

**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE QUIMICA**

**TITULO DEL TEMA**

**CALCULO DE UN SISTEMA DE EVAPORACION  
TRIPLE EFECTO CORRIENTE PARALELA  
POR PROGRAMACION DINAMICA**

62

**NOMBRE DEL SUSTENTANTE**

**JOSE FCO. PABLO CARRERA ANELL**

**CARRERA**

**INGENIERO QUIMICO**

**AÑO**

**1 9 7 4**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

44 59



**QUIMICA**

Jurado asignado originalmente  
según el tema.

PRESIDENTE ENRIQUE GARCIA LOPEZ


VOCAL GERARDO BAZAN NAVARRETE


SECRETARIO ALEJANDRO LOZADA CANIBE

1er. SUPLENTE LUIS ROMERO CERVANTES

2do. SUPLENTE GERARDO RODRIGUEZ ALONSO

Sitio donde se desarrolló el tema: BIBLIOTECA FACULTAD DE QUIMICA.

Nombre completo y firma del sustentante:  CARRERA ANELL JOSE FCO. PABLO

Nombre completo y firma del asesor del tema:  ALEJANDRO LOZADA CANIBE

Nombre completo y firma del supervisor técnico: ( NO LO HAY )

CON TODO MI CARINO, ADMIRACION Y RESPETO A MIS PADRES:

ALICIA ANELL DE CARRERA

Y

RAFAEL CARREERA GOMEZ

CON TODO MI AGRADECIMIENTO Y RECO  
NOCIMIENTO: A MI HERMANO GABRIEL.

A MIS HERMANOS :

GENOVEVA

REBECA

Y

RAFAEL

**AGRADESCO AL INGENIERO ALEJANDRO LOZADA  
LA VALIOSA AYUDA QUE ME HA PRESTADO PARA  
LA ELABORACION DE ESTA TESIS.**



## CONTENIDO

### INTRODUCCION

#### CAPITULO I.- CONCEPTOS EMPLEADOS EN ESTUDIOS DE OPTIMIZACION.

- 1.- Breve Bosquejo Histórico sobre la Teoría de Optimización.
- 2.- Modelos Matemáticos: Formulación y Clasificación.
- 3.- Selección y Formulación de funciones Objetivo.
- 4.- Grados de Libertad.
- 5.- Simulación.

#### CAPITULO II.- PROGRAMACION DINAMICA Y TECNICAS DE OPTIMIZACION.

- 1.- Introducción, breves comentarios sobre la Historia de la Programación Dinámica.
- 2.- Presentación Formal y Desarrollo de las Ecuaciones de Recursión.
- 3.- Presentación Informal e Intuitiva.
- 4.- Técnicas de Optimización.

#### CAPITULO III.- APLICACION DE LA PROGRAMACION DINAMICA A UN PROBLEMA ESPECIFICO.

- 1.- Presentación del Problema.
- 2.- Formulación del Modelo Matemático del Sistema.
- 3.- Formulación de la Función Objetivo.
- 4.- Formulación de las Ecuaciones de Recursión.
- 5.- Técnica de Optimización Empleada.

#### CAPITULO LV.- PRESENTACION DE RESULTADOS Y CONCLUSIONES.

1.- Presentación de Resultados.

2.- Conclusiones.

#### APENDICE

1.- Descripción del Programa.

2.- Listado del Programa.

BIBLIOGRAFIA.

## INTRODUCCION

En años recientes, la ciencia aplicada a alcanzado un alto nivel de competencia en el diseño industrial, científico y militar. Este progreso ha sido ayudado, por un crecimiento correspondiente en la ciencia de las -- computadoras.

En el pasado, diseños incorporando grandes márgenes de seguridad -- fueron aceptables; en la actualidad ya no lo son, debido a que la tecnología moderna avanza bastante rápido y es a la vez, altamente compleja y extremadamente competitiva. De ahí que en nuestros días el éxito sólo co rresponda a aquéllos que tienen la capacidad de hacer "las mejores deci- siones", más rápida y correctamente. Siendo la selección de "las mejores" decisiones, soluciones o condiciones para alcanzar el mejor resultado -- para una situación dada, el proceso colectivo conocido como OPTIMIZACION . Resulta obvio por lo tanto, la importancia de la misma en las activida des industriales de nuestros tiempos.

Son los anteriores, unos de los argumentos que motivaron el tema de la presente tesis y estimularon el desarrollo de la misma. Habiéndose se leccionado del amplio campo de estudio que comprende la TEORIA DE OPTIMI ZACION, la aplicación de la PROGRAMACION DINAMICA a un sistema de evapo- ración triple efecto, corriente paralela, para el proceso de concentra- ción de soluciones acuosas de Hidróxido de Sodio.

Cualquier estudio de optimización, tal como es el que se pretende -- realizar, comprende tres facetas importantes: 1.- Definición de un obje- tivo adecuado. 2.- El problema debe expresarse, por un sistema capáz de- ser representado por un modelo matemático, por medio del cual se pueda -- simular la función objetivo en términos de las variables del sistema. 3.- Optimización del sistema. Para el estudio que pretendemos realizar el -- sistema ya se encuentra definido, correspondiendo como ya se dijo a un -- evaporador triple efecto, corriente paralela.

Con el objeto de que los temas a ser tratados sigan la misma secuen- cia lógica que se acaba de mencionar, la organización de esta tesis será la siguiente:

Primersmente se presentarán y discutirán los principios para la formula-  
ción de funciones objetivo y modelos matemáticos; a continuación se pre-  
sentrá la teoría y se desarrollarán las ecuaciones de recursión de la-  
programación dinámica; concluyendose con un breve resumen de los méto-  
dos de optimización standard de una sola etapa.

Una vez que ya se hayan presentado todos los conceptos que se mane-  
jarán en nuestro estudio de optimización, a continuación procederemos -  
a aplicarlos al cálculo óptimo del evaporador triple efecto para concen-  
trar soluciones acuosa de Hidróxido de Sodio; formulando consecuentemen-  
te : la función objetivo y el modelo matemático adecuados para el siste-  
ma, desarrollando las ecuaciones de recursión de la Programación Dinámi-  
ca y seleccionando finalmente una técnica de optimización de una sola -  
etapa apropiada.

La solución del problema propuesto por Programación Dinámica, im-  
plica la ejecución de un número de operaciones, tal que resulta imprác-  
tico pensar resolverlas sin recurrir a la ayuda de la computadora digi-  
tal. Siendo necesario por lo tanto, la formulación e implementación de-  
un programa de computadora para la solución del algoritmo de la progra-  
mación dinámica para el problema bajo consideración. En un apéndice al-  
final de esta tesis se describirá la formulación y estructura del pro-  
grama antes mencionado y se anexará un listado del mismo.

Una vez que el mencionado programa haya "corrido" exitosamente, se-  
dispondrá de la información suficiente para la formulación de las con-  
clusiones resultantes, de la aplicación de la programación dinámica al-  
problema bajo estudio.

Es precisamente con la formulación de estas conclusiones, con lo -  
que se termina el contenido de la presente tesis.

Ahora bien, dada la naturaleza del sistema que se pretende optimi-  
zar, resulta obvia la necesidad de contar con un buen conocimiento de -  
las operaciones unitarias de transferencia de calor, para el cálculo --  
del sistema de evaporación. Esta tesis no incluye un capítulo para la -  
presentación de dichas operaciones, porque éstas se encuentran excelen-  
te y extensivamente tratadas en la literatura. En la Bibliografía gene-  
ral al final de esta tesis se darán referencias específicas para el es-

tudio de las operaciones unitarias de transferencia de calor.

Para concluir esta breve introducción, mencionaremos que se ha se leccionado la evaporación de una solución acuosa de Hidróxido de Sodio , sólo por la razón de que las propiedades de la misma son bien conoci das y se encuentran disponibles en la literatura.

CAPITULO 1

CONCEPTOS EMPLEADOS EN ESTUDIOS DE OPTIMIZACION

1.1 BREVE BOSQUEJO HISTORICO.- Aunque fué la decada pasada en donde se observó el nacimiento y desarrollo explosivo de las técnicas de optimización, para la formulación y solución de modelos matemáticos de optimización, debe decirse sin embargo, que los problemas de optimización caracterizados por modelos matemáticos se remontan a los antiguos — griegos y aún más.

El desarrollo formal de la teoría de optimización parte con el nacimiento del cálculo, la teoría fué desarrollada para modelos conteniendo variables continuas y funciones diferenciables. La teoría no fué aplicable para modelos conteniendo un gran número de variables, sin embargo, los problemas de aquéllos tiempos eran suficientemente caracterizados con pocas variables.

Por los años de 1940'S , hubo un rearreglo y un cambio de dirección en el estudio de la teoría de optimización, debido fundamentalmente a dos hechos importantes: el trabajo de científicos y matemáticos en problemas de operación militar y la invención y desarrollo de la computadora digital.

El tratamiento científico dado a los problemas militares antes y durante la guerra y posteriormente a problemas industriales, dió origen al campo de estudio conocido como Investigación de Operaciones.

Con el desarrollo de las computadoras digitales de alta velocidad, comenzó a ser práctico pensar resolver problemas conteniendo cientos y aún miles de variables.

Lo anterior estimuló el estudio de los esquemas de optimización iterativos y posteriormente condujo al desarrollo de la programación lineal, de la programación dinámica y de varios métodos de búsqueda.

Con el objeto de diseñar ó operar un sistema, es necesario disponer de un modelo que lo represente adecuadamente y una función objetivo (criterio) que oriente nuestras decisiones, temas que serán descritos a continuación. Finalmente es necesario seleccionar una técnica de optimización que para el caso particular bajo consideración es la pro-

granación dinámica, debido a que el objetivo fundamental de la presente tesis es la aplicación de la misma a un problema específico.

- 1.2 FORMULACION Y CLASIFICACION DE LOS MODELOS MATEMATICOS.- Los modelos son representaciones de la realidad, si fueran igualmente complejos y difíciles de controlar que la misma, no habría ninguna ventaja en usarlos. Afortunadamente, generalmente siempre se puede construir modelos más simples que la realidad y que permiten explicar y predecir un fenómeno con un alto grado de precisión.

Tres tipos de modelos son comúnmente usados: modelos Icónicos, Análogos y simbólicos ó matemáticos. La atención será enfocada hacia la formulación y clasificación de los modelos matemáticos, siendo los dos primeros tipos de modelos ampliamente descritos en la literatura como en : "FUNDAMENTALS OF OPERATION RESEARCH", por Russell L Ackoff and Maurice W Sasieni editado por John Wiley.

MODELOS MATEMATICOS.- Un modelo matemático es la representación en forma abstracta o simbólica de un problema; usa letras, números y otros tipos de símbolos para representar variables y las relaciones entre ellas. Son los modelos matemáticos, los más generales y más abstractos.

Los componentes básicos de un modelo matemático son :

1.- Las variables  $D = (D_1, D_2, \dots, D_n)$  que son aquéllos factores que pueden ser manipulados para alcanzar el objetivo deseado, estas variables son comúnmente referidas como variables independientes o de decisión.

2.- Los Parámetros.-  $Y = (Y_1, Y_2, \dots, Y_N)$  que son aquéllos factores que afectan el objetivo, pero que no son controlables.

3.- La medida de efectividad, utilidad o retorno asociada con los valores particulares de los parámetros y variables de decisión; la medida de efectividad también conocida como función objetivo, es una función de las variables de decisión y de los parámetros que puede representarse por :  $R = R(D, Y)$

En este mismo capítulo y en una sección por separado se dará la debida atención a la selección y formulación de funciones objetivo.

4.- La Región de Factibilidad ( $S$ ). En la mayoría de las circuns-

tancias las variables de decisión están limitadas en cuanto a los valores que ellas pueden tomar, los valores posibles para las variables de decisión deberán estar contenidos en el conjunto limitante  $S$ , de tal forma que  $(D \in S)$ .

Las limitaciones pueden clasificarse fundamentalmente en dos tipos: Internas y Externas; siendo las restricciones Externas un conjunto de especificaciones rígidas e incontrolables fijadas por un agente fuera de nuestra jurisdicción como podría ser la competencia, el gobierno o la naturaleza. Por otro lado las restricciones Internas son fijadas por el diseñador, un gran número de ellas se derivan de las expresiones de las leyes de conservación de la masa, momentum y energía, así como de relaciones de diseño.

Consideraciones sobre la formulación de modelos.- Un aspecto importante en la formulación de los mismos es el arreglo de las ecuaciones; se ha encontrado por experiencia que si éstas se arreglan en una secuencia lógica de causa y efecto, el modelo es estable. Esta secuencia se llama "ORDEN NATURAL", porque invariablemente va paralela a la secuencia de causa y efecto que se encuentra en la naturaleza.

Un modelo realístico es el factor más importante en cualquier estudio de optimización, es de poco provecho efectuar estudios de optimización con un modelo matemático que no representa adecuadamente al sistema ya existente o por diseñar.

Generalmente cada vez que se desea elaborar un modelo, se presenta el problema de manejar dos objetivos en conflicto: hacer un modelo más fácil de resolver que sea posible y al mismo tiempo hacerlo lo más preciso que se pueda; por lo tanto, al construir un modelo generalmente se buscará simplificar la realidad pero sólo a un punto donde no haya una pérdida significativa de precisión.

Obviamente hay un punto de balance en el manejo de estos objetivos conflictivos y este balance se verá fuertemente afectado por la potencia de las técnicas de solución; mientras mejores sean éstas, permitirán el uso de modelos más complicados.

Varios son los procedimientos de simplificación que pueden ser intentados: las Variables de Decisión y los Parámetros que aparentemente tienen un efecto despreciable en la función objetivo se pueden elimi-



nar; la naturaleza de las variables puede ser cambiada de discreta a continua o viceversa; una simplificación obvia es aproximar las funciones objetivo y las restricciones digamos por relaciones lineales.

**Clasificación de los Modelos Matemáticos.**— Es difícil presentar una clasificación mutuamente exclusiva y colectivamente exhaustiva de los modelos matemáticos, pero es útil hacer algunas distinciones.

**Modelos Determinísticos y Probabilísticos.**— Un modelo Determinístico es aquel en que el valor de la función objetivo se encuentra definido sin ambigüedades especificando los valores de las variables de decisión, no tiene variables incontrolables o aleatorias; por otra parte un modelo Probabilístico es aquel que contiene variables aleatorias — que no pueden ser controladas y cuyos valores son dados por distribuciones de probabilidad.

La naturaleza continua o discreta de las variables es otro modo de clasificación; básicamente una variable continua puede tomar cualquier valor real en un intervalo, mientras que una variable discreta está restringida a un número de valores finito.

Formas distintas de la función objetivo y de las restricciones — producen otra división de los modelos matemáticos, esta clasificación comprende los modelos lineales y no lineales.

Una última clasificación relativa a las formas distintas de las funciones objetivo, es la que se refiere a modelos matemáticos con funciones objetivo de varias variables que son separables y aquéllos en que no lo son; íntimamente relacionados a la noción de separabilidad están los modelos de una sola etapa y los modelos multietapas. En un proceso de decisión de una sola etapa, todas las decisiones son hechas simultáneamente, mientras que en un proceso de decisión en multietapas las decisiones son hechas secuencialmente.

Resulta razonable suponer que es más práctico hacer las decisiones una por una que hacerlas todas al mismo tiempo, siendo la anterior la razón de ser de la programación dinámica.

La gran ventaja de los modelos matemáticos es su generalidad y fa

cilidad de manipulación, cualquier clase de análisis de sensibilidad - tal como cambiar los valores de las variables, parámetros, restricciones y aún cambiar las relaciones funcionales es fácilmente efectuado - cuando hay un modelo matemático del sistema.

- 1.3 SELECCION Y FORMULACION DE FUNCIONES OBJETIVO.- Son muchos y variados los problemas de optimización que pueden presentarse; todos ellos teniendo la característica de que al menos una solución existe, tienen sin embargo, un número infinito de soluciones. El papel de la optimización es la selección de entre la multitud de soluciones potenciales, de aquella solución que es la mejor con respecto a un criterio bien definido; la selección de este criterio u objetivo es por esto un paso esencial de cualquier estudio de optimización, puesto que hasta que no se haya decidido sobre algún objetivo en particular, ninguna mejora puede ser obtenida de ningún estudio.

El criterio u objetivo esencialmente determina la información requerida en la optimización; así una investigación de las condiciones óptimas que produzcan la mayor ganancia, en la etapa de diseño necesita encontrar los costos de capital y de operación así como las "ventas" para el producto generado por el proceso. Por otro lado si el objetivo fuera minimizar el costo, entonces no sería necesario incluir las "ventas", de igual manera, las variables de diseño y de capital de un sistema ya existente no serían requeridas.

El objetivo puede cambiar de un problema a otro, pero para propósitos industriales puede considerarse fundamentalmente de dos tipos : económico y técnico. Normalmente sin embargo, la mayoría de las optimizaciones industriales necesitan efectuarse dentro de un marco de referencia económico.

Muchos objetivos han sido aplicados para la determinación de las mejores condiciones bajo las cuales diseñar u operar un sistema, la naturaleza de esos criterios depende del proyecto en mano, difiriendo -- quizás en las etapas de investigación, desarrollo y diseño; el efecto de imponer objetivos diferentes es producir soluciones óptimas diferentes.

En muchos casos o situaciones prácticas, más de un criterio necesitan ser satisfechos, así puede ser necesario obtener un cierto nivel de rentabilidad en un proceso dado, con una inversión mínima u obtener la máxima cantidad de producto con la mínima concentración de algún -- producto indeseable; con esos objetivos múltiples suele presentarse la posibilidad de conflicto, de tal forma, que en el análisis final puede requerirse el sacrificio de alguno de los objetivos. .

Una parte importante del análisis final es por éso un examen de la interdependencia de los criterios, de tal forma que las consecuencias-completas de seleccionar un objetivo particular puedan ser visualiza--das. Normalmente un criterio es escogido como el objetivo primario, -- con todos los demás considerados como secundarios y a ser satisfechos--de ser posible.

En general la intención principal es la maximización de la ganancia, a ser obtenida con la inversión de capital más pequeña posible y al nivel más satisfactorio de todos los otros objetivos. .

