economic rationality was possible.

His claims were striking, and, if they could be sustained, apparently devastating to the cause of socialism. The dominant Marxian conception of socialism involved the abolition of private property in the means of production and the abolition of money, but von Mises argued that

> »every step that takes us away from private ownership of the means of production and the use of money also takes us away from rational economics« (von Mises 1935: 104).

The planned economy of Marx and Engels would inevitably find itself »groping in the dark«, producing »the absurd output of a senseless apparatus« (von Mises 1935: 106). Marxists had counterposed rational planning to the alleged ›anarchy‹ of the market, but according to von Mises such claims were wholly baseless; rather, the abolition of market relations would destroy the only adequate basis for economic calculation, namely market prices. However well-meaning the socialist planners might be, they would simply lack any basis for taking sensible economic decisions: socialism was nothing other than the »abolition of rational economy«.

As regards the nature of economic rationality, it is clear that von Mises has in mind the problem of producing the maximum possible useful effect (satisfaction of wants) on the basis of a given set of economic resources. Alternatively, the problem may be stated in terms of its dual: how to choose the most efficient method of production in order to minimize the cost of producing a given useful effect. Von Mises repeatedly returns to the latter formulation in his critique of socialism, with the examples of building a railway or building a house:[1] How can the socialist planners calculate the least-cost method of achieving these objects?

As regards the means for rational decision-making, von Mises identifies three possible candidates:

1. Planning in kind (in natura).[2] This he rejects, and the validity of this rejection will be the main subject of this article.
2. Planning with the aid of an »objectively recognizable unit of value« independent of market prices and money, such as labour time. This too he rejects.
3. Economic calculation based on market prices.

It is clear that monetary calculation lends itself well to problems of the minimising or maximising sort. We can use money to find out which of several alternatives is cheaper, or which sale will yield us the most profit. But if we look in more detail at what is involved here, we shall see that a lot of calculation has to be done prior to the use of money. If an architect is planning a house, she must do a large amount of calculation in physical terms: estimating how much timber of each different type, how many bricks, how many tiles, window frames etc. will be required. Only once all the physical calculation has been done, once the bill of materials and the work schedules have been determined, then a costing can be done and presented to the client.

The architect would, in a capitalist economy, have prices of materials in mind when she chose them, but even in a capitalist economy price can not be the only factor. Actual availability of the supplies, lead time on delivery etc. are just as important. For an architect in a pre-capitalist economy, the designer of the Great Pyramid for example, these physical constraints would have been all that she had to go on. The architect in an earlier society would have done her calculations directly in terms of the available labour and natural resources, so such *in-natura* calculation has obviously been possible in the past. The question is whether it is still possible in modern societies with an extensive division of labour.

3. Planning in kind

The organisational task that faced a pyramid architect was vast. That it was possible without money was an indication that monetary calculation was not a *sine qua non* of calculation. But as the project being planned becomes more complex, then planning it in material units will become more complex. Von Mises is in effect arguing that optimization in complex systems necessarily involves arithmetic, in the form of the explicit maximization of a scalar objective

1 The railway example is in von Mises (1935). The house-building example is in von Mises (1949).

2 In talking about planning »in kind«, von Mises was responding to the proposals of Neurath (2004).

function (profit under capitalism being the paradigmatic case), and that maximising the money return on output, or minimising money cost of inputs is the only possible such scalar objective function. Von Mises argued for the impossibility of planning in kind because, he said, the human mind is limited in the degree of complexity that it can handle.

So might the employment of means other than a human mind make possible planning in kind for complex systems?

There are two »inhuman« systems to consider:

1. Bureacracies. A bureacracy is made up of individual humans, but by collaborating on information processing tasks, they can carry out tasks that are impossible to one individual.
2. Computer networks. Nobody familiar with the power of Google to consolidate and analyse information will need persuading that computers can handle volumes and complexities of information that would stupify a single human mind, so a computer network could clearly do economic calculations far beyond an individual human mind.

More generally as Turing pointed out (Turing 1937) any extensive calculation by human beings depends on artificial aides-memoir, papyrus, clay tablets, slates, etc. With the existence of such aides to memory, algorithmic calculation becomes possible, and at this point the difference between what can be calculated by a human using paper and pencil methods or a digital computer comes down only to matters of speed (Turing 1950 and 2004).

There is no question that the procedure of economic calculation considered by von Mises was primarily algorithmic. It involves a fixed process of:

1. For each possible technique of production
   a) form a physical bill of materials,
   b) use a price list to convert this into a list of money expenditures,
   c) then add up the list to form a final cost
2. Select the cheapest final cost out of all the costs of techniques of production

The question then arises as to whether there exist *in-natura* algorithms with an analogous function?

3.1 Kantorovich's method

In the 1920s and early 1930s when von Mises first advanced his arguments, no such algorithmic techniques were known. But in 1939 (Kantorovich 1960) the Soviet mathematician V. Kantorovich came up with a method which later came to be known as *linear programming* or *linear optimisation*, for which he was later awarded both Stalin and Nobel prizes. Describing his discovery he wrote:

> »I discovered that a whole range of problems of the most diverse character relating to the scientific organization of production (questions of the optimum distribution of the work of machines and mechanisms, the minimization of scrap, the best utiliza-

tion of raw materials and local materials, fuel, transportation, and so on) lead to the formulation of a single group of mathematical problems (extremal problems). These problems are not directly comparable to problems considered in mathematical analysis. It is more correct to say that they are formally similar, and even turn out to be formally very simple, but the process of solving them with which one is faced [i.e., by mathematical analysis] is practically completely unusable, since it requires the solution of tens of thousands or even millions of systems of equations for completion. I have succeeded in finding a comparatively simple general method of solving this group of problems which is applicable to all the problems I have mentioned, and is sufficiently simple and effective for their solution to be made completely achievable under practical conditions.« (Kantorovich 1960: 368)

What was significant about Kantorovich's work was that he showed that it was possible, starting out from a description in purely physical terms of the various production techniques available, to use a determinate mathematical procedure to determine which combination of techniques will best meet plan targets. He indirectly challenged von Mises,[3] both by proving that *in-natura* calculation is possible, and by showing that there can be a non monetary scalar objective function: the degree to which plan targets are met.

The practical problems with which he was concerned came up whilst working in the plywood industry. He wanted to determine the most effective way of utilising a set of machines to maximise output. Suppose we are making a final product that requires two components, an A and a B. Altogether these must be supplied in equal numbers. We also have three types of machines whose productivities are shown in the Table 1.

Table 1: Kantorovich's first example

Type of machine	# of machines	Output per machine		Total output	
		As	Bs	As	Bs
Milling machines	3	10	20	30	60
Turret lathes	3	20	30	60	90
Automatic turret lathes	1	30	80	30	80
Max total				120	230

Suppose we set each machine to produce equal numbers of As and Bs. The three milling machines can produce 30 As per hour or 60 Bs per hour. If the 3 machine produce As for 40 minutes in the hour and Bs for 20 minutes then they can produce 20 of each. Applying similar divisions of time we can produce 36 As and Bs on the Turret lathes and 21 As and Bs on the automatic turret lathe (see Table 2).

