# Optimum cross-current extraction with product recycle

## D. F. RUDD and E. D. BLUM

D. F. RUDD, *University of Wisconsin, Madison, Wisconin, U.S.A.* and E. D. BLUM, *Pure Oil Company, Crystal Lake, Illinois, and University of Michigan, Ann Arbor, Michigan, U.S.A.*

**Abstract**—The dynamic programming analysis of ARIS *et al.* is extended to the determination of the optimal operating conditions for a class of multi-stage processes with product recycle. The dynamic programming tables for the process with no recycle can be used without modification to determine the optimal operating conditions for the process with recycle. The methods are restricted to recycle back to the first stage, and to certain objective functions.

## 1. INTRODUCTION

IN A recent paper ARIS *et al.* [1] applied the concepts of dynamic programming developed by BELLMAN [2] to the problem of optimum cross-current extraction. There results from this analysis dynamic programming tables from which the optimal operating conditions of the process are obtained by a sequence of table entries and back substitutions. The methods presented are limited to the analysis of processes with no product recycle. This paper shows how these selfsame tables can be used without modification to determine the optimal operating conditions for a certain class of processes with product recycle. The method involves iterative table entries and back substitutions.

Consider first the classical dynamic programming analysis of multi-stage processes with no recycle. The nomenclature differs slightly from that of ARIS. Let the process consist of $N$ stages, and have a process stream flowing from stage to stage. Let $\mathbf{P}$ denote the quality of the process stream and $q$ its flow rate. The elements of $\mathbf{P}$ are the intensive variables such as composition or temperature which describe the quality of the feed stream. $\mathbf{P}_0$ is the quality of the feed to the first stage, $\mathbf{P}_1$ the quality of the process stream leaving the first stage and passing on to the second stage, and so forth until $\mathbf{P}_N$ is the quality of the product stream leaving the stage $N$. The flow rate $q$ is assumed constant.

Each stage transforms the quality of the material entering according to the level of its operating conditions denoted by $\mathbf{W}$. The elements of $\mathbf{W}$ are the operating conditions such as steam pressure or solvent addition rate. The transformation of the process stream quality by the $K$-th stage can be represented by the following:

$$\mathbf{P}_K = T(\mathbf{P}_{K-1}; \mathbf{W}_K) \qquad K = 1, 2, \dots N \quad (1)$$

Associated with the operation of this process is an objective function $O$. It is necessary for the analysis following to restrict the functional dependence of the objective function to the terms $(\mathbf{P}_0 - \mathbf{P}_N)q$, $\mathbf{W}_1$, $\mathbf{W}_2$, $\dots$, $\mathbf{W}_N$. Let the objective function be

$$O = R[(\mathbf{P}_0 - \mathbf{P}_N)q] - \sum_{i=1}^{N} C[\mathbf{W}_i] \quad (2)$$

where $R$ is the value function associated with the change of quality of the process stream and $C$ is the cost of maintaining the stages at their respective operating levels. The choice of operating conditions which maximize the objective function is called the optimal policy. We may write

$$f_N(\mathbf{P}_0, q) = \underset{\mathbf{W}_1,\dots,\mathbf{W}_N}{\text{Max}} \left\{ R[(\mathbf{P}_0 - \mathbf{P}_N)q] - \sum_{i=1}^{N} C(\mathbf{W}_i) \right\} (3)$$

which is the maximum achieved using an optimal policy for an $N$-stage process with feed quality $\mathbf{P}_0$ and feed rate $q$.

This process may be thought of as a single first stage followed by an $N - 1$ stage process. The first stage transforms the feed stream from quality $\mathbf{P}_0$ to quality $\mathbf{P}_1 = T(\mathbf{P}_0, \mathbf{W})$, and this is the feed to the $N - 1$ stage process. Clearly if the whole process is operating with an optimal $N$ stage policy with

respect to the feed quality $\mathbf{P}_0$, the last $N - 1$ stages must be operating with an optimal $N - 1$ stage policy with respect to its feed $\mathbf{P}_1$. Any other policy would only decrease the objective function for the whole process.

Thus

$$f_N(\mathbf{P}_0; q) = \operatorname*{Max}_{\mathbf{W}_1}\{_{JN-1}(\mathbf{P}_1) - C(\mathbf{W}_1)\} \qquad (4)$$

This equation gives $f_N$ once $f_{N-1}$ is known, and involves a maximization only over $\mathbf{W}_1$. Clearly

$$f_1(\mathbf{P}_0, q) = \operatorname*{Max}_{\mathbf{W}_1}\{R(\mathbf{P}_0 - \mathbf{P}_N)q - C(\mathbf{W}_1)\} \qquad (5)$$