1.4 GRADOS DE LIBERTAD.- Una de las representaciones más comunes de un sistema, es la de bloques o unidades interconectadas por las corrientes - del sistema, en esta representación por medio de bloques, se distin--guen fundamentalmente tres tipos de "corrientes" : "corrientes de en--trada", que son aquéllas corrientes que entran al sistema del exterior , "corrientes de salida", que se originan en el sistema y salen al exterior, "corrientes interconectantes", que se originan y terminan en - el sistema; este sistema completo recibe ciertas entradas y produce ci--ertas salidas. Ya que cada elemento del sistema tiene una base técnica de diseño o modelo dado por un conjunto de ecuaciones de transformaci--ón, las salidas son determinadas una vez que las entradas son especi--ficadas. Es esta determinación la que nos permitirá discutir los grados--de libertad del sistema.

Ya que la especificación de la entrada determina la salida del -- sistema, el número de grados de libertad, definidos como el número de--variables de entrada que pueden ser alteradas independientemente, pro--duciendo cambios en la salida o respuesta del sistema, pueden represen--tarse por :

$$D = \sum_{k=1}^p Q_k - S - L$$

- donde :
- D - Número de grados de libertad.
  - $Q_k$  - Dimensionalidad de la  $k$ -ésima entrada.
  - S - Número de componentes de entrada que no se encuentran bajo el control del optimizador.
  - L - Número de grados de libertad perdidos por especificaciones impuestas al producto.
  - Q - Número de entradas.

Claramente el primer término del segundo miembro, representa la suma de todas las variables de entrada posibles que determinan el comportamiento del sistema; sin embargo, observamos que ciertas de estas variables, S en número, están fuera del control del optimizador y no pueden ser manipuladas libremente, por eso del total del número de componentes de entrada únicamente  $\sum_{k=1}^Q Q_k - S$  serán disponibles para variación.

Si algún agente externo especifica alguna de las propiedades de salida, para cumplir con estas especificaciones, tendremos que usar una ó más de las variables de entrada que estamos en libertad de manipular; ya que las ecuaciones de transformación no pueden ser invertidas, el número de variables manipuladas L requeridas para traer el producto a especificación no pueden determinarse sin referirse al problema en particular. Por lo tanto, habrá una pérdida de grados de libertad adicionales al menos tan grande como el número de especificaciones del producto.

Con el objeto de efectuar estudios de optimización, es necesario que se tenga un sistema indeterminado con  $D > 0$ , de otra manera, no será posible obtener un rango de soluciones; esos problemas tienen en principio un número infinito de soluciones y el objetivo de la optimización es la selección de aquella que es la mejor con respecto a un criterio dado. Si  $D = 0$  habrá una solución única del sistema, éste es llamado un sistema definido; en un sistema sobrediseñado  $D < 0$ , hay más entradas especificadas de las que existen y ninguna solución puede ser obtenida.

1.5 SIMULACION.- Una vez que el objetivo en el estudio de optimización de un sistema ha sido seleccionado, se puede relacionar a las variables del sistema por un proceso llamado simulación.

En gran parte del presente capítulo, se ha recalcado la conveniencia y necesidad de contar con un modelo matemático del sistema bajo consideración.

La derivación de esos modelos es un paso esencial, ya que la función objetivo no puede de otra forma ser expresada en términos de las variables del sistema; en otras palabras, la respuesta de la función objetivo a los cambios en las variables del sistema debe poder ser calculada, para que la optimización comience.

Si la función objetivo es de naturaleza técnica, la simulación se encuentra completada, las variables técnicas tales como peso, consumo de combustible, etc. aparecen en al menos una de las ecuaciones de transformación, de hecho, el objetivo técnico es normalmente uno de los resultados y es dado directamente.

Por otra parte, la simulación de una función objetivo económica es más compleja, aunque no generalmente difícil; el primer paso es la reducción del criterio a sus componentes económicos, así una ganancia o rentabilidad depende de las (ventas), menos varios (costos), mientras que la inversión depende de los costos de compra de los componentes individuales, estos componentes de los costos pueden en un segundo paso ser relacionados a las variables técnicas adecuadas mediante datos de costos adecuados, esto significa que dadas las entradas al sistema, las variables técnicas y los costos, los valores de la función objetivo pueden ser evaluados.

Una vez que el problema de optimización ha sido escrito en alguna forma matemática, la situación física deja de tomarse en cuenta, puesto que ésta sólo necesita ser considerada en la etapa de construcción del modelo; el problema de optimización es puramente matemático y a él nos enfocaremos en el capítulo siguiente.

## C A P I T U L O 2

### PROGRAMACION DINAMICA Y TECNICAS DE OPTIMIZACION

#### 2.1 INTRODUCCION Y BREVES COMENTARIOS SOBRE LA HISTORIA DE LA PROGRAMACION

DINAMICA.- Habiendo construido un modelo matemático adecuado, debemos seleccionar una técnica de optimización para resolver el mismo. La manera como se determina una solución óptima, depende por supuesto, de la forma de la función objetivo y restricciones, naturaleza y número de variables, la clase de facilidades de cálculo disponibles y la experiencia.

A menudo antes de efectuar la optimización, es aconsejable hacer algunas transformaciones y cambios de variables, teniendo cuidado de preservar las propiedades del modelo completamente; de tal forma, que el modelo transformado tenga la misma solución óptima que el problema original, pero que sea más fácil de optimizar.

Básicamente la programación dinámica es una de esas transformaciones, toma un proceso de decisión secuencial o multietapas conteniendo muchas variables interdependientes y lo convierte en una serie de problemas de una sola etapa, cada uno conteniendo únicamente unas pocas variables. La transformación es invariante en lo que se refiere al número de soluciones posibles y el valor de la función objetivo asociado con cada solución posible, es conservado. La transformación está basada en el principio aparentemente obvio conocido como PRINCIPIO DE OPTIMIDAD que establece: "Una política óptima tiene la propiedad de que si el estado y la decisión inicial son óptimos, las decisiones siguientes deberán constituirse en una política óptima si es que parten del resultado de la primera decisión".

Puede decirse que mediante el uso de la programación Dinámica un problema con  $N$  variables de decisión, puede ser transformado en  $N$  subproblemas, cada uno conteniendo únicamente una variable de decisión

Problemas de ciertas áreas tales como teoría de inventarios, distribución de recursos, teoría de control y diseño de ingeniería Química, han sido particularmente fértiles para las aplicaciones de la programación Dinámica. La propiedad básica de esos sistemas es que el

cálculo de las decisiones óptimas puede ser hecho secuencialmente. El método de cálculo secuencial es la esencia de la programación Dinámica.

La programación Dinámica fué practicada mucho antes de que fuera llamada como tal, el trabajo de Wald's sobre teoría de Decisión secuencial, contiene la semilla del procedimiento de la programación Dinámica. Los trabajos de DVORETZKY, KIEFER Y WOLFWIZ, sobre teoría de inventarios, también contienen el espíritu de la programación dinámica.

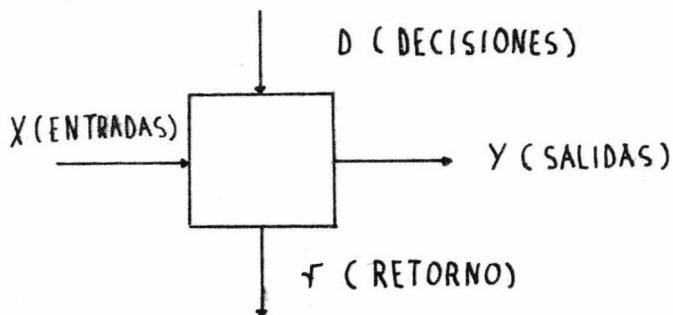
Indudablemente, sin embargo, Richard Bellman es el padre de la programación Dinámica, con sus trabajos de investigación en la Rand Corporation. Es a él a quién se debe el principio de optimidad y el nombre poco descriptivo de: "PROGRAMACION DINAMICA", un nombre más adecuado - como veremos a continuación sería: Optimización Recursiva.

2.2 PRESENTACION FORMAL DE LA PROGRAMACION DINAMICA Y DESARROLLO DE LAS ECUACIONES DE RECURSION.- Para resolver un problema complejo por programación dinámica, lo rompemos en una serie de problemas más pequeños : Descomposición y después combinamos los resultados de las soluciones de los problemas más pequeños para obtener la solución del problema completo: Composición

La programación dinámica es especialmente adecuada para la solución de problemas multietapas seriados, pues toma ventaja de una propiedad estructural de los mismos: su flujo de información no exhibe recirculación. En estos sistemas un cambio en el diseño de un componente dado, puede influenciar únicamente los componentes "Corriente Abajo", siendo el término "Corriente Abajo", referido al flujo de información el cual puede coincidir o no con el flujo de material en el diagrama de flujo del proceso.

Para continuar con el procedimiento multietapas empleado por la Programación dinámica y desarrollar las ecuaciones de recursión de la misma, comenzaremos por estudiar las propiedades de una etapa típica -

de un sistema de decisión multietapas seriado. Un sistema de una etapa es representado por un bloque que está caracterizado por cinco factores:



1.- Un estado de Entrada  $X$  que da toda la información relevante sobre las entradas al bloque ( $X$  es llamada el estado inicial, en cuanto da una descripción del sistema al comienzo de la etapa).

2.- Un Estado de Salida  $Y$  que da toda la información relevante sobre las salidas del bloque ( $Y$  es llamada el estado final, en cuanto da una descripción del sistema al final de la etapa).

3.- Una variable de decisión  $D$  que controla la operación del bloque.

4.- Un Retorno de la etapa  $f$  que es una variable escalar que mide la utilidad del bloque y que es una función de entradas, decisiones y salidas:

$$f = f(X, D, Y)$$

5.- Una Transformación para la etapa  $t$  que expresa cada componente del estado de salida, como una función del estado de entrada y de las decisiones, ésta es:

$$Y = t(X, D)$$

La diferencia matemática entre los estados de entrada y de salida es que el estado de salida ( $Y$ ) es una función de la entrada ( $X$ ) y de las decisiones, pero frecuentemente la diferencia es artificial, — por ejemplo es lo mismo si escribimos:

$$Y = X - D \quad \text{ó} \quad X = Y + D$$

O sea que tanto  $X$  como  $Y$  pueden seleccionarse como estado de entrada.



Puede emplearse la ecuación de transformación para eliminar  $y$  del retorno de la etapa, en particular :

$$y = t(X, D)$$

Es substituído en :

$$r = f(X, D, y)$$

Para producir :  $r = f(X, D, t(X, D))$

La ecuación anterior nos dice que las únicas variables independientes que afectan el retorno de la etapa, son  $X$  y  $D$  ; ya que dados valores para  $X$  y  $D$  el valor de  $y$  se encuentra fijado por la ecuación de transformación  $t$  . Esto conduce a un valor único de  $r$  , para indicar esta dependencia reescribiremos :

$$r = \bar{r}(X, D, t(X, D))$$

Simplemente como :  $r = f(X, D)$

El problema de optimización del estado inicial de una etapa es - encontrar el retorno máximo de la etapa como una función del estado - de entrada. Denotando  $f(x)$  como retorno óptimo y  $D^* = D(x)$  como la política óptima de decisión, se tiene :

$$f(X) = \bar{r}(X, D(x)) = \bar{r}(X, D^*) = \max_D \bar{r}(X, D) \geq \bar{r}(X, D)$$

Si estuviéramos en libertad de escoger  $X$  , élla sería considerada también una variable de decisión, entonces:

$$f(x^*) = \max_X f(X) = \max_{X, D} \bar{r}(X, D)$$

En ocasiones se desea obtener el retorno óptimo como una función del estado de salida ( $y$ ). Suponiendo que es posible determinar  $X$  como una función de  $y$  y  $D$  por inversión de la ecuación de transformación :  $y = t(X, D)$

Para obtener:  $X = \bar{t}(y, D)$

Entonces  $X$  se puede eliminar de la función de retorno, así, el retorno de la etapa puede ser expresado como una función de únicamente las decisiones y las salidas como:

$$r = \bar{r}(\bar{t}(y, D), D, y) = \max_D \bar{r}(y, D)$$

El problema de optimización del estado final de una etapa, es seleccionar  $D$  como una función de  $y$  para minimizar  $r$  . Denotando -  $f(y)$  como retorno óptimo y  $\bar{D} = \bar{D}(y)$  como la política óptima tenemos: -

Generalmente, si el retorno óptimo es determinado como una función de los estados de entrada y salida, entonces no habrá decisión, suponga que :  $Y = t(X, D)$

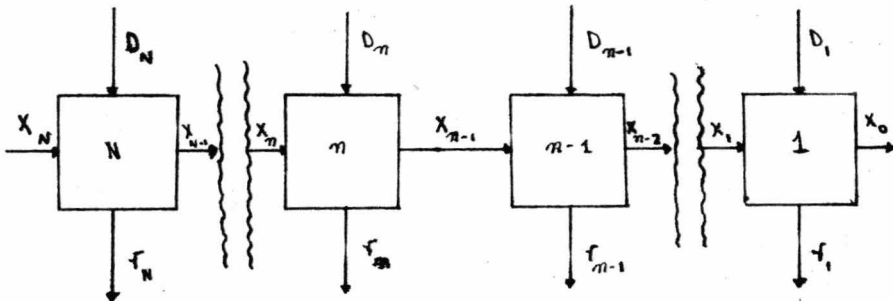
Puede ser resuelta para  $D$  en términos de  $X$  y  $Y$ , ésto es:  $D = \hat{t}(X, Y)$ . Entonces el retorno óptimo como una función de los estados de entrada y salida es:

$$f(X, Y) = \hat{r}(X, \hat{t}(X, Y))$$

Si únicamente algunos de los componentes de  $D$  pueden ser eliminados por el uso de  $\hat{t}$ , los componentes restantes son todavía variables de decisión.

Para propósitos de nomenclatura, que será igualmente válida en el estudio de los sistemas seriados multietapas, llamaremos a  $f(x)$  el problema de optimización del estado inicial,  $f(y)$  problema del estado final y  $f(x, y)$  el problema de los estados inicial-final.

Sistemas de Decisión Multietapas Seriados.- Un sistema multietapas seriado, consiste de un conjunto de etapas unidas en serie de tal forma, que la salida de una etapa se convierte en la entrada de la siguiente :



Para la etapa general  $n$  ( $n = 1, 2, \dots, N$ ) del sistema de  $N$  etapas la ecuación de transformación es :  $X_{n-1} = t_n(X_n, D_n)$  y el retorno de la etapa es :  $r_n = r_n(X_n, D_n)$

Se ha eliminado el estado de salida de la función de retorno, -- por los mismos argumentos que se presentaron en la discusión de un --

sistema de una sola etapa. Ninguna limitación ha sido puesta sobre la forma de las funciones  $f_n$  y  $t_n$ . Sin embargo, la estructura de un sistema seriado multietapas, implica algunas suposiciones importantes.

De las transformaciones se sigue que  $X_n$  depende únicamente de las decisiones hechas anteriormente a la etapa  $\pi (D_{n+1}, \dots, D_N)$  y  $X_N$ ,

$$\begin{aligned} \text{, esto es: } X_n &= t_{n+1}(X_{n+1}, D_{n+1}) = t_{n+1}(t_{n+2}(X_{n+2}, D_{n+2}), D_{n+1}) \\ &= t_{n+1}(X_{n+2}, D_{n+2}, D_{n+1}) = t_{n+1}(t_{n+3}(X_{n+3}, D_{n+3}), D_{n+2}, D_{n+1}) \\ &= t_{n+1}(X_N, D_N, \dots, D_{n+1}) \end{aligned}$$

Combinando las ecuaciones anteriores con la función de retorno, se observa que el retorno de la etapa  $\pi$  depende únicamente de las decisiones:  $(D_n, D_{n+1}, \dots, D_N)$  y  $X_N$ . Esto es:

$$f_n = f_n(X_n, D_n) = f_n(t_{n+1}(X_N, D_N, \dots, D_{n+1}), D_n)$$

O en otras palabras,  $D_n$  únicamente afecta el retorno de las etapas 1 a  $n$ .

El retorno total  $R_N$  de las etapas 1 a  $N$  es alguna función de los retornos individuales de las etapas, escrita como:

$$\begin{aligned} R_N(X_N, X_{N-1}, \dots, X_1, D_N, D_{N-1}, \dots, D_1) \\ = g[f_N(X_N, D_N), f_{N-1}(X_{N-1}, D_{N-1}), \dots, f_1(X_1, D_1)] \end{aligned}$$

Sin embargo, como se acaba de explicar,  $(X_{N-1}, \dots, X_1)$  pueden ser eliminadas de los retornos individuales de las etapas y consecuentemente del retorno total. Por lo tanto, otra alternativa para expresar  $R_N$  es:  $R_N(X_N, D_N, D_{N-1}, \dots, D_1) =$

$$= g[f_N(X_N, D_N), f_{N-1}(X_{N-1}, D_{N-1}), \dots, f_1(X_1, D_1)]$$

El problema de optimización de un sistema de  $N$  etapas en función de su estado inicial, es maximizar el retorno de las  $N$  etapas  $R_N$  sobre las variables  $D_1, \dots, D_N$ ; esto es, encontrar el retorno óptimo como una función del estado inicial  $X_N$  denotando a  $f_N(X_N)$  como el retorno máximo de las  $N$  etapas y  $D_n^* = D_n^*(X_n)$ ,  $X_n^* = t_n^*(X_n)$  como los estados y las decisiones óptimas; se tienen dos alternativas para expresar  $f_N(X_N)$ :

$$I.- f_N(x_N) = g[f_N(x_N, D_N^*), f_{N-1}(x_{N-1}, D_{N-1}^*), \dots, f_1(x_1, D_1)]$$

$$= \max_{D_N, \dots, D_1} g[f_N(x_N, D_N), f_{N-1}(x_{N-1}, D_{N-1}), \dots, f_1(x_1, D_1)]$$

$$\text{Sujeta a: } x_{m-1} = z_m(x_m, D_m), \quad m = 1, 2, \dots, N \quad \delta$$

$$II.- f_N(x_N) = g[f_N(x_N, D_N^*), f_{N-1}(x_{N-1}, D_{N-1}^*), \dots, f_1(x_1, D_1^*)]$$

$$= \max_{D_N, \dots, D_1} g[f_N(x_N, D_N), f_{N-1}(x_{N-1}, D_{N-1}), \dots, f_1(x_1, D_1)]$$

Aparentemente, la segunda formulación es preferible, ya que sólo contiene  $N$  variables de decisión ( $D_N, \dots, D_1$ ) y una variable de estado ( $x_N$ ), en cambio la primera formulación contiene  $N$  variables de estado,  $N$  variables de decisión, y  $N$  restricciones; - ya que las técnicas de optimización disminuyen su eficiencia conforme aumenta el número de variables; parecería razonable eliminar los estados intermedios ( $x_{N-1}, \dots, x_1$ ) por medio de las ecuaciones de transformación sin embargo, la primera formulación a menudo puede transformarse en  $N$  problemas de optimización, cada uno conteniendo una variable de estado y una variable de decisión. De hecho, los estados intermedios - son a menudo introducidos artificialmente. En esos casos un problema dado en la formulación II se encuentra formando parte de un problema más grande (Formulación I), que sorprendentemente es más fácil de resolver.

