Title	FLEXIBLE ADMINISTRATION WITH A WELL-DESIGNED INFORMATION STRUCTURE
Author(s)	NAGAO, TERUYA
Citation	HOKUDAI ECONOMIC PAPERS, 3, 38-45
Issue Date	1972
Doc URL	http://hdl.handle.net/2115/30644
Type	bulletin (article)
File Information	3_P38-45.pdf

FLEXIBLE ADMINISTRATION WITH A WELL-DESIGNED INFORMATION STRUCTURE

TERUYA NAGAO

> In the faces of an uncertain future, speech formulates an anticipation. It traces within the chaos of circumstances the first hints of the future. ....
>
> Speaking alters the form of the situation. It is a pledge and commitment, the signature of a contract which may appear to be a loss of freedom, but which in fact guarantees man's attainment of a new freedom through the power of obedience.
>
> Georges Gusdorf

I. THE STORY OF LONDON'S THIRD AIRPORT

Recently we saw a provocative NHK–TV program, entitled "The Residents' Selfishness and Flexible Administration". We hear that all over Japan objections are being raised against authorities' construction activities of public service facilities, such as highways, garbage dumps, airports and so on. On this program several cases in Japan were introduced and afterwards a case in England was discussed.

It was the story of the construction of London's third airport. At first, the village of Stanstead near London was proposed as the best site for it. The authorities met with the villagers' stout resistance. This plan seems to have been decided without sufficient investigation. The residents asked the authorities why their village had been selected. When the authorities met with the villagers' resistance, they abandoned their plan and established a new experts' investigation committee to search for new alternatives. The committee investigated no less than eighty proposed sites, and made the results public in voluminous reports. Finally the majority of the committee decided upon a city, but a countermovement broke out there. After further consideration, government authorities chose a village which only a minority of the committee had favored. As a result, London's third airport was situated in Foulness at the mouth of the river Thames.

After telling this story, an economist, Mitsuharu Ito concluded by saying that we should learn a lesson from such a flexible administration. When we turn our thoughts to the problem of refuse disposal which is now the most important problem in our country, the situation is the same. Everybody knows garbage must be dumped somewhere as it is nothing but a by-product of the city system. Therefore establishing facilities to burn refuse is the need of the hour. City waste may be dumped into sea, but the problem of water polution will remain.

But facilities to burn garbage are never constructed without objections being raised by those who reside in the surrounding area. We, of course, do not say a countermovement itself is wrong. On the contrary, for the sound development of democracy the residents should have the power to appeal to the government.

But herein lies another problem. What happens if the location of facilites to burn garbage is determined by the strength of the political power of the residents? Then, only those who devote their energy to the resistance movement can protect their own interests, while others are forced to accept the location of facilities to burn refuse in their district simply because they do not resist. If whether the residents resist or not greatly influences the authorities' decision, everyone will waste his time and energy on politics.

They say that the villagers of Stanstead asked the authorities to clarify why it was decided to locate the airport in Stanstead. This is the point. The best way to solve this problem is to collect all possible information, listening to the opinions of all those involved and to make a selection keeping in mind the costs and benefits of each alternative. This is indeed right, but it leads us then to this question: what is the most effective way to explain the costs and benefits of each plan to the satisfaction of the residents involved?

What Mitsuharu Ito calls a gentle administration should not be so mild that nothing is carried out for fear of repercussions. It would require that we should rather investigate as many alternatives as possible within a given limited time interval and then act accordingly.

The method, which is called cost and benefit analysis, may be useful in action-selecting and problem-solving. Making good use of computers, we can compile and investigate hundreds of alternative plans. But a computer is neither the Almighty nor a pewter who receives oracles from the gods above. We can never rely on computers until we understand reasoning perfectly. In a sense, a computer is no more than a supersonic moron. That is, it can do nothing except give a perfectly detailed prescription.

