STUDIES OF THE BEHAVIOR OF

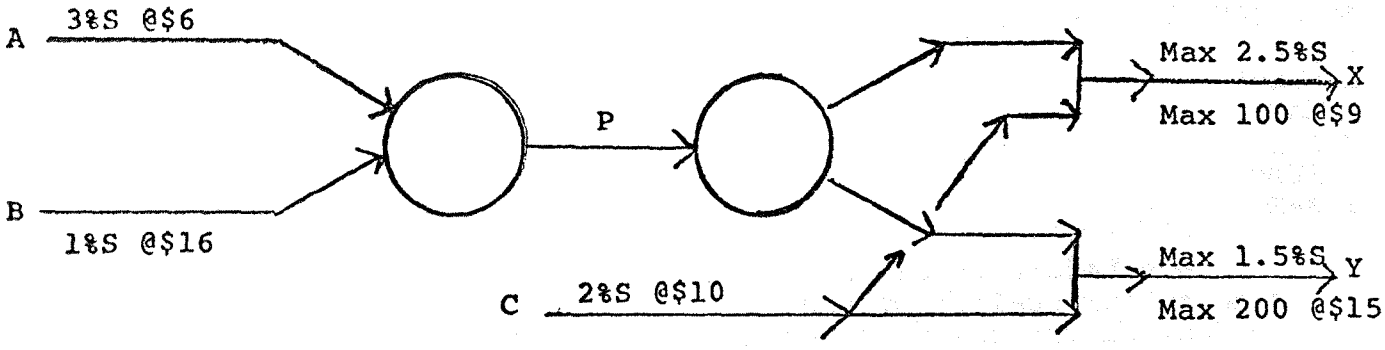
RECURSION FOR THE POOLING PROBLEM

C. A. Haverly

A study of the behavior of LP models is a badly neglected area. By model behavior I mean the way the answers of an LP model vary as the input data is varied. In some cases, we find the model behaves much like the system being modeled. But in other cases it has quite different properties. Some of the erratic behavior can be compensated for by better formulation. Some must be taken into account by the LP analyst.

Few studies of model behavior have been published. In 1977 at MUGI I gave a paper which has since been published in the SIGMAP newsletter (Issue 22, Dec. 1977). This year I'd like to report on some studies on solving the pooling model by recursion.

We define a straight forward example as follows:



If there was no pooling restriction (link P) we could formulate and solve as a linear problem.

	AX	BX	CX	X	AY	BY	CY	Y	
PROFIT	-6	-16	-10	9	-6	-16	-10	15	
X BAL	-1	- 1	- 1	1					= 0
X SULF	3	1	2	-2.5					≤ 0
X MAX				1					≤ 100
Y BAL					-1	- 1	- 1	1	= 0
Y SULF					3	1	2	-1.5	≤ 0
Y MAX								1	≤ 200

The optimal answer is: AX=CX=50,X=100, BY=CY=100,Y=200, profit=$500.

But link P (pooling) is specified. It may exist physically because there is only one tank to store A and B in. Or it may exist because A and B must be mixed and transported as a mixture (via pipeline, tanker, tank car or whatever).

Let us first consider a simple recursion formulation such as:

	A	B	PX	CX	X	PY	CY	Y	
PROFIT	-6	-16		-10	9		-10	15	
POOL BAL	-1	- 1	1			1			= 0
X BAL			-1	- 1	1				= 0
X SULF			p	2	-2.5				≤ 0
X MAX					1				≤ 100
Y BAL						-1	- 1	1	= 0
Y SULF						p	2	-1.5	≤ 0
Y MAX								1	≤ 200

But before we can solve we must estimate the coefficient p (the sulfur quality of the pool of mixed A and B). Then having solved, we can check if our assumed value of p was correct based on the amount of A and B chosen. If not we revise p, re-solve and check. This recursive process continues until our estimate of p put in the matrix is close to that calculated from the mixture of A and B chosen.

Let us assume a 50%/50% mix of A and B. The assumed p will be ½ x 3%S + ½ x 1%S = 2%S. So we set the coefficient p=2 and solve, obtaining:

Y=PY=CY=0, X=PX=A=100, profit=300.

We calculate p=(100x3%S+0)/100=3%. Since we had used 2% for p in the matrix, we are off. Therefore we revise the p coefficient to 3 and re-solve. The new optimal is:

Y=PY=CY=0, X=100,PX=CX=50,A=100, profit=100.

We calculate p=(50x3%S+0)/50=3%. This is exactly what we had used as a coefficient. Therefore we have converged. The answer appears to be a proper optimal and we will call it our FINAL optimal. We might think it to be the global optimal, i.e. the best possible optimal. When we deal with non-linear models (such as the pooling problem) there may be a number of "local"optimals - some better than others. The best is called the global optimal. However, as we will see later, this particular optimal is not a global optimal. The true global optimal is: X=0,Y=200,PY=CY=B=100, profit=400.