Descomposición para Retornos Aditivos.- Nuestro Objetivo es descomponer el problema:

$$f_N(x_N) = \max_{D_N, \dots, D_1} g[f_N(x_N, D_N), f_{N-1}(x_{N-1}, D_{N-1}), \dots, f_1(x_1, D_1)]$$

$$\text{Sujeto a: } x_{m-1} = z_m(x_m, D_m), \quad m = 1, \dots, N$$

En  $N$  problemas equivalentes, cada uno conteniendo únicamente una variable de estado y una variable de decisión; cada uno de estos subproblemas será aproximadamente equivalente al problema de optimización de una sola etapa.

En lugar de resolver un problema en que todas las decisiones son interdependientes, encontraremos las decisiones óptimas una por una.

Para efectuar esta descomposición, debe hacerse una suposición al tanto restrictiva sobre la forma de la función  $g$ . Más que establecer esta condición, haremos la descomposición para una forma particular de  $g$  que satisfaga la condición. Esta derivación proporcionará el conocimiento por medio del cual se podrá deducir una condición suficiente sobre la forma de  $g$  para la descomposición. Sea:

$$g[f_N(x_N, D_N), f_{N-1}(x_{N-1}, D_{N-1}), \dots, f_1(x_1, D_1)]$$

Así:

$$= [f_N(x_N, D_N) + f_{N-1}(x_{N-1}, D_{N-1}) + \dots + f_1(x_1, D_1)]$$

$$f_N(x_N) = \max_{D_N, \dots, D_1} [f_N(x_N, D_N) + f_{N-1}(x_{N-1}, D_{N-1}) + \dots + f_1(x_1, D_1)]$$

Sujeta a:  $x_{n-1} = z_n(x_n, D_n), \quad n = 1, \dots, N$

ya que:

- 1.- El retorno de la  $N$ -ésima etapa no depende de  $D_1, \dots, D_{N-1}$ ,
- 2.- Para funciones reales arbitrarias;  $h_1(u_1)$  y  $h_2(u_1, u_2)^{N-1}$ ,

$$\max_{u_1, u_2} [h_1(u_1) + h_2(u_1, u_2)] = \max_{u_1} [h_1(u_1) + \max_{u_2} (u_1, u_2)]$$

$$f_N(x_N) = \max_{D_N} [f_N(x_N, D_N) + \max_{D_{N-1}, \dots, D_1} (f_{N-1}(x_{N-1}, D_{N-1}) + \dots + f_1(x_1, D_1))]$$

sujeta a:  $x_{n-1} = z_n(x_n, D_n), \quad n = 1, 2, \dots, N$

de la definición de:  $f_N(x_N)$  se sigue que:

$$f_{N-1}(x_{N-1}) = \max_{D_{N-1}, \dots, D_1} [f_{N-1}(x_{N-1}, D_{N-1}) + \dots + f_1(x_1, D_1)]$$

así:

$$f_N(x_N) = \max_{D_N} [f_N(x_N, D_N) + f_{N-1}(x_{N-1})]$$

sujeta a:

$$x_{N-1} = z_N(x_N, D_N)$$

$$\delta: f_N(x_N) = \max_{D_N} [f_N(x_N, D_N) + f_{N-1}(z_N(x_N, D_N))]$$

definiendo:  $Q_N(x_N, D_N) = [f_N(x_N, D_N) + f_{N-1}(t_N(x_N, D_N))]$

La determinación de  $f_N^*(x_N)$  y  $D_N^* = D_N^*(x_N)$ , dado  $f_{N-1}(x_{N-1})$  es simplemente el problema de optimización de una etapa, en función de su estado inicial con variable de estado  $x_N$ , variable de decisión  $D_N$  y retorno  $Q_N$ , esto es:  $f_N^*(x_N) = \max_{D_N} Q_N(x_N, D_N)$  de esta forma se ha simplificado el problema original de  $N$  etapas en dos problemas de optimización más pequeños:

- 1.-  $f(x) = \max_{D_{n-1}, \dots, D_1} [f_{n-1}(x_{n-1}, D_{n-1}) + \dots + \tau(x_1, D_1)]$  sujeto a:  $x_i = t(x_i, D_i), i = 1, \dots, N-1$
  - 2.-  $f_N(x_N) = \max_{D_N} Q_N(x_N, D_N) = \max_{D_N} [f_N(x_N, D_N) + f_{N-1}(t_N(x_N, D_N))]$
- es obvio que se puede continuar con la descomposición; de hecho, tratando a  $f_{N-1}(x_{N-1})$  y después a  $f(x_{N-2}), \dots, f_2(x_2)$  de la misma manera que se hizo con  $f_N(x_N)$ , se puede descomponer el problema original en  $N$  problemas de optimización de una sola etapa en función de su estado inicial.

$$1.- f_1(x_1) = \max_{D_1} Q_1(x_1, D_1) = \max_{D_1} \tau(x_1, D_1)$$

$$m.- f_m(x_m) = \max_{D_m} Q_m(x_m, D_m) = \max_{D_m} [f_m(x_m, D_m) + f_{m-1}(t_m(x_m, D_m))]$$

$$N.- f_N(x_N) = \max_{D_N} Q_N(x_N, D_N) = \max_{D_N} [f_N(x_N, D_N) + f_{N-1}(t_N(x_N, D_N))]$$

para expresar los  $N$  problemas más compactamente:

$$f_m(x_m) = \max_{D_m} Q_m(x_m, D_m) \quad m = 1, \dots, N$$

$$Q_m(x_m, D_m) = f_m(x_m, D_m), \quad m = 1$$

$$= f_m(x_m, D_m) + f_{m-1}(t_m(x_m, D_m)) \quad m = 2, \dots, N$$

Las ecuaciones anteriores representan las ecuaciones de recursión usuales de la programación dinámica; su solución recursiva partiendo — con la etapa  $m=1$  y continuando hasta  $m=N$ , producen el retorno óptimo de las  $N$  etapas  $f_N^*(x_N)$ , las decisiones óptimas  $D_N^* = D_N^*(x_N)$  y las funciones de decisión  $D_m = D_m(x_m), m=1, \dots, N-1$

Si se desea encontrar el estado de entrada óptimo simplemente resolveremos:  $f_N^*(x_N) = \max_{x_N} f_N^*(x_N)$ .

para resolver los estados y decisiones óptimas restantes, como una función de  $X_N$ , comenzaremos con:

$$x_{N-1}^* = t_N(x_N, D_N^*) = t_N(x_N, D_N(x_N)) = t_N(x_N)$$

$$D_{N-1}^* = D_{N-1}(x_{N-1}^*) = D_{N-1}(t_N(x_N, D_N^*)) = D_{N-1}(x_N)$$

y después se produce recursivamente de  $n = N-1, \dots, 1$  usando

$$\text{las relaciones } x_{n-1}^* = t_n(x_n, D_n^*) = t_n(x_n) \text{ y } D_{n-1}^* = D_{n-1}(x_{n-1}^*) = D_{n-1}(x_n).$$

Las ecuaciones de recursión que se acaban de desarrollar, corresponden al caso particular de la optimización de un sistema multietapas seriado, en función de su estado inicial (estado de entrada a la etapa  $N$ ); sin embargo, en ocasiones se desea optimizar el sistema en función de los estados de entrada a la etapa  $N$  y de salida de la etapa  $1$ , problema conocido como problema del estado inicial-final. Conceptualmente no hay ninguna diferencia entre el punto de vista de la programación dinámica de este problema y el problema del estado inicial; siendo la única diferencia que no se elimina  $X_0$  en la etapa uno — del procedimiento de optimización recursiva. Procediendo el análisis recursivo de la siguiente manera:

En la etapa uno se resuelve el siguiente problema:

$$f_1(x_1, x_0) = \max D_1(x_1, D_1, x_0)$$

$$\text{sueto a: } x_0 = t_1(x_1, D_1)$$

Si la segunda expresión se usa para expresar a  $D_1$ , como una función de  $x_1$  y  $x_0$ , esto es:  $D_1 = \hat{t}_1(x_0, x_1)$

Entonces no habrá optimización de la etapa uno, ya que:

$$f_1(x_1, x_0) = f_1(x_1, \hat{t}_1(x_0, x_1), x_0) = f_1(x_1, x_0)$$

Después de determinar  $f_1(x_1, x_0)$ , el resto del análisis recursivo procede exactamente como antes, con la excepción de que en cada etapa del análisis  $x_0$  es considerada una variable de estado adicional. Las ecuaciones de recursión son:

$$f_n(x_n, x_0) = \max_{D_n} [f_n(x_n, D_n) + f_{n-1}(t_n(x_n, D_n), x_0)], \quad n = 2, \dots, N$$

Finalmente, en ocasiones también se presenta el problema de encontrar el retorno óptimo de un sistema, en función del estado de salida

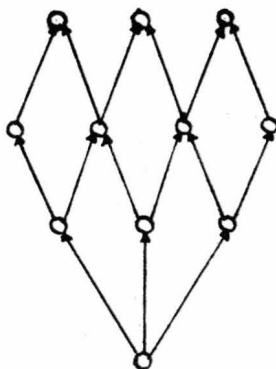
de la etapa uno ( $X_0$ ), situación conocida como problema del estado final; una manera obvia de proceder para resolver este problema es resolver el problema de los estados inicial-final y después optimizar con respecto a  $X_N$ . Esto es :

$$f_N(X_0) = \max_{X_N} f_N(X_N, X_0)$$

De esta manera se concluye la presentación formal, quizás demasiado concisa de la Programación Dinámica, no siendo descritos términos importantes tales como las condiciones de separabilidad y monotonicidad que debe satisfacer una función para poder ser descompuesta; diferencias entre recursión backward (hacia atrás) y recursión forward (hacia adelante); extensión de la Programación Dinámica a sistemas no-seriados; por encontrarse ampliamente tratados en la literatura tal como en : INTRODUCTION TO DYNAMIC PROGRAMMING por George L. Nemhauser de John Wiley and Sons, Inc.

Habiendo desarrollado las ecuaciones de recursión de la Programación Dinámica, usando un procedimiento deductivo y formal; a continuación procederemos a tratar el principio simple e intuitivo en el cual estas ecuaciones están basadas.

2.3. PRESENTACION INFORMAL E INTUITIVA DE LA PROGRAMACION DINAMICA.- Un sistema de decisión multietapas, en que cada variable de estado y cada variable de decisión puede tomar un número finito de valores, puede representarse gráficamente por un " Arbol de Decisiones " .



-----  
ETAPA UNO

-----  
ETAPA DOS

-----  
ETAPA TRES  
-----



Los círculos llamados nodos, corresponden a los estados y las líneas entre los círculos llamados arcos, corresponden a las decisiones. Partiendo del nodo en la base del árbol, que denota el estado inicial del sistema en la etapa tres, hay tres decisiones posibles representadas por tres arcos que emanan del nodo. Asociados con cada arco hay un retorno y una salida. Las salidas están representadas por los tres nodos al final de esos tres arcos. Las etapas dos y uno se interpretan en igual forma. Las soluciones posibles, corresponden a trayectorias ( un conjunto de arcos unidos ) entre el nodo de la base y un nodo cualquiera en el tope del árbol. El retorno de una trayectoria es igual a la suma de los retornos de los arcos incluidos en la trayectoria. El objetivo es encontrar la trayectoria que produzca el retorno máximo.

Trabajando "hacia atrás" para encontrar la trayectoria óptima, comenzaremos con los cuatro nodos de entrada a la etapa uno; no sabiendo de momento, cual de esos nodos está incluido en una trayectoria óptima encontraremos un arco que maximice el retorno al tope del árbol. Consecuentemente, para cada nodo de entrada a la etapa uno, el retorno y la trayectoria óptima al tope del árbol son ya conocidas, esto corresponde a la determinación de las funciones  $f_i(x_i)$  y  $D_i(x_i)$ . En particular, el retorno óptimo de un nodo específico y el arco que produjo ese retorno, son elementos específicos de  $f_i(x_i)$  y  $D_i(x_i)$ . El conjunto de arcos y retornos óptimos, uno para cada nodo de entrada a la etapa uno, corresponden a las funciones  $f_i(x_i)$  y  $D_i(x_i)$  respectivamente.

Consideremos ahora, un sistema de dos etapas consistiendo de las etapas dos y uno, que tiene tres nodos de entrada; buscaremos trayectorias y retornos óptimos de cada uno de estos nodos al tope del árbol. Esto se puede hacer encontrando arcos que maximicen los retornos de los arcos, combinados con los retornos óptimos de los nodos de salida. Una vez que estos arcos son conocidos, el mismo principio puede ser aplicado a un sistema de tres etapas para determinar la trayectoria óptima de la base al tope del árbol. El concepto fundamental es que únicamente necesitamos considerar los retornos óptimos de los nodos de entrada, ó en otras palabras, no es necesario considerar retornos que no son ópti-

mos con respecto a los nodos de salida. Después de todo, si se desea obtener una solución óptima de un sistema, cualquier porción del sistema debe ser optimizada. Es éste el principio de optimidad establecido por Bellman y que se enunció al inicio del presente capítulo.

#### 2.4 TECNICAS DE OPTIMIZACION.-

Se concluirá el presente capítulo, con un breve resumen de los métodos de optimización Standard de una sola etapa. Una vez que la formulación de la Programación Dinámica ha sido efectuada, la optimización es más fácil, pero todavía queda por hacerse. La técnica de optimización más elemental es la enumeración total o exhaustiva; esto simplemente significa calcular  $R(D)$  para todas las  $D$ 's posibles y usar la definición de optimidad directamente para identificar el conjunto de soluciones óptimas.

La enumeración es posible únicamente cuando hay un número finito de soluciones; sin embargo, en muchos casos : problemas con un número infinito de soluciones pueden ser aproximados con problemas conteniendo únicamente un número finito. Pero aún así, en ocasiones resulta impráctico sino es que imposible, resolver este tipo de problemas por búsqueda exhaustiva; requiriéndose para esas situaciones métodos más eficientes.

Sino se va a buscar exhaustivamente el óptimo global, se deben establecer reglas para seleccionar un subconjunto de soluciones posibles para evaluar la función objetivo. Debe haber además, una condición suficiente de optimidad que no dependa de la búsqueda exhaustiva. Desafortunadamente, la única condición universal suficiente de optimidad es la definición de optimidad.

En lugar de una condición suficiente de optimidad, una condición que es suficiente en casos especiales y si necesaria en todos los casos, es bastante útil. Una solución posible  $D_0$  que satisfaga :

$$R(D_0) \geq R(D_0 + \Delta D)$$

Para todos los valores posibles de la pequeña cantidad  $\Delta$  es definida como un óptimo local; obviamente el conjunto de óptimo global está contenido en el conjunto de óptimo local. Por lo tanto, esta condición

es necesario para un óptimo global y consecuentemente para funciones - con un óptimo local único (Unimodal), la definición de un óptimo local - es una condición necesaria y suficiente para un óptimo global.

Los procedimientos de búsqueda secuencial, son procedimientos iterativos basados en la estrategia de usar los resultados de evaluaciones previas, para determinar nuevos puntos para evaluar la función objetivo

. Una clase importante de este tipo de métodos: son los métodos conocidos como métodos de gradiente, cuyo principio es moverse de una solución a otra siempre procediendo en la dirección de máximo aumento en la función objetivo. Supongamos que la función objetivo ha sido evaluada en alguna solución arbitraria posible que no es un óptimo local. Puede demostrarse que la dirección de  $D_0$  en que la función objetivo aumenta - más rápidamente, es la dirección dada por el gradiente de  $R(D)$  evaluada en  $D_0$ , de esta manera, una nueva solución  $D_1$  en la línea con la pendiente de  $R(D)$  evaluada en  $D_0$ , puede ser determinada de tal forma que  $R(D_1) > R(D_0)$ ; si  $D_1$  no es un óptimo, se evalúa el gradiente en  $D_1$ , se encuentra una nueva solución  $R(D_2) > R(D_1)$  y se continúa de esta manera. Este procedimiento convergirá finalmente a un máximo local.

Para el caso especial de funciones objetivo y restricciones lineales, esto es problemas lineales, el problema de optimización es:

$$\text{MAX } cD$$

$$\text{Sujeto a: } AD \leq B$$

$$D \geq 0$$

Para problemas lineales, hay un método de búsqueda del tipo de gradiente bastante poderoso: LA PROGRAMACION LINEAL; de la teoría de la programación lineal se sabe que la solución óptima corresponde a un punto extremo del conjunto convexo definido por restricciones lineales. -- Los algoritmos de la programación lineal proporcionan métodos para moverse de un punto extremo a otra adyacente, siempre aumentando el valor de la función objetivo. Cuando un punto extremo es alcanzado que tiene la propiedad que la función objetivo no se incrementa por el movimiento a un punto extremo adyacente, un óptimo global ha sido encontrado.

Los métodos de la programación lineal han sido extendidos al área-

más general de la programación matemática, para manejar ciertas funciones no lineales; también se han desarrollado algoritmos especiales para la optimización de funciones objetivo cuadráticas y algoritmos de programación entera.