3 There is no indication that he was aware of von Mises at the time.

Table 2: Kantorovich's examples of output assignments

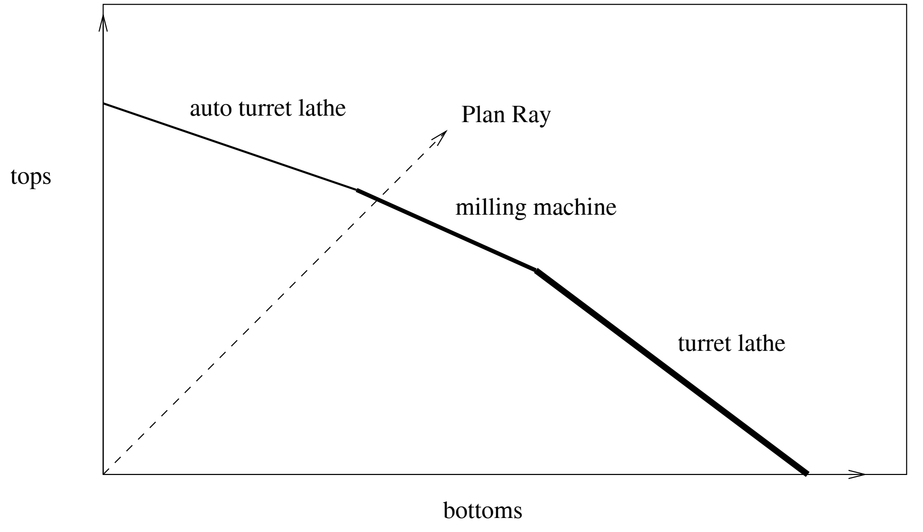
Type of machine	Simple solution		Best solution	
	As	Bs	As	Bs
Milling machines	20	20	26	6
Turret lathes	36	36	60	0
Automatic turret lathes	21	21	0	80
Max total	77	77	86	86

But Kantorovich goes on to show that this assignment of machines is not the best. If we assign the automatic lathe to producing only Bs, the turret lathe to producing only As and split the time of the milling machines so that they spend 6 minutes per hour producing Bs and the rest producing As, the total output per hour rises from 77 As and Bs to 86 As and Bs.

The key concept here is that each machine should be preferentially assigned to producing the part for which it is relatively most efficient. The relative efficiency of producing As/Bs of the three machines was milling machine = 1/2, turret lathes = 2/3, and automatic lathe = 3/8. Clearly the turret lathe is relatively most efficient at producing As, the automatic lath relatively most efficient at producing Bs and the milling machine stands in between. Thus the automatic lathe is set to produce only Bs, the turret lathes to make only As and the time of the milling machines is split so as to ensure that an equal number of each product is turned out.

The decision process is shown diagrammatically in Figure 1. The key to the construction of the diagram, and to the decision algorithm is to rank the machines in order of their relative productivities. If one does this, one obtains a convex polygon whose line segments represent the different machines. The slopes of the line segments are the relative productivities of the machines. One starts out on the left with the machine that is relatively best at producing Bs, then moves through the machines in descending order of relative productivity. Because relative productivity is monotonically decreasing one is guaranteed that the boundary will be convex. One then computes the intersection of the 45 degree line representing equal output of As and Bs with the boundary of this polygon. This intersection point is the optimal way of meeting the plan. The term linear programming stems from the fact that the production functions are represented by straight lines in the case of two products, planes for three products, and for the general higher dimensional case by linear functions. That is to say, functions in which variables only appear raised to the power one.

Figure 1: Kantorovich's example as a diagram



Note: The plan ray is the locus of all points where the output of As equals the output of Bs. The production possibility frontier is made of straight line segments whose slopes represent the relative productivities of the various machines for the two products. As a whole these make a polygon. The plan objective is best met where the plan ray intersects the boundary of this polygon.

The slope of the boundary where the plan ray intersects was called by Kantorovich the resolving ratio. Any machine whose slope is less than this should be assigned to produce Bs any machine whose slope is greater, should be assigned to produce As.

When there are only two products being considered, the method is easy and lends itself to diagramatic representation. But it can handle problems of higher dimensions, involving three or more products. In these cases we can not use graphical solutions, but Kantorovich provided an algorithmic by which the resolving ratios for different pairs of outputs could be arrived at by successive approximations. Kantorovich's work was unknown outside of the USSR until the late 1950s and prior to that Dantzig had independently developed a similar algorithm for solving linear programming problems, the so called simplex method (Dantzig/Wolfe 1961). This has subsequently been incorporated into freely available software tools.[4] These packages allow you to enter the problem as a set of linear equations or linear inequalities which they then solve. The constraints of the problem have to be expressed as a series of equations and the software package can then be treated as ›black box‹ to solve these equations. For Kantorovich's example problem we can express the production constraints as shown in Table 3.

4 For example lp_solve and GLPK.

Table 3: The constraints of Kantorovich's original problem expressed as equations

$m \le \begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix}$	Number of machines constraint
$m = \begin{bmatrix} \frac{1}{10} \\ \frac{1}{20} \\ \frac{1}{30} \end{bmatrix} x_a - \begin{bmatrix} \frac{1}{20} \\ \frac{1}{30} \\ \frac{1}{80} \end{bmatrix} x_b$	Productivity $x_a(i)$ number of As made on ith machine $x_b(i)$ number of Bs made on ith machine
$\sum x_a = A$	Total A production equals production on each machine
$\sum x_b = B$	Total B production equals production on each machine

Note: For the vectors above, index 1 means milling machines, 2 means turret lathes, and 3 automatic turret lathes.

In the West, linear programming was used to optimise the use of production facilities operating within a capitalist market. This meant that the objective function that was maximised was not a fixed mix of outputs, in Kantorovich's first example equal numbers of parts A and B, but the money that would be obtained from selling the output: price A × number of As + price B × number of Bs as expressed in Algorithm 1. Manuals and textbooks produced in association with Western linear programming software assumes this sort of objective. Thus one can get formulations which say that the task of linear programming is to maximise the objective function $f(x)$:

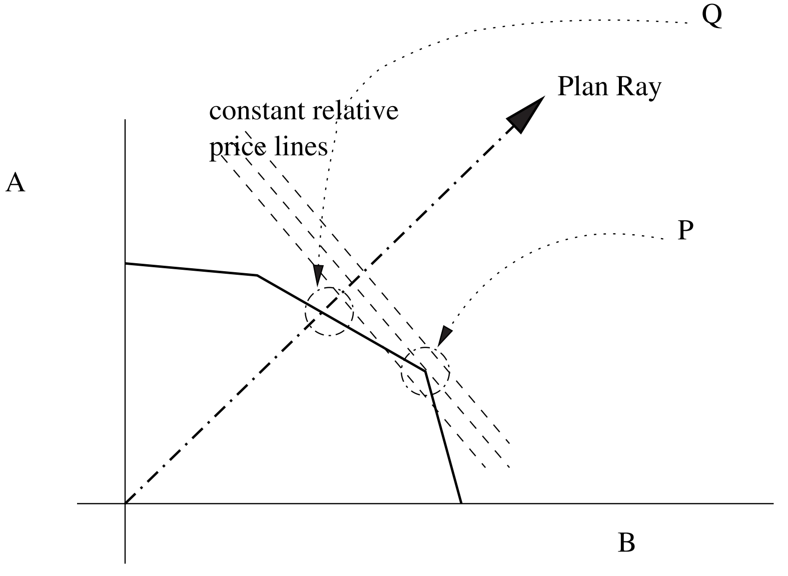
$$f(x) = c \cdot x,$$

where x is a vector of inputs or outputs, c is a unit cost or price vector. Maximisation is subject to the constraints

$$Ax \le b \text{ and } x \ge 0,$$

where A is a technology matrix and b a vector of available stocks. It is obvious that this formulation of linear programming is not *in-natura* calculation since it relies on the price vector c, readily available in a market economy, but which can not be assumed to exist in a planned economy. But the general formulation of linear programming that Kantorovich gives for the economy as a whole is an extension of the one he gave for his initial machine tool example. It again involves finding the intersection between the production possibility frontier given by the linear constraints, and a multi-dimensional plan ray. The difference between the two approaches is highlighted by Figure 2.

Figure 2: Comparison of the Western and Soviet versions of linear programming



Note: In the Western formulation the problem is to find P the maximal intersection of the production possibility frontier with lines of constant relative price for the outputs (A and B). In the Soviet formulation the problem is to find Q the intersection of the plan ray with the production possibility frontier.

In the Soviet formulation of the linear programming problem, there is no initial assumption of the existence of a set of relative prices. There is also a difference in where the solution point will occur. In the Western version of the problem the solution occurs at a vertex of the production possibility frontier, whereas in the Soviet formulation it occurs at a face of the frontier.