But what would have happened had we started with a different assumed p? We try again, starting with p=3. Immediately we get the same FINAL optimal as above.

We try again starting with p=1. Now the LP gives us a first optimal of:

Y=200,PY=CY=100,X=100,PX=X=100,A=200,profit=1700.

But calculated p=3 vs. 1 assumed. When we change the coefficient p to 3 and resolve we get the same FINAL optimal as above.

Since p cannot be less than 1 or greater than 3, we can try a variety of initial starting assumptions for p in that range. But it can be shown that every value will behave as one of the examples above: like the first for p greater than 1.5 and less than 3, and like the last for p equal to 1.5 or less.

Since no value of p will cause the model to converge to the global optimal, we have to reluctantly conclude that this simple recursion formulation is not reliable. It has a fundamental flaw involving a failure to link the economics associated with the sulfur constraints back into the feed selection process. A and B are selected solely on the basis of price (since they contribute volume to the pool equally). Therefore the cheaper (in this case A) will always be chosen regardless of its impact on the sulfur in the pool.

A BETTER RECURSION FORMULATION

Therefore we go to a formulation where a balance row has been added for the pool sulfur and where two "correction" vectors are added representing a pool quality over or under that assumed for the p coefficient. This formulation better reflects sulfur economics back to component selection.

	A	B	OVER	UNDER	PX	CX	X	PY	CY	Y	
PROFIT	-6	-16				-10	9		-10	15	
POOL BAL	-1	- 1			1			1			= 0
POOL SULF	3	1	-1	1	-p			-p			= 0
X BAL					-1	- 1	1				= 0
X SULF			f	-f	p	2	-2.5				$\leq$ 0
X MAX							1				$\leq$ 100
Y BAL								-1	- 1	1	= 0
Y SULF			1-f	f-1				p	2	-1.5	$\leq$ 0
Y MAX										1	$\leq$ 200

We still have our recursion coefficient p, but in addition, we now have another recursion coefficient f representing the estimated fraction of the pool quality which will go to product X. Therefore 1-f is the amount going to product Y.

As far as I know this formulation is the best known for pooling model. However, it too has problems as we will see later.

FIRST SERIES OF STUDIES

In this formulation we have two recursion variables p and f. The variable p can range between 1.0 and 3.0 and the variable f between 0 and 1. For p, values of 1.0, 1.2, 1.5, 2.0, 2.5, 2.9 and 3.0 were chosen. For f, values of 0, .3, .5, .7 and 1 were chosen. Computer runs for the various combinations of these starting values were made.

The most indicative results are obtained on the FIRST recursion optimal and that is what is shown in the attached table. The results of the first optimal almost always indicates whether the model will converge to the global or to a non-global optimal.

Besides the details of the FIRST optimal, we also show the number of recursion iterations required to reach final optimal and whether it is the global optimum or not.

TABLE 1 - STANDARD RUN

First Recursion Optimals for Pooling Model Obtained

from Various Starting Values of p and f

Basis of Model: as shown earlier.