Una alternativa a aumentar el valor de la función objetivo en cada paso, es reducir en cada paso la región de factibilidad de contener la solución óptima; por ejemplo : si en cada evaluación de la función objetivo pudiéramos reducir la porción de la región de factibilidad de contener la solución óptima por  $\frac{1}{q}$ ,  $q > 1$ ; después de  $N$  evaluaciones, la solución óptima estará contenida en una región de  $(\frac{1}{q})^N$  veces el tamaño de la región original.

El método de búsqueda de Fibonacci es un ejemplo interesante de esta estrategia y puede ser aplicado a cualquier función unimodal de una variable garantizando que no más de un número determinado de evaluaciones de la función objetivo necesitan ser hechas para encontrar la solución óptima. Es llamado búsqueda de Fibonacci porque el número de puntos examinados y la estrategia para colocarlos, está relacionada a la secuencia de Fibonacci.

Este método puede ser considerado como un método óptimo de búsqueda, ya que minimiza el número máximo de puntos que necesitan ser buscados para funciones unimodales de una variable.

Desafortunadamente no hay procedimientos óptimos de búsqueda para funciones arbitrarias de varias variables.

El método de las tangentes a los contornos usa una estrategia de alimentación y puede ser aplicado a funciones de varias variables, pero no da un límite superior óptimo respecto al número de puntos a ser buscados. Finalmente, la solución óptima también puede ser encontrada para funciones objetivo diferenciables dos veces y sin restricciones por medio de los procedimientos del cálculo diferencial, tema que está tratado extensivamente en la mayoría de los libros de cálculo diferencial e integral. Quedando sólo por mencionar que para el caso en que se tengan restricciones, se puede recurrir a los multiplicadores de Lagrange para transformar un problema restringido en uno equivalente sin restricciones y aplicar los procedimientos del cálculo diferencial.

CAPITULO 3

APLICACION DE LA PROGRAMACION DINAMICA A UN PROBLEMA ESPECIFICO.

3.1 PRESENTACION DEL PROBLEMA.- El objetivo central de esta tesis, es la aplicación de la Programación Dinámica, al cálculo óptimo de un sistema de evaporación triple efecto, corriente paralela, para la evaporación de soluciones acuosas de Hidróxido de Sodio. Los tres efectos de este sistema, serán evaporadores a circulación forzada, en donde el calor se suministrará por medio de cambiadores de calor externos, operados bajo cabezas suficientes, de tal manera que no ocurra ebullición en los mismos. Los cambiadores de calor de cada uno de los efectos, son cambiadores de un solo paso, tubos de Nikel 3/4" IPS Standard, en donde el calor que se cederá será únicamente calor latente, lo cual implica que todo el sobrecalentamiento se pierde en las líneas de vapor. Además, del análisis de las condiciones bajo las cuales este sistema operará, puede decirse que el cuerpo del evaporador propiamente dicho corresponde en última instancia, a solamente un tanque de Flasheo.

El sistema de evaporación que se ha seleccionado para una ilustración académica de la Programación Dinámica, efectuará la concentración de 250,000.00 Lbs./Hr. de una solución acuosa de Hidróxido de Sodio del 10 al 20%, siendo la temperatura de la solución diluída que se alimentará al sistema 110°F.

El último efecto de este sistema de evaporación trabajará a una presión tal, que la temperatura mínima de saturación del vapor generado en el mismo sea de 110°F; lo anterior, es con el objeto de tener una buena caída de temperatura en el condensador del sistema.

Finalmente, para los propósitos de calentamiento del sistema en el sistema antes mencionado, se dispondrá de un vapor vivo de hasta 60 PSI A.

La nomenclatura tradicional de las etapas de un sistema de evaporación múltiple efecto, es totalmente opuesta a la de la Programación Dinámica. En esta tesis se dará prioridad a la nomenclatura de esta última, de acuerdo con la cuál: el último efecto del sistema de evaporación-- corresponde a la etapa  $n=1$  (I) y el primer efecto corresponderá a la etapa  $n=N$  (III).

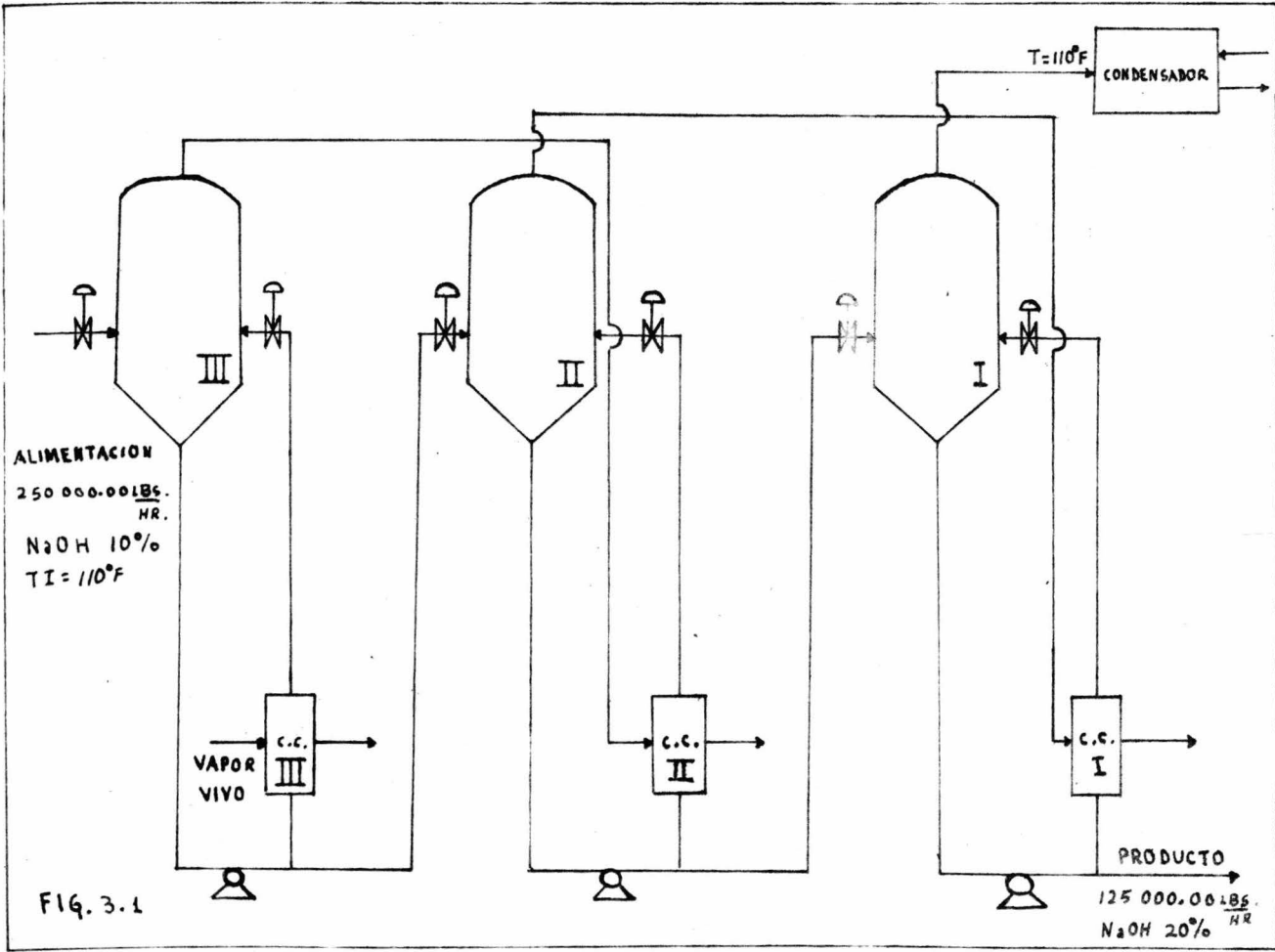


FIG. 3.1

Un diagrama esquemático del sistema de evaporación cuyo cálculo se pretende optimizar, se encuentra en la figura 3.1 .

3.2 MODELO MATEMATICO DEL SISTEMA DE EVAPORACION.- Una tendencia reciente en el desarrollo de modelos, lo es el uso del procedimiento modular. Este procedimiento consiste fundamentalmente en la elaboración uno por uno, de los modelos matemáticos de los módulos ó partes pequeñas de un sistema complejo, en lugar de considerar la elaboración del modelo matemático del sistema completo a un solo tiempo. Siendo por lo tanto, iguales las filosofías que emplean el procedimiento modular y la programación dinámica, ya sea que se trate de la elaboración de un modelo matemático ó la optimización de un sistema complejo.

Usando el procedimiento modular, podemos decir que el sistema bajo consideración está formado por tres módulos del tipo mostrado en la figura 3.2 .

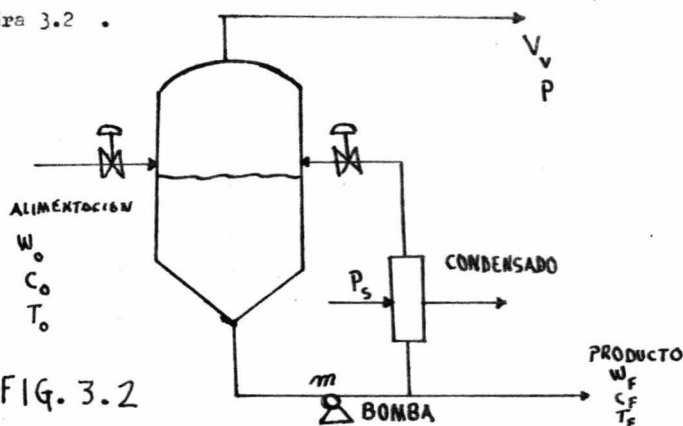


FIG. 3.2

En la práctica, el módulo propuesto puede subdividirse aún más. Se pueden considerar por separado: el cambiador de calor, la bomba, la válvula y el cuerpo del evaporador propiamente dicho; combinándose posteriormente los modelos individuales en una estructura computacional, para obtener una solución numérica. La subdivisión debe efectuarse hasta -- que cada subsistema sea entendido, siendo una cuestión de decisión personal el continuar con la subdivisión.

Un paso importante en el desarrollo de un modelo, es establecer -- qué cantidades están especificadas y cuáles necesitan ser evaluadas. --

Consideremos que el módulo mostrado anteriormente, es operado como un - simple efecto con las siguientes cantidades especificadas:

- $C_o$  Concentración de la solución de alimentación.
- $T_o$  Temperatura de la solución de alimentación.
- $W_o$  Flujo de la solución de alimentación.
- $P_e$  Presión en el efecto.
- $C_F$  Concentración de la solución producto.
- $P_s$  Presión del vapor de calentamiento.
- $T_s$  Temperatura del vapor de calentamiento.

Las siguientes cantidades deberán ser evaluadas:

- $T_F$  Temperatura del licor en el efecto.
- $W_F$  Flujo de la solución producto.
- $V_v$  Flujo del vapor generado en el efecto.
- $W_s$  Consumo de vapor de calentamiento.
- $Q$  Flujo de calor.
- $m$  Flujo a través de la bomba.
- $A$  Area para la transferencia de calor en el efecto.

La siguiente cantidad va a ser un parámetro:  
 $(T_s - T_e) = \text{cte.}$  Approach en el cambiador de calor.

Además se necesitan relaciones para determinar todas las propiedades físicas (densidad, conductividad térmica, capacidad calorífica, viscosidad y temperatura de ebullición etc. de soluciones acuosas de Hidróxido de Sodio) y el coeficiente de transferencia de calor.

Una vez hecho lo anterior, procederemos a desarrollar el modelo matemático del evaporador y finalmente fijaremos nuestra atención en la técnica para resolver las ecuaciones resultantes.

El flujo de calor está dado por cualquiera de las siguientes relaciones :

$$Q = U A \frac{(T_s - T_e) - (T_s - T_F)}{Lm \frac{(T_s - T_e)}{(T_s - T_F)}} \quad 3.1$$

$$Q = (m - W_F) C_p (T_e - T_F) \quad 3.2$$



$$Q = W_s \lambda_s \quad 3.3$$

Ya se dijo, que se necesitan relaciones para el cálculo de las propiedades físicas ( $C_p$ ,  $\lambda_s$ ) y el coeficiente de transferencia de calor.

Incluirémos estas relaciones en el conjunto de ecuaciones y las representaremos por:

$$C_p = f_1 \left( \frac{T_e + T_F}{2}, C_F \right) \quad 3.4$$

$C_p$  se evalúa a la temperatura media, usando una gráfica como la que se encuentra en la figura 11-3(C) del libro FORMULATION AND OPTIMIZATION OF MATHEMATICAL MODELS por Cecil L. Smith, Ralph W. Pike editado por INTERNATIONAL TEXTBOOK COMPANY.

$$\lambda_s = f_2 (P_s) \quad 3.5$$

La relación funcional dada por la ecuación 3.5 corresponde a las tablas de vapor:

$$U = f_3 (T_F, T_e, T_s, m, W_F, W_s, etc.) \quad 3.6$$

donde  $f_3$ , representa el procedimiento para calcular U, a partir de las cantidades encerradas dentro del paréntesis.

De la observación de las seis ecuaciones desarrolladas anteriormente, puede decirse que las mismas corresponden al modelo matemático del cambiador de calor.

$$\text{Balance total de masa: } W_o = W_F + V_v \quad 3.7$$

$$\text{Balance de masa para un componente: } W_o C_o = W_F C_F \quad 3.8$$

$$\text{Balance de entalpía: } W_o H_o + Q = W_F H_F + V_v H_v \quad 3.9$$

Estas tres últimas ecuaciones, puede decirse que corresponden al modelo matemático del cuerpo del evaporador propiamente dicho.

Las nueve ecuaciones desarrolladas anteriormente, corresponden al modelo matemático de uno cualquiera de los módulos del sistema de evaporación triple efecto, corriente paralela, propuesto para optimizarse. -

La solución de estas ecuaciones para la situación bajo consideración — se hace de la manera siguientes:

- 1.- Se calcula  $W_F$  de la ecuación 3.8

$$W_F = W_0 C_0 / C_F$$

- 2.- Se calcula  $V_V$  de la ecuación 3.7

$$V_V = W_0 - W_F$$

- 3.- Se determina  $T_F$  usando una gráfica como la que se encuentra en la — figura 5.22 del libro INTRODUCTION TO CHEMICAL ENGINEERING por Walter — L. Badger and Julius T Banchero), editado por Mc Graw Hill.

- 4.- Se evalúan todas las entalpías que aparecen en la ecuación 3.9

- 5.- La ecuación 3.9 se resuelve para  $Q$ .

$$Q = W_F H_F + V_V H_V - W_0 H_0$$

- 6.- Por medio de la relación 3.5 se determina el calor latente de eva-  
poración del vapor de calentamiento.

- 7.- La ecuación 3.3 se resuelve para  $W_S$

$$W_S = Q / \lambda_S$$

- 8.- La ecuación 3.2 se resuelve para  $m$  :

$$m = W_F + Q / (C_p (T_E - T_F))$$

Siendo  $C_p$  evaluado por medio de la relación funcional  $f_1$ , dada por—  
la ecuación 3.4 .

- 9.- El coeficiente de transferencia de calor se determina por medio de  
la relación funcional  $f_3$  dada por la ecuación 3.6 .

- 10.- La ecuación 3.1 se resuelve para  $A$  :

$$\text{Haciendo: } \frac{(T_S - T_E) - (T_S - T_F)}{L \eta \frac{(T_S - T_E)}{(T_S - T_F)}} = \Delta T M L ; \quad A = \frac{Q}{U A \Delta T M L}$$

Nota.- El consumo de potencia para cada uno de los efectos del sistema—  
de evaporación, se determina mediante la aplicación del teorema de BER  
NOULLI, para las condiciones de succión y de descarga de la bomba de re  
circulación.

Una vez que se ha formulado el modelo matemático de nuestro sistema de evaporación, a continuación procederemos a describir el ambiente económico en el cual este sistema va a funcionar y también estableceremos un criterio que usaremos durante el proceso de "diseño" y que nos conducirá al sistema económicamente óptimo.

3.3 SELECCION DE LA FUNCION OBJETIVO.- La función objetivo que se ha seleccionado para la optimización del presente sistema, es el COSTO ANUAL DE OPERACION. El costo de operación, involucra los gastos necesarios para mantener el proceso operando día con día y puede ser dividido en términos que son proporcionales a la inversión necesaria para la adquisición de los equipos, términos que son proporcionales a la cantidad de producto generado por el sistema, y términos proporcionales a la mano de obra.

En forma simplificada, los términos proporcionales a la inversión-fija son : Depreciación, Mantenimiento(materiales y mano de obra; ésta es la parte del mantenimiento que es independiente de la cantidad producida).

La mayor porción del costo de operación, es aproximadamente proporcional, a la cantidad de producto generado por el sistema, e involucra los costos incurridos por el consumo de materias primas, potencia eléctrica, vapor, agua de enfriamiento, catalizadores, etc. . Generalmente las cantidades de estos términos requeridas por el sistema, se determinan por medio de balances de masa y energía. Esta categoría también comprende los gastos de mantenimiento( incurridos por la operación del proceso).

Los términos proporcionales a la mano de obra son : mano de obra directa, supervisión etc..

Algunos de los términos mencionados, podrían incluirse en una u otra de estas tres categorías con igual precisión; el hecho realmente importante, es que deben incluirse en alguna de ellas.

Para los propósitos de la optimización del sistema bajo consideración, la función objetivo representada por el costo anual de operación-incluirá : costos por consumo de vapor, Depreciación, costos por mano -

de obra directa y de mantenimiento, y costos por consumo de potencia.

La función objetivo para los efectos I y II, comprenderá la depreciación del cuerpo del evaporador, del cambiador de calor y de la bomba; costos por mano de obra directa y de mantenimiento y costos por consumo de potencia. La función objetivo para el efecto III, comprenderá: - la depreciación del cuerpo del evaporador, del cambiador de calor y de la bomba; costos por mano de obra directa y de mantenimiento; costos por consumo de potencia y costos por consumo de vapor.

Será en la formulación de las ecuaciones de recursión, en donde se hará la formulación en términos matemáticos, de la función objetivo para cada una de las etapas (efectos) del sistema de evaporación.