Let us first look at how one could apply modern software to solve a Western style linear programming problem with the same production constraints as those given by Kantorovich. If the factory he was dealing with faced prices such that an A sold for 100 roubles and a B sold for 60 roubles, we could express the problem faced by the factory manager as: Maximise $A + 0.6B$ subject to the production constraints in Table 3. In Algorithm 1 we show how this objective function and constraints can be expressed in the notation required by lp_solve.

To get Kantorovich's type of optimisation we replace maximising $A + 0.6B$ with just maximising A and add the constraint that

$$A - B = 0 .$$

Algorithm 1: Western factory facing Kantorovich's problem would formulate it as follows

```
/* Objective function */
max:A+0.6 B ;
/* Variable bounds */
m1<=3;m2<=3;m3<=1;
m1-0.1 x1a - 0.05 x1b=0;
m2-0.05 x2a - 0.033333 x2b=0;
m3- 0.033333 x3a - 0.0125 x3b=0;
x1a+x2a+x3a - A=0;
x1b+x2b+x3b -B =0;
int A;
```

Algorithm 2 shows how to express Kantorovich's problem in lp_solve notation. When this algorithm is run it exactly reproduces the solution originally given in Kantorovich (1960).

Algorithm 2: Kantorovich's example as equations input to lp_solve

```
/* Objective function */
max:A;
/* plan ray constraint */
A-B=0;
/* Variable bounds specified as in algorithm 1*/
```

3.2 Kantorovich and Ricardo

There is a strong parallel between the arguments that Kantorovich uses and those that Ricardo used in his *Principles* to explain the benefits of international trade. He constructed an argument to the effect that if it took Portugal 80 men to produce one unit of wine and it took England 120 men to do the same (Ricardo 1951: 135). On the other hand in Portugal it took 90 men to produce one unit of cloth but in England 100. Under these circumstances he said it was advantageous for England to export cloth to Portugal and import wine. The argument was that labour in each country should be used to produce what it is relatively best at. This can now be seen as a specific case of linear optimisation.

We can set up a plan ray requiring the production of equal quantities of wine and cloth for similarity with Kantorovich's example. We will also assume that both England and Portugal have 1 million workers.

We can then express Ricardo's example as a linear program:

```
max: wine;
/* Constraints */
wine -pwine -ewine = 0;
cloth -pcloth -ecloth = 0;
```

```
wine -cloth = 0;
-90 pcloth +pcl = 0;
-80 pwine +pwl = 0;
pcl +pwl <= 1000000;
-100 ecloth +ecl = 0;
-120 ewine +ewl = 0;
ecl +ewl <= 1000000;
```

Where the variables are:

`wine`: total wine production, `cloth` similarly
`pwine`: Portuguese wine production, `ewine`, `pcloth`, `ecloth` similarly
`ecl`: English cloth producing labour, `pcl`, `ewl`, `pwl` similarly

Solving for the equations with lp-solve (Table 4) we find a net production of both wine and cloth of 11,176 units. If we now prevent trade by forcing each country separately to equate its wine and cloth production (e.g. `pcloth=pwine`), we find that total production of each falls to 10,427 units, demonstrating that Ricardo was right: overall production turns out to be seven percent higher with trade between the two countries. But whilst we can say that Ricardo recognised a specific instance of linear optimisation, it was not until Kantorovich that a general mechanism for formulating economic problems in this way was arrived at.

Table 4: Solving Ricardo's problem with lp_solve

Variables	trade allowed	trade not allowed
objective	-11176.47	-10427.80
wine	11176.47	10427.80
pwine	11176.47	5882.35
ewine	0.00	4545.45
cloth	11176.47	10427.80
pcloth	1176.47	5882.35
ecloth	10000.00	4545.45
pcl	105882.35	529411.76
pwl	894117.64	470588.23
ecl	1000000.00	454545.45
ewl	0.00	545454.54

3.3 Generalising Kantorovich's approach

In his first example Kantorovich deals with a very simple problem, producing two goods in equal proportions using a small set of machines. He was aware, even in 1939 that the potential applications of mathematical planning were much wider. We will look at two issues that he considered which are important for the more general application of the method.

1. Producing outputs in a definite ratio rather than in strictly equal quantities.
2. Taking into account consumption of raw materials and other inputs.

Suppose that instead of wanting to produce one unit of A for every unit of B, as might be the case if we were matching car engines to car bodies, we want to produce four units of A for every unit of B, as would be the case if we were matching wheels to car engines. Can Kantorovich's method deal with this as well? Consider Figure 1 again. In that the plan ray is shown at an angle of 45 degrees a slope of one to one. If we drew the plan ray at a slope of four to one, the intersection with the production frontier would provide the solution. Since this geometric approach only works for two products, let us consider the algebraic implications.

You should now be convinced that it is possible to solve Kantorovich's original problem[5] by algebraic means. In Algorithm 2 we specified that $A - B = 0$ or in other words $A = B$, if one wanted four units of A for every B we would have to specify $A = 4B$ or, expressing it in the standard form used in linear optimisation, $A - 4B = 0$. Suppose A stands for engines, B stands for wheels. If we now say wheels come in packs of 4, then we can repose the problem in terms of producing equal numbers of packs of wheels and engines. Introduce a new variable $\beta = 4B$ to stand for packs of wheels, and rewrite the equations in terms of β and we can return to an equation specifying the output mix in the form $A - \beta = 0$, which we know to be soluble.

In order to use standard linear programming packages to solve a »Soviet type« problem with a plan ray and n products we introduce $n - 1$ additional constraints of the form $A - k_b B = 0$, $A - k_c C = 0$, $\ldots A - k_n N$ and maximise on A. The constants k_b, k_c, $\ldots k_n$ specify the ratios in which the goods are to be produced in the plan.

How do we deal with consumption of raw materials or intermediate products?

In our previous example we had variables like `x1b` which stood for the output of product B on machine 1. This was always a positive quantity. Suppose that there is a third good to be considered – electricity, and that each machine consumes electricity at a different rate depending on what it is turning out. Call electricity C and introduce new variables `x1ac`, `x1bc` etc. referring to how much electricity is consumed by machine 1 producing outputs A and B. Then add equations specifying how much electricity is consumed by each machine doing each task, and the model will specify the total amount of electricity consumed.

We now know how to…

1. …use Kantorovich's approach to specify that outputs must be produced in a definite ratio.
2. …use it to take into account consumption of raw materials and other inputs.

5 Actually this was his »problem A«

If we can do these two tasks, we can in principle perform *in-natura* calculations for an entire planned economy. Given a final output bundle of consumer and investment goods to maximise (the plan ray) and given our current resources, a system of linear equations and inequalities can be solved to yield the structure of the plan. From simple beginnings, optimising the output of plywood on different machines, Kantorovich had come up with a mathematical approach which could be extended to the problem of optimising the operation of the economy as a whole.

3.4 A second example

Let us consider a more complicated example, where we have to draw up a plan for a simple economy. We imagine an economy that produces three outputs: energy, food, and machines. The production uses labour, wind and river power, and two types of land: fertile valley land, and poorer highlands. If we build dams to tap hydro power, some fertile land is flooded. Wind power on the other hand, can be produced on hilly land without compromising its use for agriculture. We want to draw up a plan that will make the most rational use of our scarce resources of people, rivers and land.

In order to plan rationally, we must know what the composition of the final output is to be – Kantorovich's ray. For simplicity we will assume that final consumption is to be made up of food and energy, and that we want to consume these in the ratio three units of food per unit of energy. We also need to provide equations relating to the productivities of our various technologies and the total resources available to us. Valleys are more fertile. When we grow food in a valley, each valley requires 10,000 workers and 1,000 machines and 20,000 units of energy to produce 50,000 units of food. If we grow food on high land, then each area of high land produces only 20,000 units of food using 10,000 workers, 800 machines and 10,000 units of energy. Electricity can be produced in two ways. A dam produces 60,000 units of energy, using one valley and 100 workers and 80 machines. A windmill produces 500 units of electricity, using four workers and six machines, but the land on which it is sited can still be used for farming. We will assume that machine production uses 20 units of electricity and ten workers per machine produced. Finally we are constrained by the total workforce, which we shall assume to be 104,000 people.