Global optimal: f=0, p=1.0

	Input		Optimal Answers												
Case	f	p	A	B	OVER	UNDER	PX	CX	X	PY	CY	Y	Profit	Global Opt.	Rec. Iter.
1	0	1.0	-	100	-	-	-	-	-	100	100	200	400	*	1
2		1.2	-	100	-	20	-	-	-	100	100	200	400	*	2
3		1.5	-	100	-	50	-	-	-	100	100	200	400	*	2
4		2.0	-	100	-	100	-	-	-	100	100	200	400	*	2
5		2.5	-	100	-	150	-	-	-	100	100	200	400	*	2
6		2.9	-	100	-	190	-	-	-	100	100	200	400	*	2
7		3.0	-	100	-	200	-	-	-	100	100	200	400	*	2
8	.3	1.0	71	129	143	-	-	86	86	200	-	200	429	*	3
9		1.2	63	137	86	-	-	51	51	200	-	200	377	*	3
10		1.5	50	150	-	-	-	-	-	200	-	200	300	s	3
11		2.0	-	143	-	143	-	-	-	143	57	200	143	*	2
12		2.5	4	96	-	143	100	-	100	-	200	200	336	*	3
13		2.9	24	76	-	143	100	-	100	-	200	200	536	*	4
14		3.0	21	72	-	143	93	7	100	-	200	200	557	*	4
15	.5	1.0	50	100	100	-	-	100	100	150	20	200	500	*	3
16		1.2	69	119	100	-	-	100	100	188	12	200	462	*	3
17		1.5	50	150	-	-	-	-	-	200	-	200	300	s	3
18		2.0	-	-	-	-	-	-	-	-	-	-	-	N	1
19		2.5	-	-	-	-	-	-	-	-	-	-	-	N	1
20		2.9	-	-	-	-	-	-	-	-	-	-	-	N	1
21		3.0	-	-	-	-	-	-	-	-	-	-	-	N	1
22	.7	1.0	36	85	71	-	-	100	100	121	79	200	529	*	3
23		1.2	51	101	71	-	-	100	100	152	48	200	498	*	3
24		1.5	50	150	-	-	-	-	-	200	-	200	300	s	3
25		2.0	-	-	-	-	-	-	-	-	-	-	-	N	1
26		2.5	75	25	-	-	100	-	100	-	-	-	50	N	3
27		2.9	66	34	-	57	100	-	100	-	34	34	136	N	3
28		3.0	64	36	-	71	100	-	100	-	43	43	157	N	3
29	1	1.0	25	75	50	-	-	100	100	100	100	200	500	*	3
30		1.2	38	87	50	-	-	100	100	125	75	200	525	*	3
31		1.5	75	125	50	-	-	100	100	200	-	200	450	*	3
32		2.0	50	-	50	-	50	50	100	-	-	-	100	N	2
33		2.5	50	-	25	-	50	50	100	-	-	-	100	N	2
34		2.9	50	-	5	-	50	50	100	-	-	-	100	N	2
35		3.0	50	-	-	-	50	50	100	-	-	-	100	N	1

* leads to global optimal, s leads to global only if converge on f as well as p,
N leads to non-global optimal

RESULTS OF FIRST SERIES OF RUNS

The series of runs shown in Table 1 are quite interesting. In 23 of the 35 runs we reached the global optimal. In no case were more than four recursion iterations required. In all cases where we started with what proved to be either the optimal f (0) or the optimal p (1.0) we converged to the global optimum (cases 1-7, 8, 15, 22 and 29). Even when our starting assumption was only good for either the f (0 to .3) or the p(1 to 1.5) we still converged to the global optimum. When our starting f and p were both bad (.5 or greater for f and 2.0 or greater for p) we did not find the global optimum.

This model and set of data converged to one of four solutions; the global optimum as in case 1, a local optimum as in case 35, a null solution where recursion "never got going" as in case 18, or a false optimum as in case 10.

In the third of these, the reason recursion "never gets going" is that the price economic driving forces are not strong enough to overcome the fractional f. Product Y can't be made at all from 2% Sulfur pool material and 2% purchased material, as its spec is 1.5% or less. It could be made if the UNDER vector came into the LP basis. But since f is .5 only half the incentive is transmitted to the pool sulfur row, and this proves insufficient to overcome the extra cost for lowering the sulfur content of the pool.

At the same time product X is uneconomical since the pool material currently costs (6+16)/2=$11 vs. a $9 selling price. It would normally bring in the OVER vector to raise the sulfur content of the pool and lower its cost below $9. But the OVER vector can't come into the solution as long as no product Y is being made (because of the coefficient in the YSULF row of vector OVER).

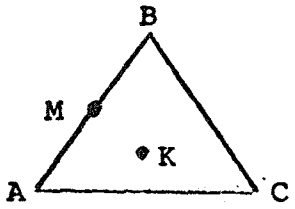
If there had been a bigger economic driving force in the form of higher prices for products X and Y the recursion would have "gotten going". This suggests that some recursion cases may not behave well when the economic driving forces are weak.

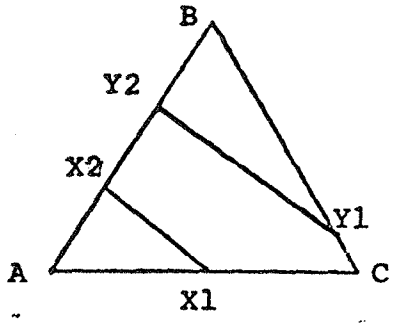
Just to demonstrate this we ran case 36 which was the same as case 18 except the selling price of Y was changed from $15 to $20 to increase the driving force. It converged in 2 recursion iterations to the global optimum as expected.

Cases 10, 17 and 24 provided the biggest surprise because at first we stopped recursion at a false optimum. We had thought it necessary to converge only to f or to p. However, in these cases p had converged perfectly and both correction vectors (OVER and UNDER) were exactly 0. However, we did not have a proper optimum. Once we required both p and f to converge, this problem disappeared and we went on to the global optimum. All recursion users should take warning.