In this story, fortunately the committee could find a suitable place at the mouth of the river Thames. And even better, the resistance of the residents was less than that of Stanstead. There is no problem in such a case of London's third airport when there is enough negotiation with the help of experts. But such happy solutions are not always to be found. But when we have limited time to discuss all the alternatives, it is a critical element whether we can draw logical conclusions easily or not.

II. WHY DO WE TAKE A "QUANTITATIVE ANALYSIS"?

There is a popular idea that the most effective method to make people understand is to offer a "quantitative analysis". In fact, it is very easy to solve problems with sufficient quantified information. When it can be said that one project offers more benefits than any other — the benefits are measured in some quantitative manner — this project will be promptly accepted. But is there so many projects of which administrative authorities can show their costs and benefits in a simple quantitative manner?

Now let us take a look at the essence of the so-called quantitative analysis in a rather broad perspective. Here are three plans, each of which we label X, Y, and Z:

Plan X gets 70 marks.
Plan Y gets 90 marks.
Plan Z gets 50 marks.

We can see at once from this list of scores the interrelationship of the outcomes of the three plants. That is to say, 70 is better than 50, and similarly 90 is better than 70. Thus numerical expressions show an order, that is, a sort of mutual relationship of the three evaluations. And we propose to use the word "structure", instead of "interrelationship". Here are other expressions:

Plan X is splendid.
Plan Y is excellent.
Plan Z is good.

These terms, "excellent", "good" and so on, are difficult to use in effective comparisons, because they don't have definite significations. It is impossible to discern a clear difference among them because they are not clearly related to one another. We say that such a system of symbols does not have a firm structure, so they are not suitable for evaluative uses. Now we adopt still other expressions:

Plan X is marked B.
Plan Y is marked A.
Plan Z is marked C.

If it is already known that A precedes B, and that B precedes C, these symbols can show an order. We say that the group of three alternatives, plan X, Y, Z are expressed by three symbols with a distinct structure. That is, they are clearly related to one another. The essence of what we emphasize with these simple examples is as follows: such and such a kind of intellectual operation requires the formal system of symbols which has such and such a kind of structure. In the above case, our task is a very simple sort of intellectual operation, that is, placing things in relative precedence.

So we need not confine ourselves to the so-called quantitative analysis. It is a particular type of symbolic devices which represent matters in structured systems of symbols. We have today many types of such systems of symbols, which we call "languages", although they are not "ordinary" or "daily" languages. With the aid of these "artificial" languages we can carry out many types of logical operations other than simple comparisons.

III. WE NEED A "LOGICAL OPERATION" FOR ADMINISTRATIVE ACTION.

In order to effectuate logical operations, we must devise a language which has a system of carefully designed syntactical rules, that is, we create and use a system of symbols which is so constructed that the logical deductions may be easily developed on it.

Our next illustration shows the "network" type of artificial language which helps us in the logical operation of finding the proper work sequence. Suppose we are going to embark on a project, at first we must make a plan, that is, we must decide upon the requirement of resources and various but interrelated activities or operations to be performed. Then we must establish their necessary sequence or order of performance, so that we get, at last, a "schedule" on which the project will be carried out. And if the schedule is a good one, it will serve as a guide, if not, many problems must be solved as they occur.

In order to find the necessary sequence from the long list of activities, we must devote ourselves to the tedious work of arranging and rearranging. This task is quite cumbersome and prone to error. Here comes "network diagramming", and it yields to us many advantages. And the larger the network, that is, the more the activities included in the project, the larger the advantage of network-modeling to overcome tedious and cumbersome work to get a proper sequence.

It is to be noted that network-representation is not a "scale" model like a replica of a building or a ship, but a "logic" model. Let us explain how to draft a network with an extremely simplified model of an airport construction project. Our project consists of five activities. They are: procurement of the materials; readjustment of the land; construction of the foundation, of the buildings, are of the landingstrip. For each of these activities, we know about what sort of activities should have been finished before we set about. We are provided with information which is a collection of "Before (Building) = {Procurement, Foundation}" which means that, before we begin construction of a building, we must procure the necessary materials and accomplish the foundation work.

The complete list of information for this case is as follows.