Tables 5 and 6 show how to express the constraints on the economy and the plan in equational form. If we feed these into lp_solve we obtain the plan shown in Table 7. The equation solver shows that the plan targets can best be met by building no dams, generating all electricity using 541 windmills, and devoting the river valleys to agriculture.

It also shows how labour should be best allocated between activities: 40,000 people should be employed in agriculture in the valleys, 109 people should work as farmers in the highlands, 2,164 people should work on energy production, and 61,727 people should work building machines.

Table 5: Variables in the example economy

e	total energy output
e_c	household energy consumption
f	food
v	valleys
w	windmills
m	machines
d	dams
u	undammed valleys
h	highland
f_h	food produced on high land
f_v	food produced in valleys

Table 6: Resource constraints and productivities in our example economy

final output mix	$f = 3e_c$
number of valleys	$v = 4$
dams use valleys	$v - u = d$
valley food output	$f_v = 50{,}000u$
valley farm labour	$l_v = 10{,}000u$
valley energy use	$e_v = 20{,}000u$
valley farm machines	$m_v = 1{,}000u$
highland food output	$f_h = 20{,}000h$
highland farm labour	$l_h = 10{,}000h$
highland energy use	$e_h = 10{,}000h$
highland farm machines	$m_h = 800h$
energy production	$e = 500\mathrm{w} + 60{,}000d$
energy workers	$l_e = 100d + 4w$
machines in energy prod	$m_e = 80d + 6w$
workers making machines	$l_m = 10m$
energy used to make machines	$e_m = 20m$
energy consumption	$e_m + e_v + e_h + e_c \leq e$
machine use	$m_e + m_h + m_v \leq m$
total food prod	$f = f_h + f_v$
workforce	$l_m + l_e + l_v + l_h \leq 104{,}000$

Table 7: Economic plan for the example economy using lp_solve

d (dams)	0
e	270,500
f	200,218
h	0.0108889
m	6172.71
u	4
v	4
w (windmills)	541
e_c	66739.3
e_h	108.889
e_m	123,454
e_v	80,000
f_h	217.778
f_v	200,000
l_e	2,164
l_h	108.889
l_m	61727.1
l_v	40,000
m_e	2,164
m_h	8.71111
m_v	4,000

The results that we have obtained were by no means obvious at the outset. It was not initially clear that it would be better to use all the river valleys for agriculture rather than building dams on some of them. In fact, whether dams or windmills are preferred turns out to depend on the whole system, not just on their individual rates of producing electricity. We can illustrate this by considering what happens if we cut the labour supply in half to 52,000 people?

If we feed this constraint into the system of equations we find the optimal use of resources has changed. The plan now involves one dam and 159 windmills. Cut the working population slightly further, down to 50,000 people and the optimal plan involves flooding two valleys with dams and building only 23 windmills. Why?

As the population is reduced, there are no longer enough people available to both farm the valleys and produce agricultural machinery. Under these circumstances the higher fertility of lowland valleys is of no importance, it is better to use one or more of them to generate electricity. By applying Kantorovich's approach it becomes possible for a socialist plan to do two things that von Mises had believed impossible:

1. It allows the plan to take into account natural resource constraints – in this case the shortage of land in river valleys which can be put to alternative uses.
2. It allows rational choices to be made between different technologies – in this case between windmills and hydro power and between lowland and highland agriculture.

Contrary to what von Mises claimed, the whole calculation can be done in physical units without any recourse to money or to prices.

4. Valuation

The core of von Mises's argument relates to the use of prices to arrive at a rational use of intermediate or capital goods. Von Mises argues that, in practice, only money prices will do for this, but concedes that, in principle, other systems of valuation, such as labour values would also be applicable. Kantorovich too, was very concerned with the problem of relative valuation (Kantorovich 1965), and developed what he called *objectively determined valuations* (ODV).

He considered a situation where planners have to deal with several different types of factories (*A..E*) each able to produce products one and two, and where the intended ratio of output of product one and two are fixed in the plan. Each class of factory *A..E* has different relative productivities for the two products.

He next looked at the apparent profitability of producing products one and two under different relative valuations. Under some schemes of relative price, all factories would find product one to be unprofitable relative to product two, under other the reverse would occur. Intermediate price schemes would allow both products to be produced, with some classes of factories specializing on one and others on two. He gives the example of children's clothing as something which, under the administratively determined valuations then used in the USSR, were unprofitable to produce, and unless factories were specifically instructed to ignore local profitability, too few children's clothes would be made.

He asks if there exists a relative valuation structure which would allow factories to concentrate on the most valuable output, and at the same time meet the specified plan targets and arrives at certain conclusions:

1. That among the very large number of possible plans there is always an optimal one which maximises output of plan goals with current resources.
2. That in the optimal plan there exists a set of *objectively determined valuations* (ODV) of goods which will ensure that each factory
   a) produces the output which will contribute most to maximising the plan goals;
   b) each factory also finds that the output which contributes to maximising plan targets is also the output which is most profitable.
3. With arbitrary valuations which differ from ODV, these conditions can not be met, and profit maximising factories will not specialise in a way that meets plan goals optimally.

It is important to understand that his ODVs are valuations that apply only for a plan which optimally meets a specific plan target. Kantorovich's procedure for arriving at an optimal plan involved successive adjustments to the ODVs and factory specialisation until both the appropriate mix of goods is reached, and at the same time each factory is producing its most profitable good. He actually gave several different mathematical procedures for arriving at such a plan and system of ODVs. The ODVs basically specify the derivatives of the production possibility frontier in the region of its intersection with the plan ray.

Table 8: Example optimal plan with technology matrix, plan ray and net output

outputs	inputs				gross outputs
	labour	corn	machines	coal	
corn	$\frac{1}{3}$	$\frac{1}{10}$	$\frac{1}{20}$		1416.76
machines	2		$\frac{3}{20}$		93.43
coal	$\frac{1}{10}$		$\frac{1}{100}$		858.49
bread	$\frac{1}{5}$			$\frac{1}{10}$	1275.08

plan ray	
coal	bread
1	2

net output	
coal	bread
1275.08	637.54

labour force
1000

Let us use another example to explore the idea of ODVs.

Although Kantorovich asserts that labour is ultimately the only source of value, his ODVs are short term valuations and differ from the classical labour theory of value, which gave valuations in terms of the long term labour reproduction costs of goods – including the reproduction costs of capital goods. Kantorovich, in contrast, is concerned with valuations which should apply with the current stock of means of production and labour resources. For example, he considers the situation of giving a valuation to electric power relative to labour. Instead of valuing it in terms of the labour required to produce electricity, he first assumes that the total electrical power available is fixed – i.e., power-stations operating at full capacity, and then works out how many person hours of labour is saved by using an additional kilowatt hour of electricity. This definition of the value of electricity in terms of labour is clearly different from the way labour value was defined by the classical economists. In their formulation the labour value of a kilowatt hour, for example, was the mean labour expended to produce a kilowatt. One would expect the classical labour value to be lower than Kantorovich's labour ODV, since otherwise the use of electricity would not be worth while.

Nemchinov (1964: 373) criticised Kantorovich for raising what the former saw as just:

> »indices expressing deficiency, limitation, and scarcity of resources [...] applicable to the economic calculations involved in discovering how best to use resources to ensure maximum fullfillment of a production programme.«,

to concepts of a universal character. For this he was »gravely at fault«.

Kantorovich's insistence on considering short term, very material constraints – so many megawatts of power, such and such a number of cutting machines, etc., gives his work an intensely practical and pragmatic character, quite different from that of most theoretical economists.