El sistema de evaporación trabajará 8000 hrs./año en un ambiente económico en donde pueden considerarse los siguientes parámetros económicos:

Costo KW - HR = \$ 0.20

Costo del Vapor Vivo de Calentamiento = \$ 8.125/1000 LBS.

Costo Mano de Obra Directa = \$ 30.00 / HR. HOMBRE

Costo Mano de Obra de Mantenimiento = \$ 30.00 / HR. HOMBRE

Costo de un Módulo (cuerpo del evaporador, cambiador de calor y bomba)  
= \$ 1500.00 / ft<sup>2</sup>.

Depreciación = Depreciación Línea Recta a 10 Años.

Para el sistema que se pretende optimizar, resulta razonable pensar que con dos operarios y trabajándose en tres turnos, se podrá operar y dar el mantenimiento adecuado al sistema. Por lo tanto, en la función-objetivo para cada uno de los efectos del sistema, sólo se hará el cargo proporcional correspondiente para los conceptos de mano de obra de operación y mantenimiento.

3-4 FORMULACION DE LAS ECUACIONES DE RECURSION.- La formulación de la programación dinámica para el problema bajo consideración, puede efectuarse en forma rigurosa o en forma simplificada. (Itahara, Seiji, Stiel, L. I., IND. ENG. CHEM. PROCESS DESIGN DEVELOP. 5, 309 (1966); Itahara, Seiji, Stiel, L. I., IND. ENG. CHEM. PROCESS DESIGN DEVELOP. Vol. 7, No. 1, January 1968).

La formulación exacta de la programación dinámica usa dos variables de estado y una variable de decisión, requiriendo ecuaciones de transformación bastante complejas y grandes cantidades de memoria de computadora. La formulación simplificada por otra parte, reduce apreciablemente los requerimientos de memoria y produce esencialmente los mismos resultados que la formulación exacta.

En el presente trabajo efectuaremos la formulación simplificada de la programación dinámica, estableciendo la hipótesis de que los vapores generados en cada uno de los efectos varían solamente en función de los cambios en las caídas de temperatura; se hace un estimado inicial del vapor generado en cada uno de los efectos por medio de la fórmula:

$$V_{n,i} = \frac{L_{n+1} - L_n}{N} \quad \text{donde: } i = 1$$

Usando estos flujos de vapor, que se suponen ahora independientes de la temperatura, se determina la distribución de temperaturas óptima por Programación Dinámica. En el paso siguiente, un nuevo conjunto de  $V_{n,i}^1$  ( $V_{n,i}^1 S$ ) es calculado usando la distribución de temperatura óptima en las ecuaciones de los balances de materia y energía. Este procedimiento es terminado cuando  $V_{n,i}^1 S$  y  $V_{n,i+1}^1 S$  satisfacen determinado criterio de convergencia en dos iteraciones sucesivas.

En esta formulación simplificada de la programación dinámica se considerará lo siguiente:

Variable de Estado =  $\sum_n =$  Temperatura del vapor entrando a la etapa n.

Variable de Decisión =  $\Delta T_n =$  Caída de temperatura en el efecto n del evaporador múltiple efecto.

Ecuación de Transformación :  $\sum_{n-1} = \sum_n - \Delta T_n - BPR_n$

Función Objetivo n=1

$$F_n(\sum_n) = \text{MIN}_{(\Delta T_n)} \left[ 1500.00 * 0.10 * A_n + \text{CONPOA}_n * 0.20 + 160,000.00 \right]$$

Función Objetivo n=2

$$F_n(\sum_n) = \text{MIN}_{(\Delta T_n)} \left[ 1500.00 * 0.10 * A_n + \text{CONPOA}_n * 0.20 + 160,000.00 + F_{n-1}(\sum_{n-1}) \right]$$

Función Objetivo n = N = 3

$$F_N(\sum_N) = \text{MIN}_{(\Delta T_n)} \left[ 1500.00 * 0.10 * A_n + \text{CONPOA}_n * 0.20 + W_{s_n} * 8000.00 * 0.008125 + 160,000.00 + F_{N-1}(\sum_{N-1}) \right]$$

Donde :  $1500.00 * 0.10 * A_n$  = Cargo por Depreciación del efecto n del sistema de evaporación.

$\text{CONPOA}_n * 0.20$  = Costo por Consumo de Potencia del Efecto n del sistema.

$W_{s_n} * 8000.00 * 0.008125$  = Costo por Consumo de Vapor Vivo de Calentamiento del efecto n = N

160 000.00 = Cargos Fijos Anuales por Mano de Obra de Operación y Mantenimiento.

3.5 TECNICA DE OPTIMIZACION SELECCIONADA.- La técnica de optimización de una sola etapa que se ha seleccionado para la optimización del sistema bajo estudio, es el Método Complejo de Box. Esta técnica fué desarrollada a partir del método Simplex de Spendley y puede decirse que es la versión "limitada" del mismo.

El método seleccionado, es una técnica de búsqueda secuencial, bas-

tánte efectivo para la optimización de funciones objetivo multivariables, no lineales; sujetas a restricciones no lineales de desigualdad :

$$\text{MAXIMIZAR } F(X_1, X_2, \dots, X_N)$$

$$\text{Sujeto a : } G_K \leq X_K \leq H_K, \quad K = 1, 2, \dots, M$$

Las variables Implícitas  $X_{N+1}, \dots, X_M$  son funciones dependientes de las variables Explícitas independientes  $X_1, X_2, \dots, X_N$ . Las restricciones Superior e Inferior  $H_K$  y  $G_K$  o son constantes o son funciones de las variables independientes.

El método Complejo de Box no requiere de derivada y maneja las restricciones, mediante el uso de una figura flexible de más de  $N+1$  vértices, que puede expandirse ó contraerse en todas las direcciones. Los vértices se eliminan y se generan como en el método Simplex, sólo que no se toma ninguna medida para mantener una figura regular en que cada vértice equidiste de todos los otros puntos.

El método Complejo de Box procede de la siguiente manera:

1.- Se genera un "complejo" original de  $K=N+1$  puntos consistiendo de un punto de partida que satisface todas las restricciones y  $K-1$  puntos adicionales generados a partir de números aleatorios y de las restricciones para cada una de las variables independientes:

$$X_{i,j} = G_i + f_{i,j} (H_i - G_i),$$

$$i = 1, 2, \dots, N$$

$$j = 1, 2, \dots, K-1$$

donde:  $f_{i,j}$  son números aleatorios entre cero y uno.

2.- Los puntos seleccionados deben satisfacer las restricciones explícitas e implícitas. Cada vez que un punto viola un límite explícito, este punto es desplazado una pequeña distancia (DELTA) dentro del límite violado y si un punto viola un límite implícito, entonces se mueve la mitad de su distancia al centroide de los puntos restantes:

$$X_{i,j} (\text{NUEVO}) = (X_{i,j} (\text{ANTERIOR}) + \bar{X}_{i,c}) / 2$$

$$i = 1, 2, \dots, N$$

Donde las coordenadas del centroide de los puntos restantes  $X_{i,c}$  están definidas por :

$$X_{i,c} = \left[ \frac{1}{K-1} \right] \left[ \sum_{j=1}^K X_{i,j} - X_{i,j} (\text{ANTERIOR}) \right], \quad i = 1, 2, \dots, N$$

Este proceso es repetido hasta que todas las restricciones implícitas sean satisfechas.

3.- La función objetivo es evaluada en cada punto. El punto que tiene el valor más bajo de la función es reemplazado por un punto localizado a una distancia de  $\lambda$  veces la distancia del punto al centroide de los puntos restantes; el nuevo punto debe ser colineal con el punto eliminado y el centroide.

4.- Si un punto continúa dando el valor más bajo de la función en varios intentos sucesivos, entonces se mueve la mitad de su distancia al centroide de los puntos restantes.

5.- El nuevo punto se checa contra las restricciones y se ajusta como antes, en el caso de que haya violado a alguna de las mismas.

6.- El método continuará a través de la eliminación y regeneración repetida, hasta que el complejo sea reducido esencialmente al centroide.

7.- El método Complejo de Box concluirá cuando se obtengan "unas cinco" evaluaciones consecutivas iguales de la función; donde el término "igual" dependerá de la precisión deseada.

Para concluir, mencionaremos que el Método Complejo de Box tiende a encontrar el máximo global. Lo anterior se infiere del hecho de que partiendo con un conjunto inicial de puntos distribuidos aleatoriamente en la región permisible, todos ellos convergen en la misma solución.



## CAPITULO 4

### PRESENTACION DE RESULTADOS Y CONCLUSIONES

4.1 PRESENTACION DE RESULTADOS.- Aunque a lo largo del desarrollo de esta tésis , se han empleado indistintamente los términos de "cálculo" y "diseño" óptimo del sistema; en realidad, como el título de la tésis lo indica, el presente trabajo efectúa solo el cálculo óptimo de los requerimientos de área para la transferencia de calor y de vapor vivo de calentamiento, así como el consumo de potencia para el sistema de evaporación -propuesto; no considerándose por lo tanto, los detalles del diseño mecánico del sistema.

En el programa de computadora implementado para el cálculo óptimo -del sistema de evaporación, se hicieron las consideraciones necesarias -para la impresión de los requerimientos de área para la transferencia de calor, así como de vapor vivo de calentamiento y el consumo de potencia-para cada una de las situaciones que se presentan , dada la forma como -procede el algoritmo de la programación dinámica y la naturaleza altamente iterativa de la técnica de optimización de una sola etapa (Método Complejo de Box); pero debido a que la impresión de dichos requerimientos -para cada una de las situaciones antes mencionadas, requiere de un gran-espacio, en la presentación de resultados que se hará a continuación solo se incluirá a aquéllos que corresponden a la solución óptima. Los resultados generados por el programa se muestran a continuación:

#### PROCEDIMIENTO DE LA PROGRAMACION DINAMICA

OPTIMIZACION DE UN SISTEMA DE 3 ETAPAS EN FUNCION DE SU ESTADO FINAL.

TABLA DE VALORES PARA LA ETAPA 1

VARIABLE DE ESTADO	VALOR DE LA FUNCION	VARIABLE DE DECISION
150.000	4.083472E+05	30.000
151.000	4.076148E+05	30.000
152.000	4.068875E+05	30.000
153.000	4.061516E+05	30.000

150.000	4.083472E+05	30.000
151.000	4.076148E+05	30.000
152.000	4.068875E+05	30.000
153.000	4.061516E+05	30.000
154.000	4.054215E+05	30.000
155.000	4.047099E+05	30.000
156.000	4.036011E+05	30.000
157.000	3.985200E+05	30.999
158.000	3.936532E+05	31.999
159.000	3.889861E+05	32.999
160.000	3.845052E+05	33.999
161.000	3.801982E+05	34.999
162.000	3.760541E+05	35.999
163.000	3.720626E+05	36.999
164.000	3.682145E+05	37.999
165.000	3.645012E+05	38.999
166.000	3.609151E+05	39.999
167.000	3.574489E+05	40.999
168.000	3.540959E+05	41.999
169.999	3.508502E+05	42.999
170.000	3.477060E+05	43.999

TABLA DE VALORES PARA LA ETAPA 2

VARIABLE DE ESTADO	VALOR DE LA FUNCION	VARIABLE DE DECISION
175.000	8.498335E+05	34.999
177.000	8.473998E+05	34.999
179.000	8.449712E+05	34.999
181.000	8.425762E+05	34.999
183.000	8.402279E+05	34.999
185.000	8.378809E+05	34.999
187.000	8.355013E+05	34.999
189.000	8.333536E+05	34.999
191.000	8.311126E+05	34.999

193.000	8.281640E+05	34.999
195.000	8.243111E+05	34.999
197.000	8.205074E+05	34.999
199.000	8.120102E+05	34.999
201.000	8.003346E+05	34.999
203.000	7.895893E+05	34.999
205.000	4.433464E+05	27.303
207.000	4.383647E+05	27.868
209.000	4.365070E+05	27.868
211.000	4.346696E+05	27.868
213.000	4.004550E+05	34.999
215.000	3.988571E+05	34.999

TABLA DE VALORES PARA LA ETAPA 3

VARIABLE DE ESTADO    VALOR DE LA FUNCION    VARIABLE DE DECISION

225.000	6.002149E+06	34.999
226.350	6.028813E+06	34.999
227.700	6.053379E+06	34.999
229.050	6.078154E+06	34.999
230.400	6.105220E+06	34.999
231.750	6.130099E+06	34.999
233.100	6.157274E+06	34.999
234.450	6.181555E+06	34.999
235.800	6.204709E+06	34.999
237.150	6.231734E+06	34.999
238.500	6.060746E+06	27.303
239.850	6.079787E+06	27.434
241.200	6.088814E+06	27.868
242.550	6.112187E+06	27.368
243.900	6.150354E+06	27.113
245.250	6.153375E+06	26.364
246.600	6.026156E+06	34.999

247.950	6.044857E+06	30.000
249.300	6.068583E+06	34.999
250.650	5.816002E+06	27.868
252.000	5.839738E+06	27.869

RETORNO OPTIMO DEL SISTEMA = 6.23178382E+06

RETORNOS MINIMOS DE LAS ETAPAS

ETAPA 1 RETORNO = 4.20310278E+05

ETAPA 2 RETORNO = 4.18159519E+05

ETAPA 3 RETORNO = 5.40747267E+06

DECISIONES OPTIMAS

ETAPA 1 X(1,1) = 3.00000000 E+01

ETAPA 2 X(2,1) = 3.49999000 E+01

ETAPA 3 X(3,1) = 3.49999000 E+01

ESTADO DE SALIDA DEL SISTEMA = 1.13000000 E+02

ESTADO DE ENTRADA A LA ETAPA 1 = 1.53000000 E+02

ESTADO DE ENTRADA A LA ETAPA 2 = 1.95000000 E+02

ESTADO DE ENTRADA AL SISTEMA = 2.37150000 E+02

Donde : como ya se dijo anteriormente, el término "retorno", ya sea que se trate del sistema completo o de cada una de las etapas, corresponde al costo anual de operación y tiene las unidades de \$/año ; las "decisiones óptimas" corresponden a las caídas de temperatura en cada uno de los efectos, tienen las unidades de °F. "Los estados de entrada y salida" corresponden a la temperatura del vapor que llega o que sale de cada uno de los efectos ó del sistema completo, tienen las unidades de °F.

Para los resultados presentados anteriormente y que corresponden a-

la aplicación de la Programación Dinámica al sistema de evaporación, dicho sistema quedará definido de la manera siguiente:

El efecto I, operará a una presión de trabajo en el cuerpo del evaporador de 1.3902 Psia.; su cambiador de calor tendrá 457 tubos de 13.53 Ft. de longitud y en él se elevará la temperatura de la solución, desde su temperatura de ebullición en el efecto ( $123^{\circ}\text{F}$ ), hasta que se alcance un acercamiento (approach) de  $20^{\circ}\text{F}$  con la temperatura del vapor de calentamiento en dicho efecto ( $153^{\circ}\text{F}$ ).

En el efecto II, la presión de trabajo en el cuerpo del evaporador será de 4.0045 Psia. ; su cambiador de calor tendrá 279 tubos de 21.3120 Ft. de longitud y en él se elevará la temperatura de la solución, desde la temperatura de ebullición de la misma en el efecto ( $160^{\circ}\text{F}$ ), hasta que se alcance un acercamiento de  $20^{\circ}\text{F}$  con la temperatura del vapor empleado para el calentamiento en dicho efecto ( $195^{\circ}\text{F}$ ).

El efecto III, operará a una presión de trabajo en el cuerpo del evaporador de 10.386 Psia.; su cambiador de calor tendrá 617 tubos de 17.071 Ft. de longitud y en él se elevará la temperatura de la solución, desde la temperatura de ebullición de la misma en el efecto ( $202.15^{\circ}\text{F}$ ), hasta que se alcance un acercamiento de  $20^{\circ}\text{F}$  con la temperatura del vapor vivo de calentamiento alimentado al sistema por el efecto III ( $237.15^{\circ}\text{F}$ ).

El vapor vivo de calentamiento de 23.720 Psia. , requerido para la operación de una hora del sistema es 73,919.5 Lbs./hr. . La cantidad de agua evaporada en los efectos I, II y III en una hora de operación del sistema será: 48,611.11 , 44637.76 y 31,751.13 Lbs. respectivamente.

Las concentraciones de la solución que hierve en los efectos I, II y III son respectivamente :  $c_1 = 0.200$  (por la definición del problema original),  $c_2 = 0.144$  y  $c_3 = 0.1145$  .

Para concluir con esta presentación de resultados, los flujos intermedios de la solución en el sistema de evaporación son:  $L_3 = 218,248.87$  Lbs./

Hr. y  $L_2 = 173,611.11$  Lbs./Hr.; Siendo los flujos de entrada y de salida del sistema, los mismos que se mencionaron en el capítulo 3.

4.2 CONCLUSIONES.- Los resultados presentados anteriormente, son una función de la naturaleza del sistema de evaporación, del ambiente económico en el cual el sistema funcionará y del método de optimización empleado. Por lo tanto, la tendencia general que se seguirá en la discusión de los resultados, será analizar el efecto en los mismos, de la naturaleza del sistema y del método de optimización; no se considerará la influencia del ambiente económico, porque el mismo ya se encuentra definido y establecido y normalmente está totalmente fuera de nuestro control.

Desde el inicio de estas discusiones, debe dejarse establecida la interrelación que existe entre la precisión de los resultados del método de optimización y el grado de conocimiento que se tiene de la naturaleza del sistema de evaporación; manifestándose dicha interrelación, sobre todo en la etapa de elaboración del modelo matemático del sistema.

La optimización que ha sido efectuada sobre el sistema de evaporación, es puramente matemática; ya que en la misma no se han tomado en consideración conceptos muy importantes, tales como : Las ventajas del patrón de flujo a contracorriente sobre el patrón a flujo paralelo, para el proceso de evaporación de soluciones acuosas de Hidróxido de Sodio.

Las soluciones acuosas de Hidróxido de Sodio, son bastante viscosas y en un caso como el presente, de patrón de flujo a corriente paralela : la solución más concentrada estará en el efecto de menor temperatura, siendo su viscosidad bastante alta, resultando por lo tanto, coeficientes de transferencia de calor bastante bajos para dicho efecto, afectándose por consiguiente el óptimo del sistema de evaporación completo.