APPENDIX A
A Graphical View of the Pooling Model

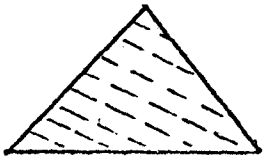
Since we have three feeds (A, B and C) we can plot on a tertiary diagram. This shows all possible percentage mixtures of our three components. Each vertex represents 100% of one component (for example A represents only A feed with no B or C). Each edge represents mixtures of only two components going from vertex A to point M to vertex.B represents going from 100% A, 0%B at point A to 50% A, 50% B at point M and 0% A, 100%B at point B. Points within the boundaries are for three components mixtures. For example K would be 20%B, 40%A, 40%C.



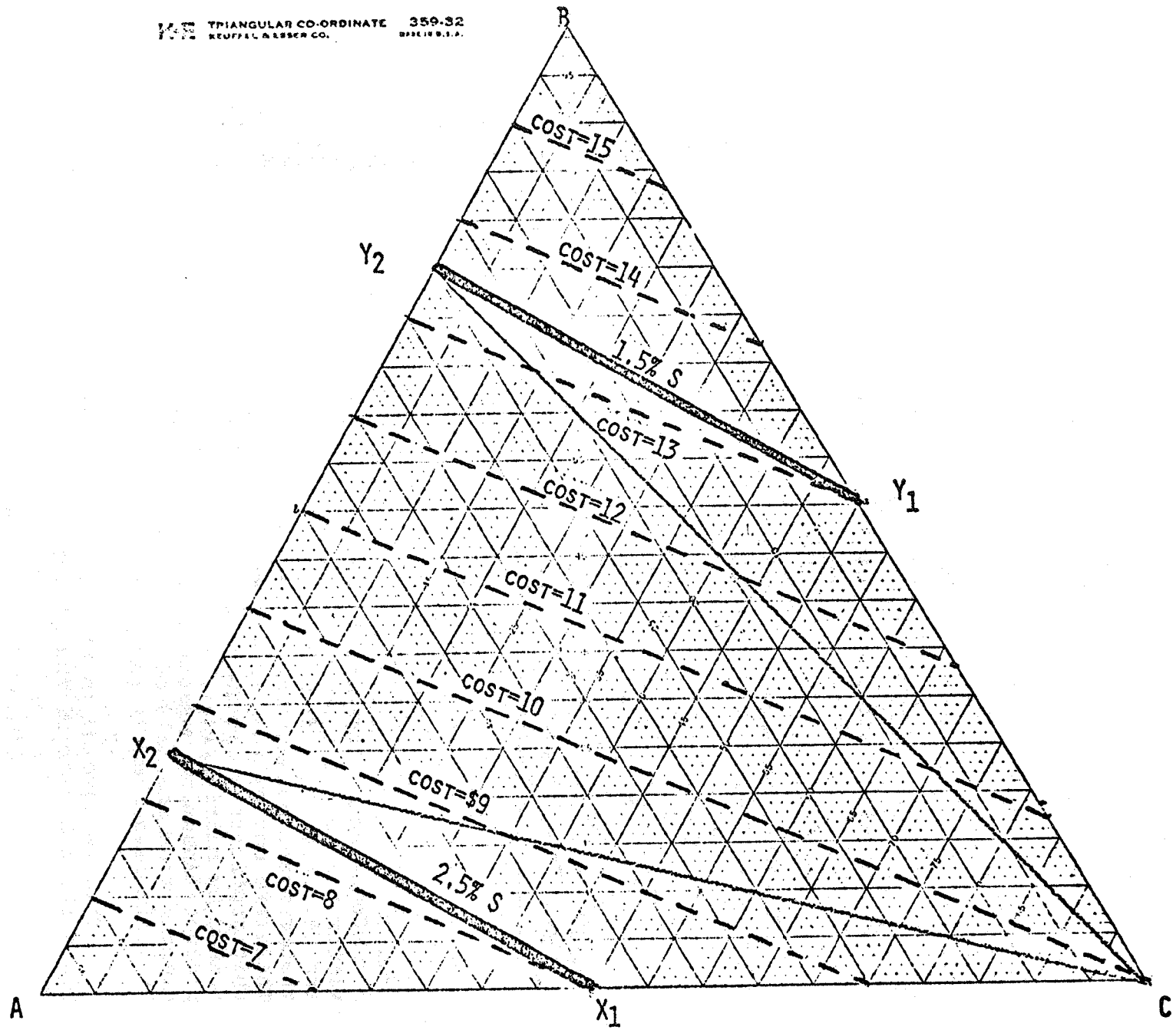


Next we plot the feasible region for each of our two products X and Y. They must meet sulfur specs of 2.5% and 1.5% respectively. The plot of the mixtures of A, B and C (3%, 1%, 2% Sulfur respectively) which exactly meet these product specs are shown as X1-X2 and Y1-Y2. All mixtures above these lines are feasible with respect to the specs of the respective products.