Why is Kantorovich so concerned with valuations and profitability? There seem to be two reasons. We should first note that by profit maximising Kantorovich actually meant maximising the value of output. This must be understood in the context of Soviet practice where mines and factories were given incentives to over-fulfill plan targets. If the output was a single good – say coal, the target could be specified in tons. But if the factory produced several goods, then the target had to be set in terms of x rubles worth of a mix of goods. With the »wrong« price structure, plants would attempt to maximise the production of the goods which were of the highest value, ignoring those of lower value, with the result that the aggregate supply of all goods was often not in the proportions that the planners intended. This practice of setting plan targets in money terms reflected the limited ability of GOS PLAN to specify detailed targets in kind as described by Nove (1983). We have to understand that he was engaging in a wider debate during the 1950s about the appropriate pricing structure for a socialist economy. Thus Nove (1964) identifies three alternative reforms being proposed for Soviet prices: the suggestion of Strumlin that prices should correspond to labour values; that of Novozilov who argued for Marxian »production prices« (Novozilov 1964: 36); and that of Kantorovich (1965: 53 – 57) who proposed »objectively determined valuations«. Nove argues that Novozilov and Kantorovich were both trying to develop rational costing models.

The second reason relates to his particular algorithm for solving linear programming problems which used an iterative adjustment to initial ODVs (resolving multipliers) until an optimal plan is achieved.

These two aspects seem intimately linked in his presentation, but the presuppositions about the incentives to factories are not brought to the fore which owes something to the cryptic language in which economic debates in the USSR were conducted. Swann (1975) relates that some of the Soviet optimal planning school, he cites Volonsky, argued that the

ODVs were all that had to be exchanged between distinct units of production each following their own optimisation goal. In this his arguments for come close to those of the Austrian school with respect to prices. In retrospect though we can see that the use resolving multipliers, however much it influenced Kantorovich's own thought, is incidental to the issue. With computer algorithms, the process of solving a linear program becomes a »black box«. The user need not concern herself with details such as the method of calculation – whether it uses Kantorovich's approach Dantzig's or Karmarkar's, except insofar as this affects the size of problem that can be handled, as we discuss in Section 5. With computer packages, ODVs would no longer be needed for computing a plan, but would they still be needed for specifying targets to factories?

This depends on the information processing capacity of the planning system. If it were capable of specifying fully disaggregated plans, then it could in principle just place orders with factories for specific quantities of each good. In these circumstances, the factories could not cheat by producing more of high value items and less of low value ones. Indeed, the very information that would be required to compute Kantorovich's ODVs, would have been sufficient for GOS PLAN to specify disaggregated orders in kind for the products that would have had valuations attached. Thus were it possible to compute ODVs then they would have been redundant for the purpose that Kantorovich originally proposed them!

There remains another level at which valuations would have been useful – when product designs were being drawn up at a local level. If a refrigerator designer was deciding on what components to use in a planned new model, she would need some way of telling which components would, from a social standpoint, have been the most economical, which implies a system of valuations. However it is not clear that the full apparatus of ODVs would be either necessary or appropriate here. ODVs correspond to a system of marginal cost, rather than average cost pricing. They reflect current marginal costs with the immediately current constraints on production. The use of such marginal costing was criticized by other Soviet economists (Grossman 1963, Menshikov 2006).

It is not clear, in retrospect, that ODVs would have been more appropriate than a system of average cost valuation if one was projecting ahead a year or so. If one draws up example plans in which all goods are reproducible from scratch, then the ODVs will just be the same as labour values. If there are some other »free« inputs left over from the past, then the ODVs begin to deviate from labour values. So in the short term ODVs could be useful, but it is not clear that they are so useful in the preparation of long term plans. Indeed, given the stochastic properties of prices in a real capitalist economy (Farjoun/Machover 1983), it is questionable whether, with the exception of certain constrained products like oil, the difference between average and marginal costs is significant in the West.

5. Complexity

Linear programming, originally pioneered by Kantorovich, provides an answer in principle to von Mises' claim that rational economic calculation is impossible without money. But this is an answer only in principle. Linear programming would only be a practical solution to the problem if it were possible, in practice, to solve the equations required in a socialist plan. This in turn requires the existence of a practical algorithm for solving them, and sufficient computational resources to implement the algorithm. Kantorovich, in an appendix to Kantorovich (1960), gave a practical algorithm, to be executed by paper and pencil mathematics. The algorithm was sufficiently tractable for these techniques to be used to solve practical problems of a modest scale. When tackling larger problems he advised the use of approximative techniques like aggregating similar production processes and treating them as a single composite process. Whilst Kantorovich's algorithm uses his ODVs, which he earlier called resolving multipliers, subsequent algorithms for linear programming do not, so the ODVs should not be considered as fundamental to the field.

Since the pioneering work on linear programming in the 1930s, computing has been transformed from something done by human »computors« to something done by electronic ones. The speed at which calculations can be done has increased many billion-fold. It is now possible to use software packages to solve huge systems of linear equations. But are computers powerful enough to be used to plan an entire economy?

In a large economy like the former USSR there were probably several million distinct types of industrial products, ranging from the various sorts of screws, washers and types of electronic components to large final products like ships and airliners. Although there was great enthusiasm for Kantorovich's methods in the USSR during the 1960s, the scale of the economy was too great for his techniques to be used for detailed planning with the then available computer technology. Instead they were used either in optimising particular production plants, or, in drawing up aggregated sectoral plans for the economy as a whole.

How has the situation changed today, given that the power of computers has continued to grow at an exponential rate since the fall of the USSR? In other words to what complexity class (Sipser 2006, Part III) does linear programming belong?

> »For a long time it was not known whether or not linear programs belonged to a non-polynomial class called ›hard‹ (such as the one the traveling salesman problem belongs to) or to an ›easy‹ polynomial class (like the one that the shortest path problem belongs to). In 1970, Victor Klee and George Minty created an example that showed that the classical simplex algorithm would require an exponential number of steps to solve a worst-case linear program (Klee/Minty 1972). In 1978, the Russian mathematician L.G. Khachian developed a polynomial-time algorithm for solving linear programs (Khachian 1979). It is an interior method using ellipsoids inscribed in the feasible region. He proved that the computing time is guaranteed to be less that a polynomial expression in the dimensions of the problem and the number of digits of input data. Although polynomial, the bound he established turned out to be too high for his algorithm to be used to solve practical problems. Karmarkar's algorithm (Karmarkar 1984) was an important improvement on the theoretical result of Khachian that a linear program can be solved in polynomial time. Moreover his

> algorithm turned out to be one which could be used to solve practical linear programs.« (Dantzig 2002)

Modern linear programming packages tend to combine Dantzig's simplex method with the more recent interior point methods. This allows the most modern implementations to solve programming problems involving up to one billion variables (Gondzio/Grothey 2006a and 2006b). For such huge problems large parallel supercomputers with over a thousand processor chips are used. But even with much more modest 4 CPU computers, linear programming problems in the million variable class were being solved in half an hour using interior point methods.[6]

These advances in linear programming algorithms and in computer technology mean that linear programming could now be applied to detailed planning at the whole economy level, rather than just at an aggregate level.

6. Deriving the plan ray

Kantorovich assumed that the plan had a given target to optimise in the form of a particular mix of goods: the plan ray. This reflected the social reality for those engaged in managing Soviet industry, in that they were given a mix of products to produce by GOS PLAN. The planning authorities themselves however, needed to specify what this ultimate output mix would be. In the early phases of Soviet planning, when Kantorovich wrote his original paper, the goals set by the planners were primarily directed at achieving rapid industrialization and building up a defence base against the threat of invasion. The planning process was successful in achieving these goals. But in an already industrialised country, in times of peace, the meeting of current social needs becomes the first priority and so the plan vector has to be pointed in that direction. A criticism commonly levelled at the Soviet-type economies – and not only by their Western detractors – is that they were unresponsive to consumer demand. It is therefore important to our general argument to demonstrate that a planned economy can be responsive to the changing pattern of consumer preferences – that the shortages, queues and surpluses of unwanted goods of which we hear so much are not an inherent feature of socialist planning. The economists Dickinson and Lange, writing just prior to Kantorovich, outlined a practical mechanism by which this could be done (Lange 1938, Dickinson 1933).