El patrón de flujo a corriente paralela, dada la temperatura bastante baja de la solución de alimentación al sistema, afectará también desde otro punto de vista el óptimo del sistema de evaporación: La Economía (cantidad de agua evaporada en el evaporador múltiple efecto por libra de vapor vivo de calentamiento suministrado) ; siendo como sabemos, la

economía del vapor vivo de calentamiento, una de las razones fundamentales para la operación de evaporadores en múltiple efecto.

Idealmente, en un sistema de evaporación triple efecto, independientemente del patrón de flujo seleccionado, una libra de vapor vivo de calentamiento, producirá la evaporación de 3 libras de agua de la solución; lo anterior no sucede en la práctica, debido a las elevaciones en el punto de ebullición y a la variación del calor latente de evaporación con la temperatura.

Además, en el sistema bajo estudio, dado el patrón de flujo seleccionado y la temperatura bastante baja de la solución de alimentación, en el efecto III, una gran parte del calor cedido por el vapor vivo de calentamiento, será empleado para calentar la solución de alimentación a su temperatura de ebullición y este calor no generará ningún vapor en los efectos posteriores resultando por lo tanto, una economía bastante baja que afectará considerablemente el óptimo del sistema.

Uno de los problemas más graves que se presenta al trabajar con la operación unitaria de evaporación, es el cálculo de los coeficientes para la transferencia de calor; reflejándose la falta de precisión en el cálculo de los mismos, en la precisión del modelo matemático del sistema.

En este trabajo, en el modelo matemático de cada una de las etapas del sistema, se ha separado la resistencia total a la transferencia de calor, en los coeficientes de superficie del vapor de calentamiento y de la Solución Acuosa de Hidróxido de Sodio.

El coeficiente de película del vapor se ha tomado como 1500 BTU/HR-<sup>o</sup>F Ft.<sup>2</sup>. El coeficiente de película de la solución acuosa de Hidróxido de Sodio se ha calculado mediante la ecuación de Boarts :

$$\frac{hD}{K} = 0.0278 \left( \frac{D \mu \rho}{\mu} \right)^{0.8} \left( \frac{C_p \mu}{K} \right)^{0.4}$$

Existe bastante constancia en la literatura, sobre la confiabilidad de la aplicación de la ecuación de Boarts a sistemas como el que nos ocupa. El coeficiente de película del vapor podría ser mejorado si se calculara mediante la ecuación de Nusselt y se considerara en la misma el efecto de los no incondensables. Pero para los propósitos de esta tesis, los coefi

cientos de transferencia de calor así calculados son bastante aceptables, no siendo por lo tanto el cálculo de los mismos, ninguna fuente de falta de precisión en el modelo matemático del sistema.

Lo que en este trabajo, sí constituye definitivamente una fuente de falta de precisión, en el modelo matemático de cada una de las etapas, es el empleo de una temperatura y un coeficiente para la transferencia de calor promedios, para la evaluación del área necesaria para la transferencia de calor en el cambiador. Lo más correcto sería, evaluar la variación del coeficiente de transferencia de calor con la temperatura y determinar el área del cambiador por un método de integración numérica que podría ser por ejemplo: el Método de Simpson. En otras palabras, el área del cambiador debiera determinarse por la expresión:

$$A = W_{sol.} \bar{C}_{p_{sol.}} \int_{T_F}^{T_e} \frac{dT}{U(T_{VAPOR} - T_{sol.})}$$

Y no por la expresión:

$$A = \frac{W_{sol.} \bar{C}_{p_{sol.}}}{U_{TAV}} \int_{T_F}^{T_e} (T_e - T_F) \Delta TLM$$

donde:

$$\Delta TLM = \frac{(T_{VAPOR} - T_e) - (T_{VAPOR} - T_F)}{L \eta \frac{(T_{VAPOR} - T_e)}{(T_{VAPOR} - T_F)}} \quad \text{y} \quad TAV = \frac{T_F + T_e}{2}$$

El procedimiento de la programación dinámica que ha sido efectuado en esta tesis, tiene la característica de no contener las suposiciones altamente simplificantes que se encuentran en los trabajos que se han publicado en la literatura sobre el tema (Referencias 8 y 9). Dichas simplificaciones consisten fundamentalmente en considerar: que el calor latente de evaporación es constante y no varía con la temperatura, y que las elevaciones en el punto de ebullición de la solución o son despreciables ó varían linealmente solo con la concentración.

Para la optimización mediante la formulación simplificada de la programación dinámica se involucra: Una Variable de Estado, Una Variable de



Decisión, una función de Retorno y una ecuación de Transformación para la que existe una relación Analítica definida. La formulación rigurosa - por otra parte, involucra dos Variables de Estado, una Variable de Decisión, una Función de Retorno y una Ecuación de Transformación para la cual no existe una relación Analítica definida y que necesita aproximarse - por una técnica de aproximación numérica. Como resulta obvio, se requiere de un mayor esfuerzo, para la elaboración e implementación del programa de computadora, para la formulación rigurosa de la Programación Dinámica. Siendo también por lo tanto, los requerimientos de memoria y de procesador central de este programa, mayores que los que requiere el programa de computadora para la formulación simplificada.

Es : por lo expuesto anteriormente, por lo que se seleccionó, la formulación simplificada sobre la formulación rigurosa de la programación dinámica. La formulación simplificada tiene sin embargo, el inconveniente - de tener serias fallas conceptuales de desligamiento de la situación física; lo anterior, debido a que en cada una de las etapas (efectos) del sistema, considera como variable de Decisión : la Caída de Temperatura; siendo como ya se dijo anteriormente, una variable de decisión : aquella variable que puede ser controlada o manipulada, y la Caída de Temperatura - no lo es. Por lo tanto, una variable de decisión más adecuada para el sistema bajo estudio, sería la presión de trabajo ó el vacío en cada una de las etapas (efectos).

En el capítulo 2, en donde se estableció la teoría y se desarrollaron las ecuaciones de recursión de la programación dinámica, se mencionó la existencia de tres tipos de problemas: Del Estado Inicial, Del Estado-Final y de Los Estados Inicial - Final. En esta tesis, dado que en el sistema de evaporación cuyo cálculo se pretendía optimizar, no existían restricciones respecto a la presión del vapor vivo de calentamiento a suministrarse a la etapa inicial (efecto III), pero sí de la temperatura del agua de enfriamiento a suministrarse al condensador, para la condensación del vapor generado en la etapa final(efecto I), es que se efectuó la solución de un problema del estado final; lo anterior fué hecho con el objeto de tener una buena caída de temperatura en el condensador.

En el apéndice: al final de esta tesis, se menciona que en el programa principal del programa de computadora implementado para resolver el problema que nos ocupó durante el desarrollo de esta tesis : se efectuó un estimado inicial de los flujos de vapor generado en cada uno de los efectos y se establece el criterio de convergencia, a ser satisfecho por los conjuntos de flujos de vapor en dos iteraciones sucesivas.

El criterio de convergencia mencionado en el párrafo anterior fue finalmente quitado, debido a que se encontró que implementando el programa con diferentes estimados iniciales de los flujos de vapor, los resultados generados por el mismo, convergían siempre en una misma solución óptima en la primera iteración; por lo cual se infirió que la distribución de temperatura óptima era una, e independiente del estimado inicial con el cual se implementara el programa.

Los resultados que fueron presentados en la sección anterior, corresponden a un approach (acercamiento) de  $20^{\circ}\text{F}$  en los cambiadores de calor de cada uno de los efectos del sistema de evaporación. En este trabajo, también se trató de observar la variación en el comportamiento del sistema, por la variación de los approaches en los cambiadores de calor, pero los resultados no se muestran, debido a las formas bastante "irregulares" que se obtuvieron para los cambiadores (Número de tubos y Longitud de los mismos).

Para terminar con la presentación de resultados y conclusiones resultantes de la aplicación de la programación dinámica al sistema de evaporación, mencionaremos que la solución del problema por computadora es también bastante aceptable desde el punto de vista económico, debido al tiempo de procesador central bastante pequeño que requirió la implementación del programa (45 segundos).

### APENDICE

En este apéndice, se presenta la estructura del programa de computadora implementado para obtener la solución del problema bajo estudio; — también se discuten brevemente, algunos de los problemas presentados durante la formulación e implementación del mismo.

Se mencionó en el capítulo 3, en lo que se refería a la formulación de las ecuaciones de recursión, que el procedimiento que se seguiría implicaba la suposición, de que los flujos de vapor generado en cada uno — de los efectos variaban solamente en función de los cambios en las caídas de temperatura.

En dicho procedimiento se hace un estimado inicial del vapor generado en cada uno de los efectos por medio de la fórmula:

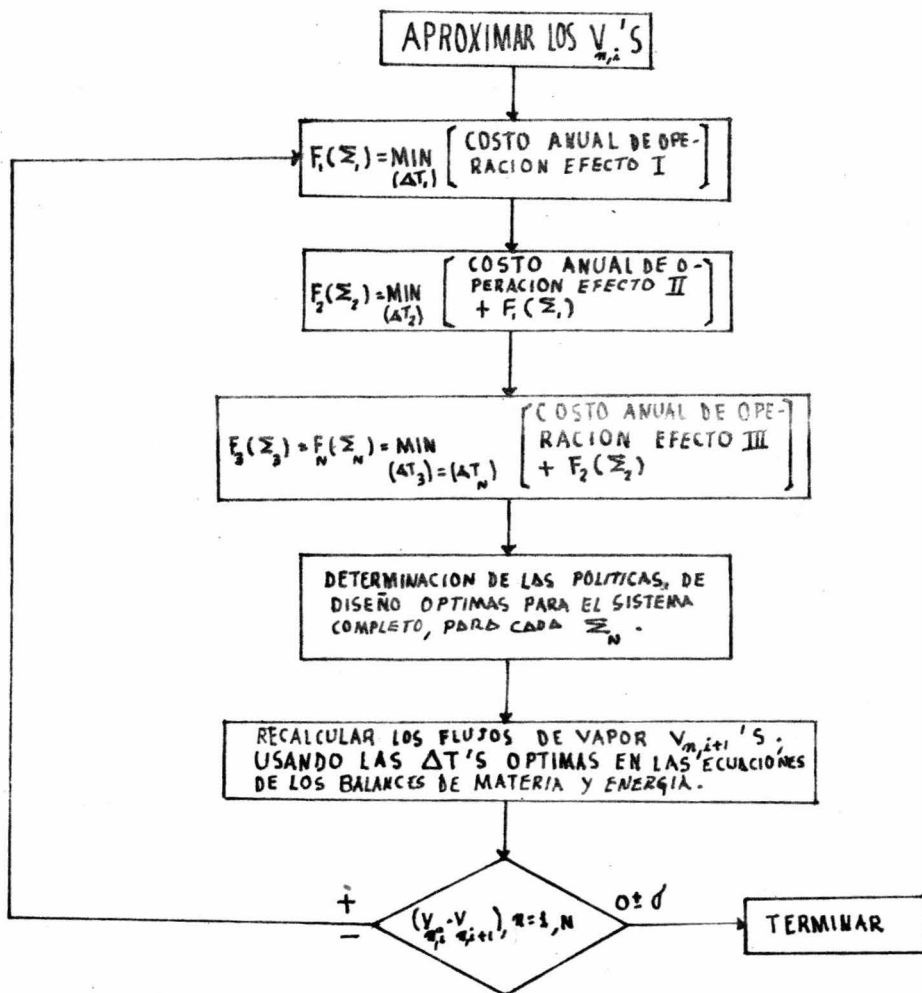
$$V_{n,i} = (L_{N+i} - L_i) / N \quad \text{donde } i = 1$$

A continuación, se usan estos flujos de vapor que se suponen ahora independientes de la temperatura, para obtener la distribución de temperaturas óptima para el sistema por Programación Dinámica. Una vez que se ha hecho lo anterior, se calcula un nuevo conjunto de  $V_{n,i}'S$  ( $V_{n,i+1}'S$ ) — usando la distribución de temperatura óptima en las ecuaciones de los balances de materia y energía. Este procedimiento termina cuando los conjuntos de  $V_{n,i}'S$  y  $V_{n,i+1}'S$  satisfacen determinado criterio de convergencia en 2 iteraciones sucesivas. El esquema computacional anterior, se muestra en el diagrama de bloques de la fig. A.1 .

Para la ejecución de los pasos involucrados en el diagrama de bloques anterior, el programa desarrollado para la solución del problema bajo consideración, consiste fundamentalmente de :

- 1.- Un Programa Principal bastante pequeño, en donde se efectúa el estimado inicial de los flujos de vapor generado en cada uno de los efectos — y donde se establece el criterio de convergencia a ser satisfecho por — los conjuntos de flujos de vapor en dos iteraciones sucesivas.
- 2.- La sección en donde se efectúa la optimización propiamente dicha del sistema por Programación Dinámica y que comprende las subrutinas :  
PRODIN, COMPLX, CHECK, CENTR, FUNC, CONST, SCONST, RETURS, TRANS, CALSII

CALSI2, CALSI3, MODEVA y CALPOT.



3.- Los estimados posteriores de los flujos de vapor generado en cada uno de los efectos, se efectúan por medio de la subrutina CALSIS( X,SN,V<sub>1</sub>1,V21,V31), al aplicar la distribución de temperatura óptima obtenida en

la iteración anterior, a las ecuaciones de los balances de materia y energía.

4.- Finalmente, la ejecución del programa, requiere el manejo extensivo de las propiedades de la solución que se está evaporando en el sistema y del vapor empleado para su calentamiento; haciendo imprescindible por lo tanto, la presencia de subrutinas que manejen dichas propiedades. En el presente programa, dichas subrutinas son fundamentalmente de 2 tipos:

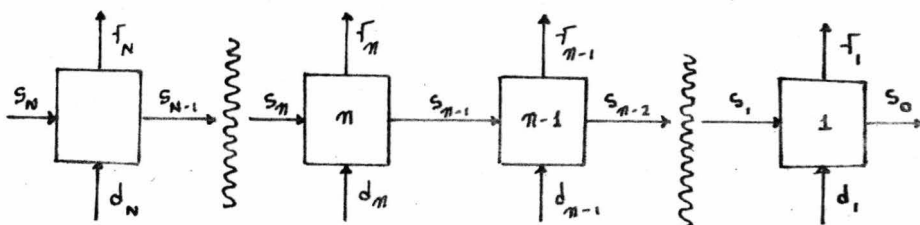
a).- Subrutinas que manejan propiedades por interpolación en tablas, como son las funciones : TLULIN y FUN2. Dichas funciones fueron tomadas de : DIGITAL COMPUTING AND NUMERICAL METHODS por Brice Carnahan and James O. Wilkes y de MODELING AND SIMULATION IN CHEMICAL ENGINEERING por Roger G. E. Franks editados por JOHN WILEY y WILEY INTERSCIENCE respectivamente. Por medio de la función TLULIN se determina por ejemplo, el calor latente de evaporación del vapor empleado en el calentamiento, en función de la temperatura del mismo, así como otras propiedades del vapor. Por medio de la función FUN2 se determina por ejemplo, la temperatura de ebullición de la solución, en función de su concentración y de la temperatura de ebullición del agua pura a la misma presión.

b).- Subrutinas que evalúan propiedades por medio de relaciones funcionales. En esta categoría se encuentran las que evalúan fundamentalmente las propiedades de la solución, como una función de la temperatura y concentración de la misma. Dichas subrutinas son: ENTAL (Entalpía), CALCCP (Capacidad Calorífica), CALCK (Conductividad Térmica), CALDEN (Densidad) y CALVIS (Viscosidad).

Vale la pena mencionar, que las relaciones funcionales que se emplean en las ecuaciones anteriores, se obtuvieron por el método de regresión lineal múltiple. Ajustando dichas relaciones funcionales, por medio de datos tomados de gráficas y tablas de datos de libros tales como: INTRODUCTION TO CHEMICAL ENGINEERING (Badger and Banchero), PROCESS HEAT TRANSFER (Kern) etc.. Y aplicando el programa para regresión lineal múltiple que se encuentra en el capítulo 6 sección I del libro OPTIMIZATION TECHNIQUES WITH FORTRAN por James L. Kuester y Joe H. Mize editado por MC GRAW HILL.

Como puede observarse en el listado del programa que se anexa al final de este apéndice, el Programa Principal, la subrutina CALSIS y las subrutinas para el manejo de propiedades, son bastante pequeñas y se explican por sí solas; por lo que el resto de este apéndice, se dedicará integralmente a la sección del programa que efectúa la optimización propiamente dicha.

El proceso que se va a optimizar, tiene una estructura que esquemáticamente puede representarse por la figura A.2 .



Donde:  $S_n$  = Temperatura del vapor de calentamiento que llega a la etapa n  
 $d_n$  = Caída de temperatura en el efecto n  
 $f_n$  = Costo anual de operación del efecto n  
 $N$  = Número de Efectos = 3

El algoritmo para dicha sección procede como sigue:

- 1).- Se determinan los límites superior e inferior de la variable de estado de entrada a la etapa 1. El intervalo del límite superior al límite inferior se divide en una serie de pasos finitos equidistantes.
- 2).- Comenzando en el límite inferior de la variable de estado de entrada a la etapa 1 ( $S_1$ ), se encuentra el retorno óptimo en el rango de valores permisibles de la variable de decisión  $d_1$ :

$$f_1(S_1) = \text{OPT}_{d_1} [f_1(S_1, d_1)]$$

El retorno óptimo para cada una de las etapas se encuentra por el Método Complejo de Box.

- 3).- la variable de estado es entonces incrementada y el óptimo es encontrado para este nuevo valor de  $S_1$ .

4).- Este proceso es repetido, hasta que se llega al límite superior de la variable de estado  $S_1$ .

5).- Comenzando con la etapa  $n = 2$ , el proceso es efectuado hacia atrás (backward), a través de la secuencia de etapas.

Se evalúan los límites para la variable de estado  $S_n$  y el intervalo del límite inferior al superior se divide en pasos equidistantes.