The determination of the optimal policy using the dynamic programming algorithm, equations (4) and (5), involves maximizations only over one $\mathbf{W}$ whereas the original problem (equation 6) involves maximizations over $N$ $\mathbf{W}$'s. This reduction in dimension makes the numerical solution for the optimal policy tractable. The results of the computations, normally performed on a digital computer, consist of $f_1(\mathbf{P}, q)$, $f_2(\mathbf{P},q)$, $\ldots f_N(\mathbf{P},q)$, and the corresponding optimal $\mathbf{W}_1$'s. With a given feed quality $\mathbf{P}_0$ and flow rate $q$, entering the table $N$ yields the optimal $\mathbf{W}_1$ and the quality of the stream $\mathbf{P}_1$ which is the feed to the $N - 1$ stage process. Using $\mathbf{P}_1$ as the $\mathbf{P}_0$ for the $N - 1$ stage process, table $N - 1$ gives the optimal $\mathbf{W}_1$ for the $N - 1$ stage process and the quality leaving that stage. This process of table entry and back substitution is repeated until $\mathbf{W}_1$ for a one-stage process is determined and the entire optimal policy obtained. The important feature is that for a given $\mathbf{P}_0$ and $q$ it is possible from these tables to obtain the optimal policy and the resulting product quality $\mathbf{P}_N$. The optimal policy thus obtained is for the process with no recycle. An iterative method of entry and back substitution which is quite similar to the above method will now be developed to determine the optimal policy for the process with product recycle to the first stage.

## 2. Multistage Process with Recycle

Chemical engineering abounds with recycle problems. Under certain condition, which are not unduly restrictive, it is possible to determine the optimal policy for a stage-wise process with product
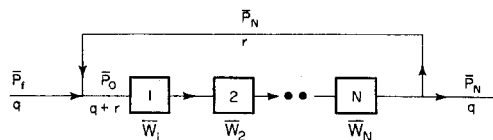


Fig. 1. Multi-stage process with recycle.

recycle from the dynamic programming tables constructed for the process with no recycle. One gets two for the price of one.

Consider the recycle problem where the fresh feed entering the process with a quality $\mathbf{P}_f$ and at a flow rate $q$ is mixed with part of the product stream leaving the process of quality $\mathbf{P}_N$ and recycle rate $r$. (See Fig. 1). The quality of the mixed stream which enters the first stage of the process is

$$\mathbf{P}_0 = \frac{q\mathbf{P}_f + r\mathbf{P}_N}{q + r} \qquad (6)$$

and has a flow rate $q + r$.

The objective function for this recycle process is

$$O = R[(\mathbf{P}_f - \mathbf{P}_N)q] - \sum C(\mathbf{W}_i)$$

The optimal policy then gives

$$f(\mathbf{P}_f, q) = \operatorname*{Max}_{\mathbf{W}_1 \ldots \mathbf{W}_N}\left\{R[(\mathbf{P}_f - \mathbf{P}_N)q] - \sum_{i=1}^{N} C(\mathbf{W}_i)\right\} \qquad (7)$$

but substituting for $\mathbf{P}_f$ from equation (6) into equation (7) gives

$$f_N(\mathbf{P}_f, q) = \operatorname*{Max}_{\mathbf{W}_1 \ldots \mathbf{W}_N}\left\{R[(\mathbf{P}_0 - \mathbf{P}_N)(q + r)] - \sum_{i=1}^{N} C(\mathbf{W}_i)\right\}$$

which is clearly the dynamic programming problem for the sub-process within the recycle stream, which was solved when the process without recycle was considered. The only difficulty encountered is that the quality of the feed to the subprocess $\mathbf{P}_0$ depends on the quality of the product stream $\mathbf{P}_N$ by equation (6). An iterative method of table entry and back substitution is now developed which circumvents this difficulty.

It is now necessary to modify the method of obtaining the optimal policy from the dynamic programming tables to take into account the dependence of $\mathbf{P}_0$ on $\mathbf{P}_N$. Consider the following

method. Choose an initial $P_0^{(0)}$ (possibly $P_f$) and enter the dynamic programming tables according to the methods previously described to obtain the corresponding $P_N^{(0)}$. Using equation (6), the mixing equation, obtain a new estimate of the quality of the process stream entering the first stage.

$$P_0^{(1)} = \frac{qP_f + rP_N^{(0)}}{q + r}$$

and in general

$$P_0^{(i)} = \frac{qP_f + rP_N^{(i-1)}}{q + r} \qquad (8)$$

Now clearly if the limit of the sequence $P_0^{(0)}$, $P_0^{(1)}, \dots, P_0^{(i)}$ exists equation (6) is satisfied and the limiting policy is the optimal policy for the subprocess. From the preceding arguments that policy is the optimal policy for the recycle process. Only the question of convergence of the iterative method of table entry needs answering.