They proposed that the state wholesale sector should operate on a break-even basis with flexible prices. Wholesale managers would set market clearing prices for the products on sale as consumer goods. These wholesale prices would then act as a guide to the plan authorities, telling them whether to increase or decrease production of particular lines of product. If prices were high, then that line of product would have its output increased, otherwise its planned output would be reduced.

6 See Bienstock (2002: Chap 4). The Harmony Algorithm for constructing plans, given in Cottrell/Cockshott (1992), is an instance of the class of algorithm discussed by Bienstock.

The basic idea is clear, the same principle that adjusts production of consumer goods in a capitalist economy was to be employed. But this then raises the problem of how one determines that a price is high or low. High or low relative to what? What would be the basis of valuation used? Although Marx and Engels had laid great stress on planning as an allocation of labour time, this conception had been more or less abandoned by English speaking socialist economists by the late 1930s. Neither Lange nor Dickinson relied on the classical theory of value in their arguments. Writing in 1930, Appel et al. (1990) had laid great stress on the relevance of the labour theory of value for socialist economics, but their ideas were largely ignored. More recent writers have again laid emphasis on Marx's theory of value as a guide to socialist planning (Dieterich 2002, Peters 1996 and Peters 2000). It has been argued that labour values are effectively calculable and that in combination with Dickinson's proposals for socialist markets they provide a pragmatic way of obtaining a plan ray that conforms to consumer demand (Cottrell/Cockshott 1992).

Whilst the task of determining the plan ray itself can be solved by Dickinson's method, determining its intercept with the production frontier remains problematic unless all the technical coefficients of production are available to the planning system. In this context the Austrian school has tended to emphasise the importance of tacit or private knowledge. They have argued that a key role of the price system is its ability to make public, data that was previously held privately by firms. The gist of the argument is that although firms may wish to keep private the technical details of production, they are forced by the market to make public such portions of the data as relate to their interaction with other firms through the prices that they bid for inputs. But this view of the price system as the principle channel of inter firm information is demonstrably wrong. Prices are only a small part of the information that is exchanged between firms. Details about quantities, specification of components, delivery times etc., all have to be exchanged between supplier and consumer. In quantity, this other information far outweighs the information content of the price that is finally agreed upon.[7] If you were to gather together the information of this sort that a firm communicates to its suppliers and to its customers, you would be able to reconstruct a pretty accurate linear model of the production processes it was undertaking. In a modern economy this information is already largely computerised. Orders are entered into purchasing systems that record the purchases in a database and typically transmit the details electronically to suppliers. It requires no great feat of imagination to envisage a planned economy in which a set of standardised order control packages are generally used. These packages, instead of sending the orders directly to supplier, could route them via central servers which record the information in databases used by the planning computers. In the process, checks could be made to see if the anticipated use of critical inputs was likely to exceed availability. A generalised solution to the linear programming problem on a national scale can then be invoked to adjust up or down the intensities of outputs of different processes in order to converge on the Kantorovich ray.

7 The metric used for measuring information in this context could either be that Shannon (1948) or Li/Vitanyi (1997).

7. Conclusion

The Soviet mathematical school founded by Kantorovich and the Austrian school exemplified by von Mises took radically different positions on the feasibility of socialist economic calculation. To a large extent they ignored one another. The Austrian school largely concentrated on criticising Western trained socialist economists like Lange and the Soviet school appears to have ignored von Mises completely. Even when the key participants met, the issue was not raised. Menshikov writes:

> »It is interesting that in the account of his trip to Sweden for receiving the Nobel Prize, Kantorovich mentions an informal reception with the participation of several American economists – Nobel Prize laureates – including Hayek, Leontief, and Samuelson. But, apparently, neither at this reception, nor during other meetings, this issue was never raised. In January 1976, when I worked in USA as the Director of the United Nation Projections and Perspective Studies Branch, I was asked to present L.V. Kantorovich as a new Nobel Prize laureate at the annual meeting of the American Economic Association in Atlantic City. Of course, I put the emphasis on the economic discovery of the laureate. In the discussion, none of the audience, which included T. Koopmans and L. Klein, a future Nobel Prize laureate, mentioned the question of actual Kantorovich's answer to a part of Hayek's argumentation.« (Menshikov 2006: 1396)

With the political demise of the USSR, the Austrian school have tended to assume that von Mises arguments have been vindicated, but theoretical economic arguments are not finally resolved by politics. No, one has to bring economic arguments head to head in their own terms. Kantorovich, an absent participant in the Western debate on socialist calculation, is still worth paying attention to.

A. Appendix: Kantorovich's Algorithm

We refer in the main text of the article to Kantorovich's method of resolving multipliers. In Kantorovich (1960) he gives what is essentially a paper and pencil algorithm for his problems. The algorithm described there requires a certain residual of human intelligence to implement. In what follows I give a representation of his algorithm in a form sufficiently unambiguous as to allow computer implementation.

What follows is a program written in Vector Pascal (Cockshott 2002) a dialect of Pascal (Jensen/Wirth 1991) extended with elements of Iversons notation (Iverson 2007). Pascal is a strongly typed language which helps guard against programming errors. Iverson developed his notation whilst he was a PhD student of Leontief and was looking for a notation suitable for unambiguous expression of algorithms, initially algorithms needed for computerised prepartion of Input/Output tables. The program has been processed by a literate programming tool similar to that described in Knuth (1984a) and typeset using TEX (Knuth

1984b). The text in roman font that follow are comments describing the algorithm. The program code is generally in san-serif font. The text in roman font that follows are comments. The program code is generally in san-serif font, and the whole, is in the literate programming output format generated by the compiler.

```
program excavate;
```

The objective of the program is to solve Kantorovich's soil excavation problem by his method of resolving multipliers. It starts out from the data provided in Table 9. In the table the *norms* for the excavator types are shown in italics. In Soviet parlance, a norm appears to have meant the expected output per hour of the A-machine applied to a particular type of work. Thus Excavator A can dig out 105 m^3/hr of soil type I, 56 m^3/hr of soil of type II etc.

The objective is given in the last column: 20,000 m^3 of each type of soil.

Table 9: Simplified version of Kantorovich (1939: Table 5)

Kinds of soil	Machinery for the work						
	Excavator A		Excavator B		Excavator C		
I	105	190	107	0	64	0	20,000
II	56	92	66	222	38	0	20,000
III	56	0	83	60	53	282	20,000
Total Hours		282		282		282	

Following the columns of the norms Kantorovich gives the optimal allocation of machine times to activities to minimise overall time taken to do the digging. The program will reproduce this result by applying his method of resolving multipliers or objectively determined valuations. We first introduce our domain of discourse: the types of soil, the types of machine and the units of measurement we are using.

```
type
    soil = (I, II, III);
    excavator = (A, B, C);
    units =(hr, meter);
```

Now we introduce the dimensions in which volume, time and norms are specified. For instance norms are real numbers denoting cubic meter per hour. The word **pow** in what follows means raised to the power, and * is the multiplication operator.

```
type
    volume = real of meter pow 3;
    duration = real of hr;
    norm = real of meter pow 3 * hr pow –1;
const
    hour: duration = 1.0;
```

```
        cubicmeter: volume = 1.0;
        epsilon = 0.001;
```

The production norms for the machines working on each kind of soil and targets for soil to be moved are copied from Kantorovichs Table 5 and stored in an appropriate matrix called norms, and a scalar called targets. In a more general algorithm this could be a vector, but since all targets are the same I use a scalar.

```
const
        norms : array [soil, excavator] of norm =
                ((105, 107, 64),
                (56, 66,38),
                (56, 83, 53));
        identity: array [soil, soil] of real =
                ((1, 0, 0),
                (0, 1, 0),
                (0, 0, 1));
        target: volume = 20000 ;
```