Next we examine the economics. A, B and C are priced at $6, $16 and $10 respectively. The gradients of equal costs are added to the diagram.



The completed graph is on the next page.

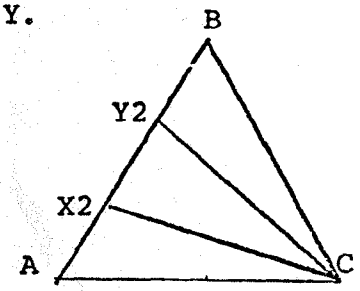

TRIANGULAR CO-ORDINATE 359-32
KEUFFEL & ESSER CO.
B
COST=15
COST=14
Y2
1.5% S
COST=13
Y1
COST=12
COST=11
COST=10
X2
COST=9
2.5% S
COST=8
COST=7
A
X1
C

COMMENTS ON THE GRAPH

If there were no pooling restrictions, the LP would find the most economical way of making product X to be the point X1. The optimal mixture would be 50%A, 50%B and the cost $8. Since product X sells for $9 there is unit profit of $1.

Similarly the optimal mixture for product Y would be at point Y1. The optimal mixture would be 50%B, 50%C and the cost $13. Since product Y sells for $15, there is a unit profit of $2. or double the driving force of product X. The LP will first make all the Y it can, then all the X. If supply of A, B and C are unrestricted, then the maximum demand for X and Y will be satisfied. If there are restrictions on feed, the LP will find its optimal mixture for X in the space between the feasibility line X1, X2 and the breakeven cost gradient $9. Similarly the optimal mixture for Y will be between Y1,Y2 and the breakeven cost gradient $15. We call these spaces the optimizing spaces.

However when the pooling restriction is imposed on A and B they must be used in the same proportions in both products X and Y.



Constant mixtures of A and B are a family of lines from vertex C across the space. Four such lines are CA, CX_2, CY_2, and CB. Note that for our problem there is no constant mixture line that crosses both of the optimizing spaces for products X and Y. Therefore our optimal answer will be either at X1 or at Y1. The unit profit is greater at Y1. However total profit will be obtained by multiplying the unit profit by the demand for that product (assuming unrestricted supply). If the demand for Y is greater than one half of the demand for X then Y1 is optimal. If it is less,then X1 is optimal.

This background in the underlying relationships helps explain the behavior of the LP during recursion.

Behavior of Recursion Model - More Studies

by C. A. Haverly

In our continuing interest in the subject of the behavior of L.P. Models, I have hit upon a rather interesting Recursion Model. It was described in my previous article in the Sigmap Bulletin(1). This article describes further studies made on the model.

This recursion model is a blending model with three components, two products, one quality, and pooling of two of the component streams. The model is interesting because its behavior changes dramatically when there are changes in the input data but without changes in the model structure. Thirty-six case studies were reported in the prior Sigmap article(1). This article presents the results of an additional 92 case studies.

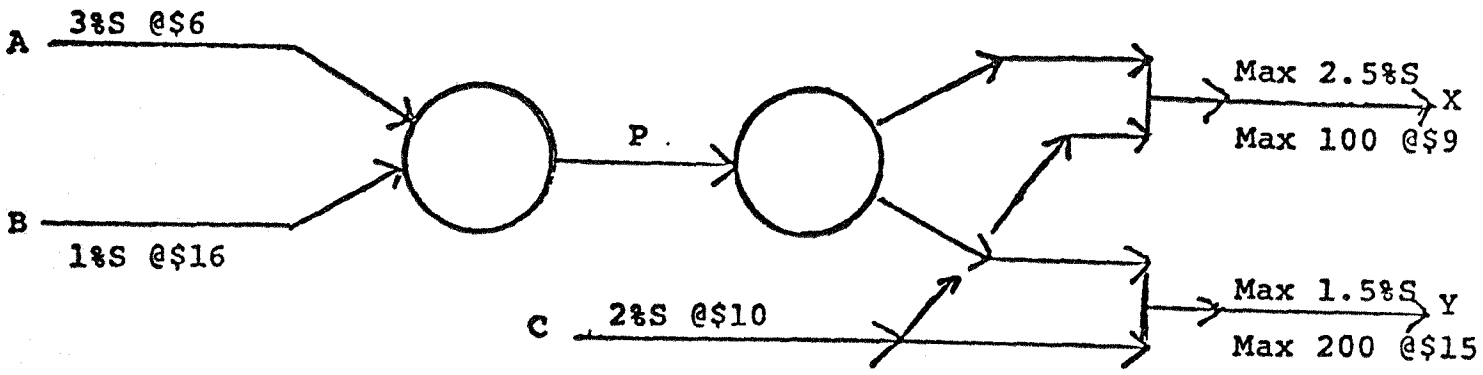
Basic Model

A and B are pooled, and the resulting mixture blended with C to make products X and Y. Sulfur specifications for X and Y must be met. Pertinent data is as follows:

	Cost	Sulfur		Price	Sulfur	Demand
A	$ 6.00	3%	X	$ 9	2.5% max	100 m
B	$16.00	1%	Y	$15	1.5% max	200 m
C	$10.00	2%				

The recursion is done on two coefficients. They are:

f=estimated fraction of pool which goes to produce x, and
p=estimate sulfur quality of the pool.



SECOND SERIES OF RUNS

In the first series of 35 runs, the global optimum was found to be "make only product Y". The input data was such that both the unit profit of making Y and the demand for Y were greater than the corresponding numbers for product X. Thus two major driving forces helped the LP find the correct global optimum.

In the second series we asked the question as to how the model would behave when the driving forces were in competition. We let the unit profit still favor making Y, but changed the quantity driving force to favor X. The driving forces were changed enough so that making X instead of Y becomes the global optimum. This was done by keeping all the data in the model the same except that the maximum demand for X was raised from 100 to 600 (i.e. only the RHS value for XMAX was changed).

We ran 35 cases starting from values for f and p which spanned the entire possible range. The results are given in Table 2.

CONCLUSIONS FROM SECOND SERIES

As expected, Recursion did not find the global optimum quite as often as in the first series where the driving forces were in conjunction (20 out of 35 vs. 23 out of 35). However, when the estimated starting values for p and/or f were good, the global optimum was always found. Only when the values of both p and f were picked badly did the convergence end up at a non-global optimum.

The results reemphasize the need to pick starting value for p and f with care.

THIRD SERIES OF RUNS

In both the first and second series of runs, it was found that the global optimum involved a pool consisting of only one component. The component was B in the first series and A in the second. In the course of the convergence, of course, there were many intermediate optimal solutions where the pool included both A and B.

It was desired to try a series of runs where the global optimum converged to a mixture in the pool. An examination of the tertiary diagram (reproduced in the prior article) indicated that the global optimum could be shifted from Y1 to Y2 (or X1 to X2) by "tilting" the cost gradient. Therefore the third series was set up using exactly the same data as the first series except the cost of component B was changed from $16 to $13. The results of the 35 cases then run with this data is shown in table 3.

CONCLUSIONS FROM THIRD SERIES

The runs showed that Recursion had no special difficulty in converging to the optimum mixture. No more than 5 recursion iterations were required. 27 of the 35 cases converged to the global optimum. As in prior series the only cases converging to non-global optimum were those where the starting estimates of f and p were both bad.

This series had more runs which converged to the global optimum than any of the first three series. It appears that this is because in lowering the price of B to tilt the cost gradient, we also caused the unit profit for production of Y to increase. This extra economic incentive caused Recursion to converge to the global (rather than non-global) optimum over a wider range of starting estimates for f and p.

FOURTH SERIES OF RUNS

In the fourth series of runs we decided to explore the situation suggested by Bill Hart (2); namely that we should try pooling the closely related streams rather than the stream of greatest difference in properties. The model includes stream A with 3% sulfur quality, stream B with 1% and stream C with 2%. Rather than pool A and B as in all the prior runs we decided to explore what would happen if A and C were pooled instead. The global optimum turns out to be 50%A-50%C in the pool. The pool then meets the 2.5% quality specification for product X. The tighter specification 1.5% for product Y can then be met by blending 2 parts of stream B with each part of the pooled material. The optimal solution has A, B, C, X and Y all at non-zero levels.

Cases were run for f between 0 and 1 and for p between 2.0% and 3.0%. The results are shown in table 4.

CONCLUSIONS FROM FOURTH SERIES

Fifteen cases were run. All converged without difficulty and in no case was more than 4 recursion iterations required. Ten of the 15 cases converged to the global optimum while 5 converged to a non-global optimum. In this series a good estimate for f proved to be more effective than a good estimate for p.

OVERALL DISCUSSION

I have used Recursion models on real problems without serious difficulty. However the time has never been available to do much research on these models. Therefore it has been fun to take a model and do rather extensive research on its behavior over a wide range of conditions. There have been a number of surprises. Hopefully, the results will help lead to recursion application models which have good behavior over the broadest possible range of data input.

Like all research, the study answers some questions but raises new ones. Obvious questions left unanswered involve the behavior if we pool B and C, quantifying the effect of economic driving forces, and studying further the interaction between prices and demands. Studies of models with more variables (and hence more alternate optimal solutions) is also likely to prove interesting.