6).- El retorno óptimo es encontrado para cada valor de la variable de estado  $S_n$  por medio de la fórmula:

$$f_n(S_n) = \text{OPT}_{d_n} \left\{ f_n(S_n, d_n) + f_{n-1}(S_n, d_n) \right\}$$

7).- La recursión es seguida hasta que se llega a la etapa  $N$ , donde el óptimo se encuentra para cada valor posible de  $S_N$ .

8).- El algoritmo sigue entonces la trayectoria hacia adelante (forward) para recuperar las decisiones óptimas de la siguiente manera: como el problema de optimización es del estado final, en la etapa  $N$ , la variable de estado óptima  $S_N$  se encuentra en la tabla, con la función objetivo y - decisiones óptimas correspondientes. Se usa la ecuación de transformación entre  $S_N$  y  $S_{N-1}$  para encontrar la  $S_{N-1}$  óptima y sus valores correspondientes. Esta trayectoria hacia adelante (forward), se efectúa a través de todas las etapas, hasta que se llega a la variable de estado  $S_1$ , para la cual se encuentran  $f_1$  y las decisiones óptimas correspondientes.

Un diagrama de flujo ilustrando el procedimiento anterior, se muestra en la figura A.3.

A continuación, mencionaremos la función de cada una de las subrutinas que se emplean para la optimización propiamente dicha y también se describirán algunos de los parámetros más importantes de las mismas.

SUBROUTINA PRODIN (X,SN,NO,SMAX), coordina la optimización de las etapas y efectúa la recursión para encontrar la trayectoria óptima. En esta subrutina se establece: Los valores iniciales de las variables de decisión para cada una de las etapas, el número de pasos para las variables de estado, los parámetros de solución y la clave para la impresión de los resultados intermedios ó finales.

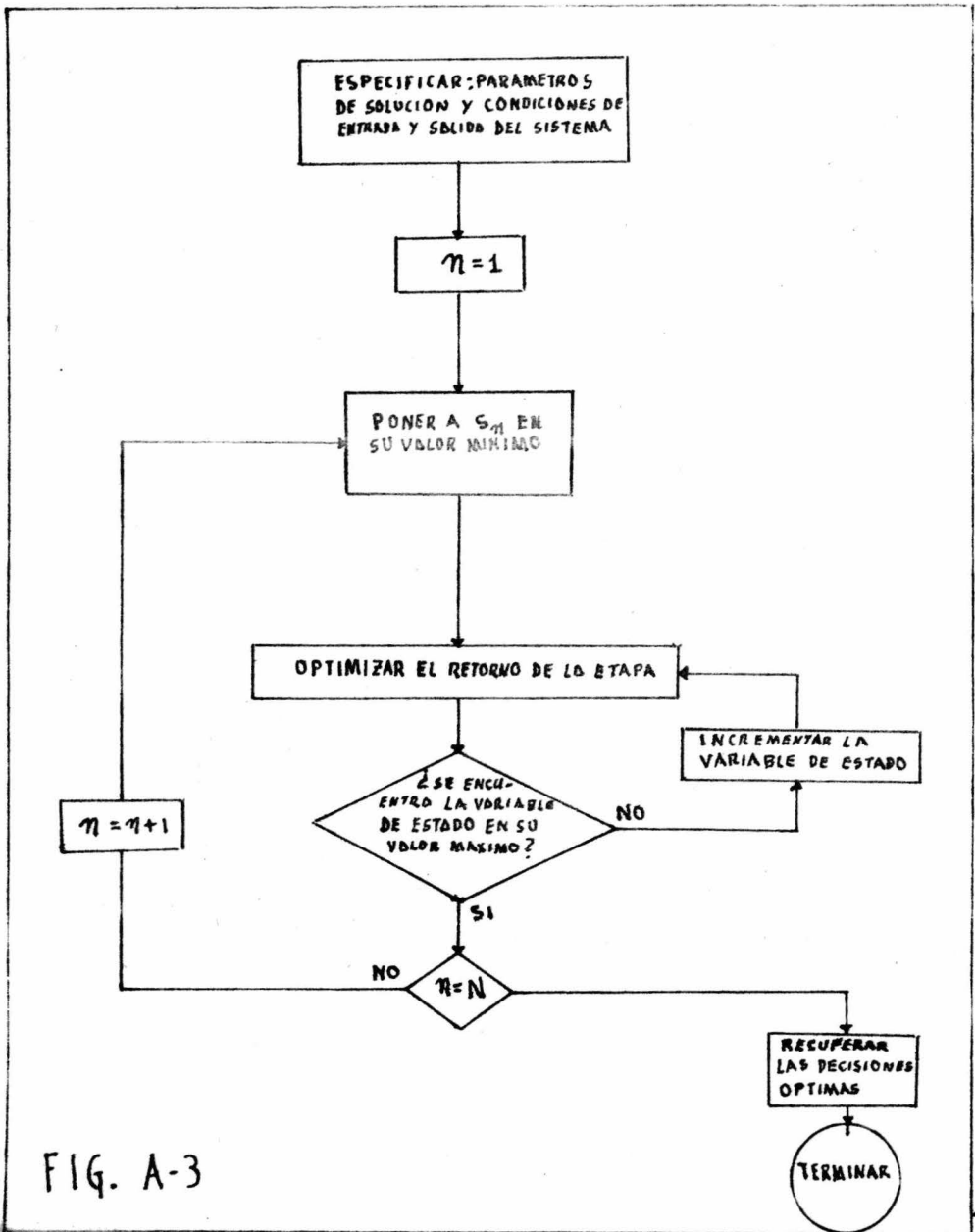


FIG. A-3



SUBROUTINA COMPLX(N,M,K,ITMAX,ALPHA,BETA,GAMMA,DELTA,X,F,BPR,IT,IEV2,NO,-  
IPRINT,R,G,H,XC,RET,BPRNIP,NUMST,IP) llamada de la Subrutina Prodi;coord  
ina las subrutinas de propósito especial(CHECK,CENTR,FUNC,CONST) y efec  
túa la búsqueda por el Método Complejo de Box, para encontrar el retorno  
óptimo para cada una de las etapas.

SUBROUTINA CHECK(N,M,K,X,G,H,I,KODE,XC,DELTA,K1) checa todos los puntos -  
contra las restricciones explícitas e implícitas y aplica correcciones -  
si es que encuentra que alguna de las restricciones ha sido violada.

SUBROUTINA CENTR(N,M,K,IEV1,I,XC,X,K1) calcula el centroide de los puntos  
del complejo.

SUBROUTINA FUNC (N,M,K,X,F,BPR,I,RET,BPRNIP) especifica la función a ser-  
optimizada en cada etapa, que en nuestro caso es la suma de los retornos  
de las etapas hasta ese punto.

SUBROUTINA CONST (N,M,K,X,G,H,I) especifica los límites explícitos e im-  
plícitos para las variables de decisión.

SUBROUTINA SCONST(N,SCON,KODE) especifica los límites superior e inferior  
para la variable de estado.

SUBROUTINA RETURS(X,I,RET,BPRNIP) especifica el retorno para cada etapa -  
, como una función de las variables de estado y de decisión(costo anual-  
de operación para el caso bajo estudio).

Las SUBROUTINAS CALSI1(X,I,WO,CO,TO,CF,PT), CALSI2(X,I,WO,CO,TO,CF,PT) y-  
CALSI3(X,I,WO,CO,TO,CF,PT) especifican y evalúan las condiciones para -  
las cuales van a determinarse los requerimientos de área para la transfe  
rencia de calor y los consumos de vapor y de potencia, a utilizarse en -  
las expresiones para el retorno de cada una de las etapas del sistema de  
evaporación.

Las SUBROUTINAS MODEVA(WO,CO,TO,CF,PT,AREA,NRE,LONGTU,G,DI,WS,BPR) y CAL-  
POT(NRE,TE,PI,L,CF,TF,G,DI,POTHPB,POTHPM,CONPOA), evalúan los requerimi-  
entos de área y los consumos de vapor y de potencia, antes mencionados.

La evaluación de los coeficientes para la transferencia de calor, -  
a emplearse en el modelo matemático de cada una de las etapas del siste-  
ma de evaporación, se efectúa por medio de las SUBROUTINAS UO(TAV,CF,VEL,  
DI,DO,RDI,RDO,HO,U,NRE) y CALHI(TAV,CF,VEL,DI,NRE).

Finalmente, la SUBROUTINA TRANS(IBAC,SMAX,DECMAX,SO,KODE) especifica la ecuación de transformación para cada una de las etapas del sistema.

DESCRIPCION DE LOS PARAMETROS Y DE LOS ARGUMENTOS MAS IMPORTANTES DE LAS SUBROUTINAS ANTERIORES.

- NSTAGE** Número de etapas en el sistema. Definido en la Subrutina PRODIN
- IPRINT** Clave para controlar la impresión de los resultados intermedios . IPRINT = 2, hace que las tablas de valores se impriman después de que cada etapa es optimizada. IPRINT = 1, hace que todos los valores buscados y tablas se impriman. IPRINT = 0, suprime la impresión hasta que se obtiene la solución final. Se define en la Subrutina PRODIN.
- IPROB** Clave para definir el tipo de problema. IPROB = 1, para problemas del estado inicial. IPROB = 2, para problemas del estado final. IPROB = 3, para problemas de los estados inicial-final. Se define en la Subrutina PRODIN.
- IOPT** Clave de la función objetivo en el óptimo. IOPT = -1 para mínimo. IOPT = +1 para máximo. Se define en la Subrutina PRODIN.
- SN** Variable de estado de entrada al sistema. Definida en la Subrutina PRODIN.
- NDECIS** Número de variables de decisión en cada etapa del sistema. Definida en la Subrutina PRODIN.
- M** Número de restricciones para las variables de decisión en cada una de las etapas. Definida en la Subrutina PRODIN.
- NSTEP** Número de pasos en que se va a dividir el intervalo de la variable de estado. Se define en la Subrutina PRODIN.
- DECIS** Variables de decisión. El valor inicial de estas variables de decisión, se encuentra definido en la Subrutina PRODIN.
- K** Número de puntos en el complejo. Definido en la Subrutina PRODIN
- ALPHA** Factor de Reflexión. Definido en la Subrutina PRODIN.
- BETA** Parámetro de Convergencia. Definido en la Subrutina PRODIN.
- GAMMA** Parámetro de Convergencia. Definido en la Subrutina PRODIN.

- DELTA Corrección por violación a un límite explícito. Definida en la subrutina PRODIN.
- F Función objetivo-Definida en la subrutina FUNC.
- IT Índice de Iteración. Definido en la subrutina COMPLX.
- IEV2 Índice del punto con el valor más bajo de la función. Definido en la subrutina COMPLX.
- IEV1 Índice del punto con el valor más grande de la función. Definido en la subrutina COMPLX.
- NI Número de la unidad lectora de tarjetas. Definido en el programa principal.
- NO Número de la unidad impresora. Definido en el programa principal
- R Números aleatorios entre 0 y 1. Definidos en la subrutina COM--PLX.
- G Límites inferiores para las variables de decisión. Definidos - en la subrutina CONST.
- H Límites superiores para las variables de decisión. Definidos en la subrutina CONST.
- XC Centroide. Definido en la subrutina CENTR.
- I Índice de punto. Definido en la subrutina COMPLX.
- KODE Clave usada para determinar si existen restricciones implícitas para las variables de decisión. Definida en la subrutina COMPLX. También se usa para la determinación de los límites superior e inferior, de la variable de estado. Definida en la subrutina PRO DIN.
- K1 Límite de un "Do Loop": Definido en la subrutina COMPLX.
- SCON Valor límite de la variable de estado. Definido en la subrutina-SCONST.
- IBAC Índice de la etapa. Definido en la subrutina PRODIN y TRANS.
- SMAX Valores óptimos de las variables de estado. Definidos en las subrutinas PRODIN Y TRANS.
- BPR Elevación en el punto de ebullición. Definido en las subrutinas-

MODEVA Y RETURS.

- WO Gasto de la solución de alimentación a un efecto cualquiera del sistema. Definido en las subrutinas CALSI1, CALSI2, CALSI3 y MODEVA.
- TO Temperatura de la solución de alimentación al efecto. Definida en las subrutinas CALSI1, CALSI2, CALSI3 y MODEVA.
- CO Concentración de la solución de alimentación al efecto. Definida en la subrutina CALSI1, CALSI2, CALSI3 y MODEVA.
- CF Concentración de la solución a la salida de un efecto. Definida en las subrutinas CALSI1, CALSI2, CALSI3, MODEVA, CALPOT, UO y -CALHI.
- PT Presión de Trabajo en el efecto. Definida en las subrutinas CALSI1, CALSI2, CALSI3, MODEVA y CALPOT.
- ARSA Area del combinador de calor, de uno cualquiera de los efectos.- Definida en las subrutinas MODEVA Y RETURS.
- NRE Número de Reynolds en los tubos del cambiador de calor. Definido en las subrutinas MODEVA, CALPOT y CALHI.
- NTUBOS Número de tubos del cambiador de calor. Definido en la subrutina MODEVA y CALPOT.
- LONGTU Longitud de los tubos del cambiador de calor. Definida en las sub-rutinas MODEVA y CALPOT.
- WS Consumo de vapor de calentamiento. Definido en las subrutinas MODEVA Y RETURS.
- G Masa de Velocidad en los tubos del cambiador de calor. Definida en las subrutinas MODEVA, UO, CALHI y CALPOT.
- POTHPB Caballaje de la bomba en HP's. Definido en la subrutina CALPOT.
- CONPOA Consumo anual de potencia KW-HR/ANO. Definido en las subrutinas CALPOT Y RETURS.
- POTHEM Caballaje del motor en HP's. Definido en la subrutina CALPOT
- EFBOMB Eficiencia de la bomba. Definida en la subrutina CALPOT.
- EFMOTO Eficiencia del motor. Definida en la subrutina CALPOT.

- TAV      Temperatura promedio, entre las temperaturas de entrada y de salida del cambiador de calor. Definida en las subrutinas MODEVA, UO y CALHI.
- VEL      Velocidad en los tubos del cambiador de calor. Definida en las subrutinas MODEVA, UO y CALHI.
- HI      Coeficiente individual para la transferencia de calor de la solución. Definido en las subrutinas MODEVA, UO y CALHI.
- HO      Coeficiente individual de transferencia de calor del vapor de calentamiento. Definido en las subrutinas MODEVA y UO.
- RDI      Factor de incrustación interno. Definido en las subrutinas MODEVA y UO.
- RDO      Factor de incrustación externo. Definido en las subrutinas MODEVA y UO.

El programa de computadora implementado para resolver el problema que nos ocupó, durante el desarrollo de esta tesis. Se encuentra basado fundamentalmente, en el programa general para el algoritmo de la Programación Dinámica, que se encuentra en el libro OPTIMIZATION TECHNIQUES — WITH FORTRAN Págs. 193-202 por James L. Kuester y Joe H. Mize, editado por Mc Graw Hill.

Las subrutinas FUNC, RETURNS, SCONST y TRANS del programa mencionado, fueron ajustadas íntegramente al problema bajo consideración. La subrutina PRODIN de nuestro programa, que corresponde al programa principal del programa de computadora que se encuentra en el libro antes mencionado, también fue adaptada al problema bajo estudio, sobre todo, en la parte donde se efectúa la recuperación de las decisiones óptimas. De tal forma, que del programa que se encuentra en dicho libro, sólo permanecen inalteradas las subrutinas COMPLX, CHECK y CNTR, que son parte integral de la búsqueda del óptimo de las etapas, por el Método Complejo de Box.

Obviamente, al programa base antes mencionado, se agregaron las subrutinas indispensables para la solución del problema específico que nos ocupó durante el desarrollo de esta tesis y que son: subrutinas para el manejo de propiedades físicas y las subrutinas para la solución del modelo

matemático del sistema de evaporación.

NOTA.- El programa cuya descripción ha sido el objeto de este apéndice, se implementó mediante el compilador de Fortran Extendido de la Burroughs 6700 del Centro de Servicios de Computo de la Universidad Nacional Autónoma de México.