Before (Procurement) = { }
Before (Readjustment) = { }
Before (Foundation) = {Readjustment}
Before (Building) = {Procurement, Foundation}
Before (Landing Strip) = {Readjustment}

It is very easy for us to get from this list a "network" diagram, which is shown in Exhibit 1.

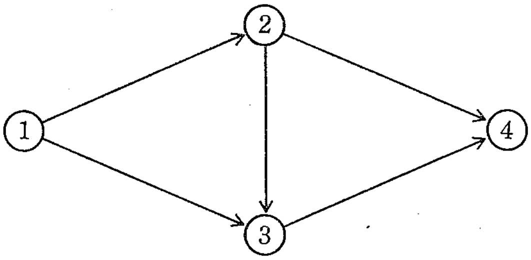


Exhibit 1.
1–2 readjustment of the land
1–3 procurement of the materials
2–3 construction of the foundation
2–4 construction of the landing strip
3–4 construction of the building

Of course in a real situation, a project such as airport construction can be represented by nothing but an extremely large network which is to be drafted on a huge sheet of paper of, perhaps, several meters in length. And that, to represent a project by means of a network diagram is, in other words, to talk about it in "network language". We must emphasize that the computer can understand such language and will come to discuss it with us in making a good schedule, and will store in its memory every detail thereof in order readily to make use of it whenever necessary.

Now we will explain how flexible a control system can be, when supported by information storage written in "network" language. In the process of accomplishing a project, nobody can predict perfectly what changes in the environment may occur. A project might encounter an accident, but it must be completed within a certain period. In a traditional, rigid administration, what would happen if, for example, a machine is out of order and an activity cannot be started as scheduled? It could be a disaster in a large scale project which has a great many interrelated activities. For in this traditional system, once all the activities are scheduled, none can be easily changed until the completion of the whole project. Even a small change might badly interfere with the schedules of the remaining activities. Sadly, traditional administrative systems are locked in to a narrow pattern of alternatives, so it can only push along on its original plan at whatever the cost. On the other hand, suppose we could change the date of an activity, and make relevant changes of the dates of hundreds of activities and yet accomplish the project within a prescribed period!

It is not too much to say that it is a flexible administration which can not only adapt to any changes in situations, but which can also attain its objectives. When a network language is put to use and the list of all activities is written down according to the grammatical rules thereof, the computer can read and store every detail of the project. The whole scope of information is quickly processed and a proper schedule of the project is automatically made. It is also shown which jobs are critical, that is, which ones must be completed at a predetermined time, and which jobs can be adjusted easily. If an unexpected change occurs in the environment, all the information is quickly re-processed and a new schedule is automatically made which can meet new constraints and also meet the deadline of the project. Having well-designed syntactical rules, the network language is a good example of the device of "structurization" of information. Now let us turn our discussion to the traditional "resource allocation" setting and again to take the simplest example.

We are given a set of jobs to be performed by a team of workers in the least possible time. There are four persons, whom we can call A, B, C, and D respectively and four jobs, which we can call a, b, c, and d respectively. We are to choose one alternative action, that is, to select one from twenty-four assignments, which are shown in Exhibit 2.

A→a	A→a	A→a		A→d
B→b	B→b	B→c	…………	B→c
C→c	C→d	C→b		C→b
D→d	D→c	D→d		D→a

Exhibit 2. alternative actions (24 assignments)

The amount of time required by each worker to do each job is known. So it is, of course, easy to select the best plan from among the twenty-four alternatives. But in the case of ten workers and ten machines, the number of possible alternatives will be 3,628,800. If twenty men are each to be given a job, the number of alternative actions amounts to the frightful figure of 2,432,902,008,176,640,000. If we want to know the merits of twenty persons for doing twenty jobs, it suffices for us to get the 400 data of man-job pairs. But in order to select the best assignments from the above astronomical number of alternative actions, even the largest computer in the world would take many, many years.