However some general conclusions have been observed. These are discussed in the next section.

OVERALL CONCLUSIONS

1. A user should not assume that the recursion will always find the global optimum.

2. Recursion iterations should be made until both f and p have converged.

3. Good estimates for starting values of f and p are important.

4. The global optimum is more likely found when the economic driving forces are high.

5. The global optimum is more likely found when the unit profit driving force and the quantity driving forces are in conjunction.

References

1. C. A. Haverly, Studies of the Behavior of Recursion for the Pooling Problem, SIGMAP Bulletin #25, December 1978, p. 19-28.

2. W. D. Hart, LP Behavior - Recursion Example Comments, SIGMAP Bulletin #25, December 1978, p. 29-32.

TABLE 2 - LARGE DEMAND FOR X

First Recursion Optimals for Pooling Model Obtained

from Various Starting Values of p and f

Basis of Model: as shown earlier except XMAX changed from 100 to 600

Global optimal: f=1, p=3.0

Input		Optimal Answers											Global	Rec.
e	f	p	A	B	OVER	UNDER	PX	CX	X	PY	CY	Y	Profit	Opt.	Iter.
	0	1.0		100						100	100	200	400	N	1
		1.2		100		20				100	100	200	400	N	2
		1.5		100		50				100	100	200	400	N	2
		2.0		100		100				100	100	200	400	N	2
		2.5	400	200		100	600	-	600	-	200	200	800	*	3
		2.9	267	66		100	333	267	600	-	200	200	1067	*	3
		3.0	250	50		100	300	300	600	-	200	200	1100	*	3
	.3	1.0	71	129	142	-	-	86	86	200	-	200	429	N	3
		1.2	63	137	86	-	-	51	51	200	-	200	377	N	3
		1.5	50	150	-	-	-	-	-	200	-	200	300	N	3
		2.0	-	143	-	143	-	-	-	143	57	200	143	N	2
		2.5	379	221	-	143	600	-	600	-	200	200	586	*	3
		2.9	290	91	-	143	381	219	600	-	200	200	1019	*	3
		3.0	271	72	-	143	343	257	600	-	200	200	1057	*	3
	.5	1.0	100	100	200	-	-	200	200	200	-	200	600	N	3
		1.2	80	120	120	-	-	120	120	200	-	200	480	N	3
		1.5	50	150	-	-	-	-	-	200	-	200	350	N	3
		2.0	-	-	-	-	-	-	-	-	-	-	-	N	1
		2.5	450	150	-	-	600	-	600	-	-	-	300	*	3
		2.9	322	122	-	200	444	156	600	-	200	200	956	*	3
		3.0	300	100	-	200	400	200	600	-	200	200	1000	*	3
	.7	1.0	167	33	333	-	-	467	467	200	-	200	1000	*	4
		1.2	120	80	200	-	-	280	280	200	-	200	720	N	3
		1.5	50	150	-	-	-	-	-	200	-	200	300	N	3
		2.0	-	-	-	-	-	-	-	-	-	-	0	N	1
		2.5	450	150	-	-	600	-	600	-	-	-	300	*	3
		2.9	396	196	-	333	593	7	600	-	200	200	807	*	3
		3.0	367	167	-	333	533	67	600	-	200	200	867	*	3
	1.0	1.0	200	-	400	-	100	500	600	100	100	200	1200	*	4
		1.2	200	-	360	-	75	525	600	125	75	200	1200	*	4
		1.5	200	-	300	-	-	600	600	200	-	200	1200	*	4
		2.0	300	-	300	-	300	300	600	-	-	-	600	*	2
		2.5	300	-	150	-	300	300	600	-	-	-	600	*	2
		2.9	300	-	30	-	300	300	600	-	-	-	600	*	2
		3.0	300	-	-	-	300	300	600	-	-	-	600	*	1

TABLE 3 - LOWER PRICE FOR B

First Recursion Optimals for Pooling Model Obtained

from Various Starting Values of p and f

Basis of Model: original data except cost of B changed $16 to $13.