To prove convergence it is necessary to impose certain restrictions on the vectors $P$. Fortunately these restrictions commonly occur physically, and are sufficient but not necessary. That is, the iterative method may converge when the conditions are not satisfied, and must converge when the conditions are satisfied. So much the better. The restrictions are that every element $p$ of the vectors $P$ must satisfy either of the following

$$p_0 > p_N > B_*$$

or

$$B^* > p_N > p_0 \qquad (9)$$

where $B_*$ and $B^*$ are fixed bounds.

Consider the case where the first of these are satisfied.
Then,

$$p_0^{(i)} > p_N^{(i)}$$

and

$$p_0^{(i)} = \frac{qp_f + rp_N^{(i-1)}}{q + r}$$

is a monotonic function of $p_N^{(i-1)}$.
Hence

$$p_f = p_0^{(1)} > p_0^{(2)} > \dots, > p_0^{(i)} > B_*$$

The sequence $p_0^{(i)}$ is a monotone decreasing and bounded from below and convergence is established.

Consider the case where an element of the quality vector satisfies the second condition. Then,

$$p_0^{(i)} < p_N^{(i)}$$

and $p_0^{(i)}$ is a monotonic function of $p_N^{(i-1)}$, hence

$$B^* > p_0^{(i)} > p_0^{(i-1)} > \dots > p_0^{(1)} = p_f$$

The sequence $p_0^{(i)}$ is monotone increasing and bounded from above and convergence is established.

Therefore if every element of the vector $P$ satisfies either one of the two restrictions (equation 9), the iterative method will converge. It may converge in certain cases when equation (9) is not satisfied but no proof is available.

## 3. An Example

The use of this iterative method is now illustrated by considering the addition of a recycle stream to the cross-current extraction problem considered by Aris et al. [1].

A brief description of the process and of the dynamic programming tables developed by Aris is here presented; a detailed description of the development of the tables is to be found in the original article. The cross-current extraction process consists of $N$ equilibrium stages through which a solvent containing a solute passes. The solute is extracted from the solvent by the addition of wash water at each stage. The wash water and solvent are assumed immiscible. The solvent flows from stage to stage at a rate $q$ and its quality is the concentration of solute $x$. The operating variable at each stage is the amount of wash water added $w$. The gross return from the process is the amount of solute extracted by the wash water from the solvent. The costs are those associated with the wash water. The objective function is then

$$O = q(x_0 - x_N) - \lambda \sum w_i$$

where $\lambda$ is a ratio of the return from the extracted solvent and the cost of the wash water. It can be shown that this objective function also corresponds to that for the problem of maximizing the amount extracted with limited total wash water. The optimization problem is to assign the wash water $w_i$ to maximize the objective function for the process.

*Table 1. Dynamic programming table for cross-current extraction*

| $x_0$ | N = 1 | | N = 2 | | N = 3 | |
|---|---|---|---|---|---|---|
| | $x_1$ | $w_1/q$ | $x_1/q$ | $w_1/q$ | $x_1$ | $w_1/q$ |
| 0·03 | 0·02482 | 0·08349 | 0·02628 | 0·05663 | 0·02701 | 0·04429 |
| 0·04 | 0·02799 | 0·17164 | 0·03185 | 0·10490 | 0·03400 | 0·07273 |
| 0·05 | 0·03083 | 0·25405 | 0·03730 | 0·13995 | 0·04086 | 0·09187 |
| 0·06 | 0·03600 | 0·27429 | 0·04160 | 0·18128 | 0·04502 | 0·13619 |
| 0·07 | 0·03820 | 0·34281 | 0·04499 | 0·22747 | 0·05076 | 0·15835 |
| 0·08 | 0·04082 | 0·39457 | 0·04778 | 0·27876 | 0·05611 | 0·18066 |
| 0·09 | 0·04350 | 0·43765 | 0·05375 | 0·28431 | 0·06017 | 0·21266 |
| 0·10 | 0·04503 | 0·50072 | 0·05781 | 0·31106 | 0·06306 | 0·25585 |
| 0·11 | 0·04531 | 0·58475 | 0·05997 | 0·35751 | 0·06740 | 0·28312 |
| 0·12 | 0·04695 | 0·64130 | 0·06014 | 0·42759 | 0·06970 | 0·32877 |
| 0·13 | 0·04930 | 0·68212 | 0·06456 | 0·44750 | 0·07148 | 0·37831 |
| 0·14 | 0·05123 | 0·72493 | 0·06703 | 0·48639 | 0·07600 | 0·40001 |
| 0·15 | 0·05352 | 0·75936 | 0·06702 | 0·55314 | 0·07612 | 0·46230 |
| 0·16 | 0·05597 | 0·78837 | 0·07123 | 0·57383 | 0·07596 | 0·52535 |
| 0·17 | 0·05720 | 0·84103 | 0·07586 | 0·58890 | 0·07740 | 0·55165 |
| 0·18 | 0·05903 | 0·87809 | 0·07597 | 0·65027 | 0·08997 | 0·52965 |
| 0·19 | 0·05996 | 0·92934 | 0·07585 | 0·71424 | 0·08969 | 0·59082 |
| 0·20 | 0·06000 | 1·00000 | 0·07600 | 0·77500 | 0·09000 | 0·64706 |