Now the problem presented above is that which is a special simple type of linear programming model. It can be solved easily by simplex algorithm, that is, a general problem-solving procedure applicable to linear programming problems. At the same time, this type of problem falls into the family of "minimal cost flow problems" on networks. It also belongs to the set of problems which can be treated by the powerful tool called branch and bound algorithm. So, we can say, it can be treated by any of the "linear-programming" language "network" language, and the "branch-and-bound" language. We also use another denotation for each of these, "Dantzig" language, "Ford-Fulkerson" language, "Golomb-Baumert" language, respectively.

They are formal systems, that is they are special types of artificial languages having limited usage. When we learn the "grammar" of either of these systems, we can then leave the actual problem-solving process to computers.

Our point is that these languages are to be properly accepted as media for dialogues between men and computers. They, the mathematical programming models, or the network expressions are, never to be taken literally as a picture of the real world. They are at best abstract constructions or symbolizations for use in communication, and should not be thought of as descriptions of any real situations.

— To be continued in the next issue —

ACKNOWLEDGEMENT

The author wishes to express his gratitude to Mr. J. C. A. Rundall, First Secretary, Science and Technology Dept., British Embassy, for his precise information about the circumstances of London's third airport case.

(a)

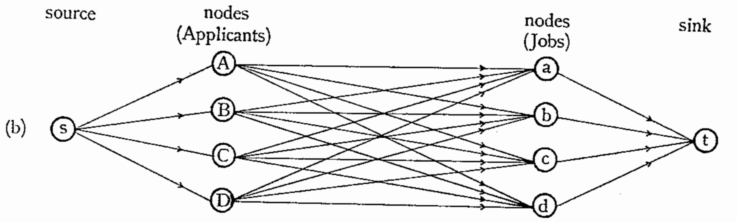
initial tableau
Destinations
(Jobs)

Origins (Applicants)	a	b	c	d
A	5	13	9	5
B	10	15	16	9
C	8	8	6	13
D	6	3	8	7

final tableau

	a	b	c	d
A	$\boxed{0}$			$\cancel{0}$
B				$\boxed{0}$
C			$\boxed{0}$	
D		$\boxed{0}$		

(b)



(c)

minimize the function

$$-5x_{11}-13x_{12}-9x_{13}-5x_{14}-10x_{21}-15x_{22}-16x_{23}-9x_{24}-8x_{31}-8x_{32}-6x_{33}-13x_{34}-6x_{41}-3x_{42}-8x_{43}-7x_{44}$$

subject to the conditions

$$\begin{aligned}
&x_{11}+x_{12}+x_{13}+x_{14} \leqq 1\\
&x_{21}+x_{22}+x_{23}+x_{24} \leqq 1\\
&x_{31}+x_{32}+x_{33}+x_{34} \leqq 1\\
&x_{41}+x_{42}+x_{43}+x_{44} \leqq 1
\end{aligned}$$

$$\begin{aligned}
&x_{11}+x_{21}+x_{31}+x_{41} \geqq 1\\
&x_{12}+x_{22}+x_{32}+x_{42} \geqq 1\\
&x_{13}+x_{23}+x_{33}+x_{43} \geqq 1\\
&x_{14}+x_{24}+x_{34}+x_{44} \geqq 1
\end{aligned}$$

$$x_{11},\ x_{12},\ x_{13},\ x_{14},\quad x_{21},\ x_{22},\ x_{23},\ x_{24},\quad x_{31},\ x_{32},\ x_{33},\ x_{34},\quad x_{41},\ x_{42},\ x_{43},\ x_{44} \geqq 0$$

(d)

initial step
Destinations
(Jobs)

Origins (Applicants)	a	b	c	d
A	⑤			⑤
B				
C			⑥	
D		③		

final tree

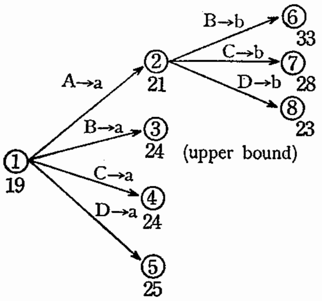


Exhibit 3.