Global optimum: f=0, p=1.5

	Input														
ase	f	p	A	B	OVER	UNDER	PX	CX	X	PY	CY	Y	Profit	Global Opt.	Re It
72	0	1.0	50	150	100	-	-	-	-	200	-	200	750	*	2
73		1.2	50	150	60	-	-	-	-	200	-	200	750	*	2
74		1.5	50	150	-	-	-	-	-	200	-	200	750	*	1
75		2.0	50	150	-	100	-	-	-	200	-	200	750	*	2
76		2.5	125	175	-	200	100	-	100	200	-	200	875	*	4
77		2.9	103	153	-	280	56	44	100	200	-	200	853	*	4
78		3.0	100	150	-	300	50	50	100	200	-	200	850	*	4
79	.3	1.0	71	129	143	-	-	86	86	200	-	200	814	*	3
80		1.2	63	137	86	-	-	51	51	200	-	200	789	*	3
81		1.5	50	150	-	-	-	-	-	200	-	200	750	*	2
82		2.0	29	171	-	143	-	-	-	200	-	200	600	*	3
83		2.5	4	96	-	143	100	-	100	-	200	200	625	*	5
84		2.9	24	76	-	143	100	-	100	-	200	200	765	*	5
85		3.0	33	84	-	167	100	-	100	17	183	200	783	*	4
86	.5	1.0	100	150	200	-	50	50	100	200	-	200	850	*	3
87		1.2	81	131	120	-	12	88	100	200	-	200	831	*	3
88		1.5	50	150	-	-	-	-	-	200	-	200	750	*	2
89		2.0	-	200	-	200	-	-	-	200	-	200	400	*	3
90		2.5	-	100	-	150	100	-	100	-	150	150	350	*	5
91		2.9	-	100	-	190	100	-	100	-	190	190	550	*	5
92		3.0	-	100	-	200	100	-	100	-	200	200	600	*	5
93	.7	1.0	107	157	214	-	100	-	100	164	36	200	857	*	4
94		1.2	122	173	185	-	100	-	100	195	5	200	872	*	4
95		1.5	50	150	-	-	-	-	-	200	-	200	750	*	2
96		2.0	-	-	-	-	-	-	-	-	-	-	-	N	1
97		2.5	75	25	-	-	100	-	100	-	-	-	125	N	2
98		2.9	66	34	-	57	100	-	100	-	34	34	236	N	3
99		3.0	64	36	-	71	100	-	100	-	43	43	264	N	3
00	1.0	1.0	75	125	150	-	100	-	100	100	100	200	825	*	4
01		1.2	87	138	130	-	100	-	100	125	75	200	838	*	4
02		1.5	125	175	100	-	100	-	100	200	-	200	875	*	4
03		2.0	75	25	50	-	100	-	100	-	-	-	125	N	2
04		2.5	75	25	-	-	100	-	100	-	-	-	125	N	1
05		2.9	75	25	-	40	100	-	100	-	-	-	125	N	2
06		3.0	75	25	-	50	100	-	100	-	-	-	125	N	2

TABLE 4 - POOL A and C

First Recursion Optimals for Pooling Model Obtained

from Various Starting Values of P and f

Basis of Model: Original data except pooling A and C instead of A and B resulting in blending a higher sulfur pool with a low sulfur individual stream.

Global optimum: f=.6, p-2.5%

Input Case	f	p	A	C	OVER	UNDER	PX	BX	X	PY	BY	Y	Profit	Global Opt.	Rec. Iter.
L07	0	2.0	-	100	-	-	-	-	-	100	100	200	400	N	1
L08		2.5	-	250	-	125	100	-	100	150	30	200	600	N	3
L09		3.0	-	250	-	250	75	25	100	175	25	200	600	N	3
L10	.3	2.0	-	100	-	-	-	-	-	100	100	200	400	N	2
L11		2.5	83	83	-	-	100	-	100	66	134	200	433	*	2
L12		3.0	41	167	-	167	100	-	100	108	92	200	517	*	4
.13	.5	2.0	-	100	-	-	-	-	-	100	100	200	400	N	2
.14		2.5	83	83	-	-	100	-	100	66	134	200	433	*	2
.15		3.0	75	100	-	100	100	-	100	75	125	200	450	*	4
.16	.7	2.0	72	107	71	-	100	-	100	79	121	200	457	*	4
.17		2.5	83	83	-	-	100	-	100	66	134	200	433	*	2
.18		3.0	89	72	-	72	100	-	100	61	139	200	421	*	4
.19	1.0	2.0	50	150	50	-	100	-	100	100	100	200	500	*	4
20		2.5	83	83	-	-	100	-	100	66	134	200	433	*	2
21		3.0	100	50	-	50	100	-	100	50	150	200	400	*	4

Second Recurse Iteration

se	f	p	A	C	OVER	UNDER	PX	PB	X	PY	BY	Y	Profit
2A	.481	2.2	93	63	62	-	100	-	100	56	144	200	413
5A	.571	2.43	84	82	12	-	100	-	100	66	134	200	433
6A	.559	2.40	84	82	18	-	100	-	100	66	134	200	432
8A	.621	2.55	83	83	-	8	100	-	100	66	134	200	433
9A	.500	2.25	90	70	50	-	100	-	100	60	140	200	420
1A	.667	2.67	85	80	-	25	100	-	100	65	135	200	430