We now introduce the variables of the problem: a matrix x which will encode the time each machine spends on each type of soil; L, a vector of objectively determined valuations of different soils. The standardised output of each machine for each soil type is obtained by applying resolving multipliers to the soil types.

```
var
        x: array [soil ,excavator ] of duration;
        Let dx ∈ duration;
        L: array [soil] of real;
        standardisedoutput: array [excavator, soil] of norm;
        outputs: array [soil] of norm;
        totals: array [soil] of volume;
        Let ok ∈ boolean;
        Let greatestsoil, leastsoil, deltam ∈ volume;
        Let best ∈ norm;
        Let e, Scoop ∈ excavator;
        Let s, least, m, j ∈ soil;
        Let λ, f ∈ real;
        Let count ∈ integer;
        equated: array [soil] of real;
procedure ComputeTotalsEtc; (see Section A1)
function marginalgain (s: soil; d: excavator): volume; (see Section A2)
function mainSoilProducedBy (e: excavator): soil; (see Section A3)
function findScoop (var s: soil): excavator; (see Section A4)
begin
```

```
L ← 1;
ok ← false;
count ← 0;
f ← 0.3;
```

Iterate the following steps until we have a satisfactory answer.

```
while not ok do
begin
    x ← 0 × hour;
```

Use the *L* to get a standardised performance for each machine.

```
    standardisedoutput ← (norms × L^T )^T ;
```

For each machine find the soil for which it has the best performance

```
    for e ← A to C do
    begin
```

find the best performance of the machine on any soil

```
        outputs ← standardisedoutput_e;
        best ← \max outputs;
```

set each machine to work on the soil it is best at

```
        for s ← I to III do
            if standardisedoutput_e,s = best then x_s,e ← hour;
    end;
    totals ← ∑ (norms × x);
    greatestsoil ← \max totals;
    leastsoil ← \min totals;
    if leastsoil ≤ 0.0 × cubicmeter then
    begin
```

check if any soil has a zero output and raise its value if it has

```
        for s ← I to III do
        if totals_s < greatestsoil then L_s ← L_s(1.02 + (ord(s)/10))
    end
    else ok ← true;
end;
count ← 0;
```

At this point our estimate of the resolving multipliers is accurate enough to ensure that some of each soil is now being moved, but we have not yet met the requirement that the same amount of each soil must be moved. We now try to get a more precise estimate of the resolving multipliers and in the process we adjust the amounts of each soil being moved. It is important to note at this point that any further adjustments must come by de-specialising some of the excavators so that they move more than one soil type. The resolving multipliers have until now been used to weight the outputs of different soil types in order to assign each digger to the soil it is best suited to. If a machine is no longer specialised, that is if it moves more than one soil, then the weights must be such that it is no longer *best* at one particular soil type. The multipliers must be set so that the marginal weighted output of the excavator on any of the soils on which it is employed are the same. Thus if a machine *k* is employed on two soils *i*, *j* then *standardisedoutput[k, j] = standardisedoutput[k, i]*.

In turn this implies that for any machine that is employed to move two soils the ratio of the resolving multipliers must be the inverse of the ratio of the norms.

The algorithm will work soil a time bringing ever more soil outputs into equality. We define the set of soils whose outputs has been brought into equality as the equated set.

For those soils in the equated set, the resolving multipliers of the soils will have been corrected so that for any machine moving more than one soil they stand in inverse ratio to that digger's norms.

computeTotalsEtc;
repeat

Find which machine not in the equated set is 2nd best at producing this soil under current resolving multipliers. Call this machine Scoop.

Scoop ← *findScoop* (*least*);
m ← *mainSoilProducedBy* (*Scoop*);

Adjust the resolving multiplier ratio between Scoops soil and the least produced soil to ratio of Scoops norms.

$$L_m \leftarrow \frac{L_{least} \times norms_{least,scoop}}{norms_{m,scoop}};$$

It is now necessary to reduce the output of scoop on scoops soil and increase it on the least produced soil. It is necessary to compute how much to reduce scoops soil by. The resolving multipliers give us substitution ratios between different soil outputs. Suppose that we want to reduce output of soil *m* by one unit and increase the output of soil *j*, the increase in *j* we get is

$$\Delta_j = \frac{L_m}{L_j} \quad .$$

If we want to reduce the output of *i* and we have two other soils *j*, *k* which we want to increase equally then we have

$$\Delta_k = \Delta_j$$

where Δ_k means change in *x* and

$$-\Delta_m L_m = \Delta_k L_k + \Delta_j L_j = \Delta_k (L_j + L_k) \ .$$

Thus

$$\Delta_m = -\Delta_k \frac{L_j + L_k}{L_m} \ .$$

Let C_m, C_j, C_k be the current outputs of each soil; given that $C_j = C_k$ we have to chose the Δs so that

$$C_m + \Delta_m = C_j + \Delta_j = C_k + \Delta_k \, .$$

It follows that

$$C_m - \Delta_k \frac{L_j + L_k}{L_m} = C_k + \Delta_k \, .$$

and

$$C_m - C_k = \Delta_k \frac{L_j + L_k}{L_m} + \Delta_k = \Delta_k (1 + \frac{L_j + L_k}{L_m})$$

so

$$\Delta_j = \frac{C_m - C_j}{1 + \frac{L_j + L_k}{L_m}} \ .$$

We next compute the reduction to be made in soil *m* from the formula

$$\Delta_m = -\Delta_k \frac{L_j + L_k}{L_m}$$

substituting we get

$$\Delta_m = -\frac{C_m - C_j}{1 + \frac{L_j + L_k}{L_m}} (\frac{L_j + L_k}{L_m}) \ .$$

Translating this to the variables used in the program we have:

$$j \leftarrow \textit{least};$$

$$\lambda \leftarrow \frac{\textit{L.equated}}{L_m};$$

$$\textit{deltam} \leftarrow \frac{(-1) \times \lambda \times (\textit{totals}_m - \textit{totals}_j)}{1 + \lambda};$$

Note that in the line above we are generalising the term $L_j + L_k$ to an arbitrary number of multipliers (1 or 2 in this program) by computing the inner product between the equated vector and the multipliers. This works because the equated vector has a 1 for all soils in the equated set. We now compute the change in duration that Scoop spends on its best soil (*dx*) by scaling *deltam* by Scoops norm for soil *m*.

$dx \leftarrow \frac{deltam}{norms_{m,Scoop}};$

$x_{m,Scoop} \leftarrow x_{m,Scoop} + dx;$

reallocate this time to Scoops best soil in the equated set which we will now call *j*

```
best ← norms_{I,Scoop} × 0;
j ← I;
for s ← I to III do
    if norms_{s,Scoop} × equated_s > best then
    begin
        best ← norms_{s,Scoop};
        j ← s;
    end ;
    x_{j,Scoop} ← x_{j,Scoop} – dx;
    computeTotalsEtc;
    count ← count + 1;
    until ((∑ equated ) = 3) ∨ (count > 10);
    writeln('answer arrived at after', count, 'trys');
    writeln('allocation', (x × target / totalsI) / hour );
end;
```

A.1 ComputeTotalsEtc

```
procedure ComputeTotalsEtc;
```

Work out how much is being produced, which soil is being produced least and which soils outputs are equals to this,

```
var
    d: array [soil] of real;
begin
    totals ←∑ (norms × x);
    leastsoil ← \min totals;
```

Find the soil that is least produced.

```
    for s ← I to III do
        if totals_s = leastsoil then least ← s;
```

Find the ones on the plan ray

$$d \leftarrow \frac{totals\text{-}leastsoil}{cubicmeter};$$

$$equated \leftarrow \begin{cases} 1.0 & \text{if } (d<\varepsilon) \wedge (d>-\varepsilon) \\ 0.0 & \text{otherwise} \end{cases};$$

end;

A.2 marginalgain

function *marginalgain* (*s*: *soil*; *d*: excavator): *volume*;

This computes the marginal gain, under the weighting imposed by the current resolving multipliers, of a small shift of the digger *d*'s time to the specified soil type *s*. We compute the effect of multiplying all current time allocations to $1-\varepsilon$ whilst increasing the allocation of time to soil *s* by ε. The assumption here is that for now each machine has only one hour to allocate.