The dynamic programming tables developed for physical situation which involved a non-linear equilibrium isotherm between solute in wash water and solvent phases, and a $\lambda = 0.05$ are presented in Table 1 for three stages. These tables contain the optimal policy for the process for any $q$ and $x_0$ in the range zero to 0·20. Now clearly from physical reasoning $x_0 > x_N > 0$ and the first condition of equation (9) is satisfied. Convergence of the iterative table entry method for the determination of the optimal policy with recycle is assured.

First consider the normal use of these tables for the cross-current extraction problem with $x_0 = 0.20$, $q = 1$ and no recycle. Entering the Table at $x_0 = 0.20$ and reading over to columns 6 and 7 gives $x_1 = 0.090$ and $w_1 = 0.647$. The process stream from the first of the three stages is the feed to the second stage. Hence $x_0 = 0.090$ for the two stage process. Reading over to column 4 and 5 at $x_0 = 0.090$ gives $x_2 = 0.054$ and $w_2 = 0.284$. This $x_2 = 0.054$ is the feed to the remaining one-stage process and reading columns 2 and 3 at $x_0 = 0.054$ by interpolation gives $x_3 = 0.0328$ and $w_3 = 0.259$. The optimal policy for the no recycle case is $w_1 = 0.647$, $w_2 = 0.284$ and $w_3 = 0.259$, with an associated $f_3 (0.20, 1) = 0.10753$ and 83·6 per cent of the solute extracted. Consider now the use of these tables to obtain the optimal policy with $x_f = 0.20$, $q = 1$ and with a recycle rate of product stream $r = 1$. The iterative method is

*Table 2. Outline of table entry sequence and optimal policy*

| Iteration | $x_0$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|---|
| 0 | 0·2000 | 0·0900 | 0·0540 | 0·0328 |
| 1 | 0·1164 | 0·0689 | 0·0446 | 0·0293 |
| 2 | 0·1147 | 0·0685 | 0·0445 | 0·0293 |

Optimal policy $w_1 = 0.610$  $w_2 = 0.44$  $w_3 = 0.42$
$f_3 (0.20, 1) = 0.0974$
85·5 per cent extracted

identical to that described above except that the resulting product of concentration $x_3$ is recycled back and mixed according to equation (6) with the fresh feed $x_f = 0.20$. It should also be noted that the amount of material flowing through the extractors is $q + r = 2.0$ in this case. The iterative entry and back substitution in the table is outlined in Table 2. The first choice of $x_0^{(c)} = x_f = 0.20$.

Hence with a recycle rate equal to the feed rate 85·5 per cent is extracted with a profit of 0·0974. Without recycle 83·6 per cent is extracted with a profit of 0·1075.

## NOTATION

| | |
|---|---|
| $B^*$, $B_*$ | Upper and lower bounds |
| $C$ | Cost function |
| $f$ | Maximum of objective function |
| $O$ | Objective function |
| $P$ | Quality vector |
| $p$ | Element of quality vector |
| $q$ | Flow rate |
| $R$ | Gross receipts |
| $r$ | Recycle flow rate |
| $T$ | Transformation function |
| $W$ | Operating condition vector |
| $w$ | Wash water rate |
| $x$ | Solute concentration |
| $\lambda$ | Cost ratio |

## REFERENCES

[1] ARIS R., RUDD D. F. and AMUNDSON N. R., *Chem. Engng. Sci.* 1960 **12** 88.
[2] BELLMAN R., *Dynamic Programming* Princeton University Press 1957.

**Résumé**—La programmation dynamique d'Aris et Collab est étendue à l'optimisation des conditions opératoires pour une catégorie de procédés à étages multiples avec recyclage. Les tables de programmation dynamique pour le procédé sans recyclage peuvent être utilisées sans modification pour optimiser les conditions d'opération dans le procédé avec recyclage. Les méthodes sont limitées au recyclage du 1er étage et à certaines fonctions objectives.

**Zusammenfassung**—Die von ARIS *et al.* angewandte Methode des dynamischen Programmierens wird auf die Bestimmung der optimalen Betriebsbedingungen für eine Gruppe vielstufiger Prozesse mit Produktrückführung erweitert. Die Tabellen des dynamischen Programmierens für den rückführungsfreien Prozess können unverändert zur Bestimmung der optimalen Betriebsbedingungen für den Prozess mit Rückführung benutzt werden. Die Methoden sind auf die Rückführung zur ersten Stufe und auf bestimmte Zielfunktionen beschränkt.