const
  epsilon = 0.001;
var
  Let *currentoutput* ∈ volume;
begin
  $currentoutput \leftarrow \Sigma\, x_d \times norms_{I0,d}$;
  $marginalgain \leftarrow ((\varepsilon \times hour) \times norms_{s,d}) - \varepsilon \times currentoutput$;
end;

A.3 mainSoilProducedBy

function *mainSoilProducedBy* (*e*: *excavator*): *soil*;

This determines which soil excavator *e* produces the most of.

var
  Let *v* ∈ volume;
  Let *j*, *s* ∈ soil;
begin
  $v \leftarrow 0 \times cubicmeter$;
  $s \leftarrow i$;
  for $j \leftarrow I$ **to** *III* **do**
  begin
    if $v < norms_{j,e} \times x_{j,e}$ **then**
    begin
      $v \leftarrow norms_{j,e} \times x_{j,e}$;
      $s \leftarrow j$;

```
            end;
        end;
        mainSoilProducedBy ← s;
    end;
```

A.4 findScoop

function *findScoop* (**var** *s*: *soil*): *excavator*;

Find which machine not currently fully committed is best at producing the soil *s*. The parameter s is updated by the call. The soil s must be drawn from one of those in the equated set. We call this machine Scoop. The algorithm searches to find which machine will have the greatest marginal output of the soil in the equated set per unit of other soil it gives up by switching to produce *s*.

```
var
    Let gain ∈ volume;
    Let j, m ∈ soil;
    Let d, Scoop ∈ excavator;
begin
    Scoop ← A;
    gain ← (–maxint) × cubicmeter;
    for d ← A to C do
        for j ← I to III do
            if equated_j > 0 then
            begin
                m ← mainsoilproducedby (d);
                if marginalgain (j, d) > gain then
                begin
                    if equated_m < 1 then
                    begin
                        gain ← marginalgain (j, d);
                        Scoop ← d;
                        s ← j;
                    end;
                end;
            end;
            findScoop ← Scoop;
end;
```

References

Appel, J., Baker, M., Deutschlands, A.A., Kommunisten, G.I., (1990): Fundamental principles of communist production and distribution, Movement for Workers' Councils, Kollektivarbeit der Gruppe Internationaler Kommunisten – GIK [Allgemeine Arbeiter Union Deutschlands – AAUD].

Bienstock, D. (2002): *Potential Function Methods for approximately solving Linear Programming Problems: Theory and Practice*, Norwell, Massachusetts: Kluwer Academic Publishers.

Cottrell, A., Cockshott, P. (1992): *Towards a New Socialism*, Nottingham: Bertrand Russell Press.

Cockshott, P. (2002): Vector pascal reference manual, in: ACM SIGPLAN Notices, 37(6), New York: ACM Press, 59 – 81, URL: http://doi.acm.org/10.1145/571727.571737.

Dantzig, G.B., Wolfe, P. (1961): The decomposition algorithm for linear programming, in: *Econometrica*, 29(4), 767 – 778.

Dickinson, H.D. (1933): Price formation in a socialist community, in: *The Economic Journal*, 43(170), 237 – 250.

Dieterich, H. (2002): *La Democracia Participativa: El Socialismo del Siglo XXI*, Baigorri: Pais Vasco.

Farjoun E., Machover M. (1983): *Laws of Chaos, A Probabilistic Approach to Political Economy*, London: Verso.

Gondzio, J., Grothey, A. (2006a): Solving nonlinear financial planning problems with 10^9 decision variables on massively parallel architectures, in: Constantino, M., Brebbia, C.A. (eds.), *Computational Finance and its Applications II*, Southampton/Boston: WIT Press, 95 – 108.

Gondzio, J., Grothey, A. (2006b): Solving distribution planning problems with the interior point method, Technical Report MS-06-001, School of Mathematics, University of Edinburgh, Edinburgh EH9 3JZ, Scotland, UK.

Grossman, G. (1963): Review: Against bourgeois economic pseudo-theories of socialism, in: *The American Economic Review*, 53(1), 211 – 213.

Iverson, K. (2007): Notation as a tool of thought, in: ACM Turing Award Lectures, New York: ACM, 1979, URL: http://doi.acm.org/10.1145/1283920.1283935.

Jensen, K. Wirth, N. (1991): *PASCAL User Manual and Report: ISO PASCAL standard*, Berlin: Springer.

Kantorovich, L.V. (1960): Mathematical methods of organizing and planning production, in: *Management Science*, 6(4), 366 – 422.

Kantorovich, L.V. (1965): *The Best Use of Economic Resources*, Oxford: Harvard University Press.

Karmarkar, N. (1984): A new polynomial-time algorithm for linear programming, in: *Combinatorica*, 4(4), 373 – 395.

Khachian, L. (1979): A polynomial algorithm in linear programming, in: *Soviet Mathematics Doklady*, 20, 191 – 194.

Klee V., Minty G. (1972): How good is the simplex algorithm, in: Shisha, O. (ed.), *Inequalities-III*, New York: Academic Press, 159 – 175.

Knuth, D.E. (1984a): Literate programming, in: *The Computer Journal*, 27(2), 97 – 111.

Knuth, D.E. (1984b): *The TeXbook*, Addison-Wesley.

Lange, O. (1938): *On the Economic Theory of Socialism*, Minnesota: University of Minnesota Press.

Li, M., Vitanyi, P.M.B. (1997): *An Introduction to Kolmogorov Complexity and Its Applications*, New York: Springer.

Menshikov, S.M. (2006): Topicality of Kantorovich's economic model, in: *Journal of Mathematical Sciences*, 133(4), 1391 – 1397.

Nemchinov, V.S. (1964): *The use of Mathematics in Economics*, Cambridge: MIT Press.

Neurath, O. (2004): Economic plan and calculation in kind, in: Uebel, T.E., Cohen, R.S. (eds.) *The Use of mathematics in economics,* edited by A. Nove, publisher Oliver and Boyd, Edinburgh, 1964, London: Kluwer.

Nove, A. (1983): *The Economics of Feasible Socialism*, London: George Allen and Unwin.

Nove, A. (1964): Introduction, in: Use of mathematics in economics, Oliver and Boyd, wpc, 2008.11.04.

Novozilov V.V. (1964): Cost benefit comparisons in a socialist economy, in: Use of mathematics in economics, Oliver and Boyd, wpc, 2008.11.04.

Peters, A. (1996): *Das Äquivalenzprinzip als Grundlage der Global Ökonomie*, Akademische Verlagsanstalt.

Peters, A. (2000): *Was ist und wie verwirklicht sich Computer-Sozialismus: Gespräche mit Konrad Zuse*, Berlin: Verlag Neues Leben.

Ricardo, D. (1951): Principles of political economy and taxation, in: Sraffa, P. (ed.), *The Works and Correspondence of David Ricardo*, Volume 1, Cambridge: Cambridge University Press, 249 – 298.

Shannon, C. (1948): A mathematical theory of communication, in: *The Bell System Technical Journal*, 43, 379 – 423, 623 – 56.

Sipser, M. (2006): *Introduction to the Theory of Computation*, Oxford: Thomson.

Swann, M.J. (1975): On the theory of optimal planning in the Soviet Union, in: *Australian Economic Papers*, 14(24), 41 – 56.

Turing, A. (1937): On computable numbers. With an application to the Entscheidungsproblem, in: *Proceedings of the London Mathematical Society*, 1(2), 126 – 151.

Turing, A. (1950): Computing machinery and intelligence, in: *Mind*, 49, 433 – 460.

Turing, A. (2004): Lecture on the automatic computing engine, in: Copeland, B.J. (ed.), *The Essential Turing*, New York: Oxford University Press.

von Mises, L. (1935): Economic calculation in the socialist commonwealth, in: Hayek, F.A. (ed.), *Collectivist Economic Planning*, London: Routledge and Kegan Paul, 87 – 130.

von Mises, L. (1949): *Human Action*, London: Hodge and Company.