LISTADO DEL PROGRAMA

-----

-----

-----

-----

-----

-----

-----

-----

-----

```

*****
*
* UNIVERSIDAD NACIONAL AUTONOMA
* CENTRO DE SERVICIOS DE
*
* COMPILADOR FORTRAN, NIVEL 2.5.015, BUR
*
* JUEVES 31 DE OCTUBRE DE 1974
*
*****

```

NOMBRE DEL PROGRAMA = PROG/DINA

```

FILE 6=FILE6,UNIT=PRINTER
FILE 5=FILE5,UNIT=READER
DIMENSION X(3,3),SMAX(3)
COMMON /V1V2V3/ V1, V2, V3
COMMON /AA/ PTSTAR(65), TWSTAR(65), BTH(56), CSTAR(56), TSTAR(56),
1HPRINV(65), HFG(65)
COMMON /CC/ BPSOL(64), CSOL(64), TWATER(64)
COMMON /ZZ/ N, S(3,21), IP, SD, FDP(20,3,21), NTDTN, NTDTM1, IOPT
COMMON /KKKKK/ KKK
LOGICAL WAR
LOGICAL VAR
REAL L1, L2, L3, L4, LAHTW2
NI = 5
NO = 6
READ (NI,035) (PTSTAR(I), I=1,65)
035 FORMAT (8F10.4)
READ (NI,034) (TWSTAR(I), I=1,65)
034 FORMAT (8F10.2)
READ (NI,031) (BTH(I), I=1,56)
031 FORMAT (8F10.3)
READ (NI,032) (GSTAR(I), I=1,56)
032 FORMAT (8F10.3)
READ (NI,033) (TSTAR(I), I=1,56)
033 FORMAT (8F10.2)
READ (NI,036) (HPRINV(I), I=1,65)
036 FORMAT (8F10.1)
READ (NI,037) (HFG(I), I=1,65)
037 FORMAT (8F10.1)
READ (NI,038) (BPSOL(I), I=1,64)
038 FORMAT (8F10.2)
READ (NI,039) (CSOL(I), I=1,64)
039 FORMAT (8F10.4)
READ (NI,041) (TWATER(I), I=1,64)
041 FORMAT (8F10.2)
L4 = 250000.00
C4 = 0.100

```



```

C1 = 0.200
T1 = 110.00
L1 = (L4 * C4)/C1
V = (L4 - L1)/3
V1 = V
V2 = V
V3 = V
KKK = 1
5 CONTINUE
CALL PRUDIN(X,SN,NI,NU,SHAX)
CALL CALSIS(X,SN,V1,V2,V3,SMAX)
WRITE(6,55) V1,V2,V3,V11,V21,V31
55 FORMAT(6F12.2)
V1 = V31
V2 = V21
V3 = V11
KKK = KKK + 1
IF(KKK.GT.2) GO TO 500
GO TO 5
500 CONTINUE
CALL EXIT
END

```

MAT  
 S1

```

SUBROUTINE CALSIS(X,SN,V11,V21,V31,SMAX)
DIMENSION X(3,3),SMAX(3)
COMMON /AA/ PISTAR(65), TWSTAR(65), BTW(56), CSTAR(56), T
HPRIMV(65), HFG(65)
COMMON /CC/ BPSOL(64), CSOL(64), TWATER(64)
COMMON /ZZ/ NPS(3,21), IP,SO,FDP(20,3,21),NTOTN,NTOTH1,IOPT
LOGICAL FLAG
REAL L1, L2, L3, L4, LANW2
L4 = 250000.00
C4 = 0.100
C1 = 0.200
TI = 110.00
L1 = (L4*C4)/C1
T1 = FUN2(SO,C1,56,8,BTW,CSSTAR,TSTAR)
T2 = SMAX(2) - X(2,1)
TW2 = SMAX(1)
C2 = 0.11
5 CONTINUE
T22 = FUN2(TW2,C2,56,8,BTW,CSSTAR,TSTAR)
IF((T2-T22)/0.2)10,15,20
10 IF(ABS(T2-T22).LE.0.2) GO TO 15
C2 = C2 - 0.005
GO TO 5
20 IF(ABS(T2-T22).LE.0.2) GO TO 15
C2 = C2 + 0.005
GO TO 5
15 CONTINUE
L2 = (L1*C1)/C2
V11 = L2 - L1
HG1 = TLULIN(TWSTAR,HPRIMV,SO,65,FLAG)
HV1 = HG1 + 0.46*(T1-SO)
CALL ENTAL (T1,C1,HL1)
CALL ENTAL (T2,C2,HL2)
Q1 = L1 * HL1 - L2 * HL2 + V11 * HV1
LANW2 = TLULIN(TWSTAR,HFG,TW2,65,FLAG)
V21 = (Q1)/(LANW2)
L3 = L2 + V21
V31 = L4 - L3
RETURN
END

```

```

SUBROUTINE PRODIN(X,SN,NI,NO,SHAX)
PROGRAMA PRINCIPAL PARA EL ALGORITMO DE LA PROGRAMACION DINAMICA.
DIMENSION NDECIS(3),X(3,3),F(3),DECMAX(3,3),NTOT(3),SHAX(3),R(3,1)
1,G(3),H(3),XC(2),RETMAX(3),BPR(3),BPRMAX(3),RET(3,21),BPRNIP(3,21)
2,DECIS(3,1,21)

```

```
COMMON /ZZ/ N,S(3,21), IP, SO, FDP(20,3,21),NTOTH,NTOTH1,IOPT
```

```
COMMON /KKKK/ KKK
```

```
INTEGER GAMMA
```

```
NSTAGE = 3
```

```
ITMAX = 100
```

```
IPRINT = 2
```

```
IPROB = 2
```

```
IOPT = -1
```

```
IPHOLD = IPRINT
```

```
SN = 252.00
```

```
SO = 115.00
```

```
ALPHA = 1.3
```

```
BETA = 50.00
```

```
GAMMA = 5
```

```
DELTA = 0.0001
```

```
FMAX = 0.00
```

```
WRITE (NO,015)
```

```
015 FORMAT (10X,"PROCEDIMIENTO DE LA PROGRAMACION DINAMICA")
```

```
GO TO (220,230,240), IPROB
```

```
220 WRITE (NO,021) NSTAGE
```

```
021 FORMAT (///,2X,"OPTIMIZACION DE UN SISTEMA DE ",I2,"ETAPAS EN FUNCION DE SU ESTADO INICIAL")
```

```
GO TO 250
```

```
230 WRITE (NO,022) NSTAGE
```

```
022 FORMAT (///,2X,"OPTIMIZACION DE UN SISTEMA DE ",I2,"ETAPAS EN FUNCION DE SU ESTADO FINAL")
```

```
GO TO 250
```

```
240 WRITE (NO,023) NSTAGE
```

```
023 FORMAT (///,2X,"OPTIMIZACION DE UN SISTEMA DE ",I2,"ETAPAS EN FUNCION DE LOS ESTADOS INICIAL Y FINAL DEL SISTEMA")
```

```
250 CONTINUE
```

```
C ANALISIS RECURSIVO DE LAS ETAPAS PARTIENDO DE LA ETAPA N = 1
```

```
N = 1
```

```
10 NS = N
```

```
NDECIS(1) = 1
```

```
NDECIS(2) = 1
```

```
NDECIS(3) = 1
```

```
M = 3
```

```
NSTEP = 20
```

```
K = 3*NDECIS(N)
```

```
NDEC = NDECIS(N)
```

```
DECIS(1,1,1)=30.00
```

```
DECIS(2,1,1)=30.00
```

```
DECIS(3,1,1)=30.00
```

```
C
```

```
KODE = 0
```

```
CALL SCONST (N, SLOW, KODE)
```

```
KODE = 1
```

```
CALL SCONST (N, SHIGH, KODE)
```

```
NTOT(N) = NSTEP + 1
```

```
STEP = (SHIGH - SLOW)/NSTEP
```

```
NTOTN = NTOT(N)
```

```

IF (N.GE.2) NTOTM1 = NTOT(N-1)
C
C
C
EJECUCION DE LA BUSQUEDA EN LA ETAPA N PARA UN ESTADO DE ENTRADA
DADO.
C
DO 100 IP = 1,NTOTN
C
IF (IPRINT = 1) 28, 28, 26
26 IPRINT = 0
28 IM1 = IP - 1
S(N,IP) = SLOW + IM1 * STEP
C
DO 30 J = 1,NDEC
30 X(1,J) = DECIS(N,J,1)
C
CALL COMPLX(NDEC,M,K,ITHAX,ALPHA,BETA,GAMMA,DELTA,X,F,BPR,IT,IEV2,
1NO,IPRINT,R,G,H,XC,RET,BPRNIP,NUHST,IP)
C
FDP(KKK,N,IP) = IOPT*F(IEV2)
DO 40 J=1,NDEC
40 DECIS(N,J,IP) = X(IEV2,J)
IF (IPHOLD = 1) 100, 45, 45
45 IPRINT = IPHOLD
50 IF (IP = NTOTN) 100,55, 55
55 WRITE(N0,003) N
003 FORMAT (//,2X,"TABLA DE VALORES PARA LA ETAPA",I2)
WRITE (N0,005)
005 FORMAT (//,2X," VARIABLE DE ESTADO",7X,"VALOR DE LA FUNCION",9X,
1"VARIABLE DE DECISION",/)
DO 60 II = 1,NTOTN
60 WRITE(N0,004) S(N,II), FDP(KKK,N,II),
1(DECIS(N,J,II), J=1,NDEC)
004 FORMAT (2X,1PE14.6,4X,1PE14.6,3(4X,1PE14.6))
C
100 CONTINUE
IF (N = NSTAGE) 110, 120, 120
110 N = N + 1
GO TO 10
C
C
C
SOLUCION "HACIA ADELANTE PARA ENCONTRAR LA TRAYECTORIA OPTIMA"
C
120 DO 200 II = 1,NSTAGE
IBAC = NSTAGE + 1 - II
NTOTN = NTOT(IBAC)
NDEC = NDECIS(IBAC)
SMAX(IBAC) = S(IBAC,NTOTN)
RETMAX(IBAC) = RET(IBAC,NTOTN)
BPRMAX(IBAC) = BPRNIP(IBAC,NTOTN)
DO 128 K=1,NDEC
128 DECMAX(IBAC,K) = DECIS(IBAC,K,NTOTN)
IF (IBAC.NE.NSTAGE) GO TO 160
FMAX = FDP(KKK,IBAC,NTOTN)
DO 150 J=1,NTOTN
GO TO (122,124,122), IPROB
122 IF(S(IBAC,J) = SN) 150, 130, 130
124 IF(FDP(KKK,IBAC,J)=FMAX)150,130,130
130 FMAX = FDP(KKK,IBAC,J)
DO 140 K=1,NDEC
140 DECMAX(IBAC,K) = DECIS(IBAC,K,J)
SMAX(IBAC) = S(IBAC,J)
RETMAX(IBAC) = RET(IBAC,J)

```

```

    BPRMAX(IBAC) = BPRNIP(IBAC,J)
    IF (IPROB.NE.2) GO TO 200
150 CONTINUE
    GO TO 200
160 KODE = 0
    CALL TRANS (IBAC,SMAX,DECHMAX,BPRMAX,SO,KODE)
    DO 190 J=1,NTOTN
    IF (S(IBAC,J) = SMAX(IBAC)) 190, 170, 170
170 SMAX(IBAC) = S(IBAC,J)
    RETMAX(IBAC) = RET(IBAC,J)
    BPRMAX(IBAC) = BPRNIP(IBAC,J)
    DO 180 K = 1,NDEC
180 DECHMAX(IBAC,K) = DECIS(IBAC,K,J)
    GO TO 200
190 CONTINUE
200 CONTINUE
    KODE = 1
    CALL TRANS (IBAC,SMAX,DECHMAX,BPRMAX,SO,KODE)
C
    DO 300 N=1,NSTAGE
    NDEC = NDECIS(N)
    DO 210 II = 1,NDEC
210 X(N,II) = DECHMAX(N,II)
    IP = 1
    CALL RETURS(X,N,RET,BPRNIP)
300 CONTINUE
    WRITE (NO,011) KKK, FMAX
011 FORMAT (///,"NO. DE CORRIDA ",2X,I2,2X,"RETORNO OPTIMO DEL SISTEMA
1 = ",1PE16.8)
    WRITE (NO,012)
012 FORMAT (///,10X,"RETORNDS MAXIMOS DE LAS ETAPAS")
    DO 400 I=1,NSTAGE
400 WRITE(NO,013) I, RETMAX(I)
013 FORMAT (///,2X,"ETAPA",I2,"RETORNO = ",1PE16.8)
    WRITE(NO,014)
014 FORMAT (///,10X,"DECISIONES OPTIMAS")
    DO 500 J=1,NSTAGE
    NDEC = NDECIS(J)
    WRITE (NO,016) J, (J, I, X(J,I), I=1,NDEC)
016 FORMAT (/,2X,6HSTAGE ,I2,6X,2(4X,2HX(I,1,1H,,I1,4H) = ,1PE16.8))
500 CONTINUE
    WRITE (NO,018) SO
018 FORMAT (/,2X,"ESTADO DE SALIDA DEL SISTEMA = ",1PE16.8)
    NSTM1 = NSTAGE - 1
    DO 600 JJ=1,NSTM1
    WRITE(NO,019) JJ, SMAX(JJ)
019 FORMAT (/,2X,"ESTADO DE ENTRADA A LA ETAPA ",I2," = ",1PE16.8)
600 CONTINUE
    SN = SMAX(NSTAGE)
    WRITE (NO,017) SN
017 FORMAT (/,2X,"ESTADO DE ENTRADA AL SISTEMA = ",1PE16.8)
    RETURN
    END

```

FORM

```

SUBROUTINE COMPLX (N,M,K,ITMAX,ALPHA,BETA,GAMMA,DELTA,X,F,BPR,IT,I
C
C 1EV2,NO,IPRINT,R,G,H,XC,RET,BPRNIP,NUMST,IP)
COORDINA A LAS SUBROUTINAS DE PROPOSITO ESPECIAL
C
C LISTA DE ARGUMENTOS
C
C IT = INOICE DE ITERACION
C
C IEV1 = INDICE DEL PUNTO CON EL VALOR MINIMO DE LA FUNCION.
C
C IEV2 INDICE DEL PUNTO CON EL VALOR MAXIMO DE LA FUNCION.
C
C I = INDICE DEL PUNTO
C
C KODE = CLAVE USADA PARA DETERMINAR SI EXISTEN LIMITES IMPLICITOS
C
C K1 = LIMITE DE UN DO LOOP
C
C TODOS LOS OTROS ARGUMENTOS SE ENCUENTRAN DEFINIDOS EN EL PROGRAMA
C PRINCIPAL.
C
C DIMENSION X(K,M),R(K,N),F(K),G(M),H(N),XC(N),RET(3,21),BPRNIP(3,21
1),BPR(K)
C
C INTEGER GAMMA
C
C
C IT = 1
C
C KODE = 0
C
C IF (M=N) 20,20,10
10 KODE = 1
20 CONTINUE
C
C DO 40 II=2,K
C
C DO 30 JJ=1,N
30 X(II,J) = 0.0
40 CONTINUE
C
C
C DO 45 II=2,K
C
C DO 45 JJ=1,N
C
C
C GENERE NUMEROS ALEATORIOS PARA CALCULAR EL COMPLEJO.
C
C
C R(II,JJ) = RANDOM(Z)
45 CONTINUE
C
C
C IF (IPRINT) 46, 48, 46
46 WRITE(NO,001)
001 FORMAT (2X,"PARAMETROS")
WRITE (NO,002) N, M, K, ITMAX, IC, ALPHA, BETA, GAMMA, DELTA
002 FORMAT (//,2X,4H N = ,I2,3X,4H M = ,I2,3X,4H K = ,I2,2X,8HITMAX = ,
1I4,2X,5HIC = ,I2,//,2X,8HALPHA = ,F5.2,5X,7HBETA = ,F10.5,3X,
28HGAMMA = ,I2,3X,8HDELTA = ,E12.6)
WRITE (NO,003)
003 FORMAT (//,2X,"NUMEROS ALEATORIOS")
DO 47 JJ=2,K
WRITE (NO,004) (J, I, R(J,I), I=1,N)
004 FORMAT (/,2X,3(2HR(,I2,1H,,I2,4H) = ,F6.4,2X))
47 CONTINUE
C
C ENCUENTRE LOS PUNTOS DEL COMPLEJO Y CHEQUELOS CONTRA LOS LIMITES.
C
C
C 48 DO 65 II=2,K
C
C DO 50 JJ=1,N
C
C I = II
C
C CALL CONST (N,M,K,X,G,H,I)
C
C X(II,J) = G(J) + R(II,J) * (H(J) - G(J))
50 CONTINUE
C
C K1 = II
C
C CALL CHECK (N,M,K,X,G,H,I,KODE,XC,DELTA,K1)
C
C IF (II=2) 51, 51, 55
51 IF (IPRINT) 52, 65, 52

```

```

52 WRITE (NO,018)
018 FORMAT (//,2X,"COORDENADAS DEL COMPLEJO INICIAL")
   IO = 1
   WRITE (NO,019) (IO, J, X(IO,J), J = 1,N)
019 FORMAT (/,3(2X,2HX(I2,1H,,I2,4H) = ,1PE13.6))
55 IF (IPRINT) 56, 65, 56
56 WRITE (NO,019) (II,J,X(II,J), J=1,N)
65 CONTINUE
   KI = K
   DO 70 I=1,K
   CALL FUNC(N,M,K,X,F,BPR,I,RET,BPRNIP)
70 CONTINUE
   KOUNT = 1
   IA = 0
C   BUSQUEDA DEL PUNTO CON EL VALOR MINIMO DE LA FUNCION
   IF (IPRINT) 72, 80, 72
72 WRITE (NO,021)
021 FORMAT (/,"VALORES DE LA FUNCION")
   WRITE (NO,022) (J, F(J), J=1,K)
022 FORMAT (/,3(2X,2HF(I2,4H) = ,1PE13.6))
80 IEV1 = 1
   DO 100 ICM = 2,K
   IF (F(IEV1) - F(ICM)) 100, 100, 90
90 IEV1 = ICM
100 CONTINUE
C
C   BUSQUEDA DEL PUNTO CON EL VALOR MAXIMO DE LA FUNCION.
   IEV2 = 1
   DO 120 ICM=2,K
   IF (F(IEV2) - F(ICM)) 110, 110, 120
110 IEV2 = ICM
120 CONTINUE
C   CHEQUEO DEL CRITERIO DE CONVERGENCIA.
C
   IF (F(IEV2) - F(IEV1) - (ABS(BETA*F(IEV1)))) 140, 130, 130
130 KOUNT = 1
   GO TO 150
140 KOUNT = KOUNT + 1
   IF (KOUNT-GAMMA) 150,240,240
C   REEMPLAZAMIENTO DEL PUNTO CON EL VALOR MINIMO DE LA FUNCION.
150 CALL CENTR (N,M,K,IEV1,I,XC,X,K1)
   DO 160 JJ=1,N
160 X(IEV1,JJ) = (1.0+ALPHA)*XC(JJ)-ALPHA*(X(IEV1,JJ))
   I = IEV1
   CALL CHECK (N,M,K,X,G,H,I,KODE,XC,DELTA,K1)
   CALL FUNC(N,M,K,X,F,BPR,I,RET,BPRNIP)
CC  REEMPLAZAMIENTO DEL NUEVO PUNTO SI ES QUE CONTINUA PRODUCIENDO
C   EL VALOR MINIMO DE LA FUNCION.
170 IEV2 = 1
   DO 190 ICM=2,K
   IF (F(IEV2) - F(ICM)) 190, 190, 180
180 IEV2 = ICM
190 CONTINUE
   IF (IEV2-IEV1) 220,200,220
200 DO 210 JJ=1,N
   X(IEV1,JJ) = (X(IEV1,JJ) + XC(JJ))/2.0
210 CONTINUE
   I = IEV1
   CALL CHECK (N,M,K,X,G,H,I,KODE,XC,DELTA,K1)
   CALL FUNC(N,M,K,X,F,BPR,I,RET,BPRNIP)

```

```
220 CONTINUE
    IF (IPRINT) 230,228,230
230 WRITE(NU,023) IT
023 FORMAT(/,2X,"NUMERO DE ITERACION",15)
    WRITE (NO,024)
024 FORMAT (/,2X,"COORDENADAS DEL PUNTO CORREGIDO")
    WRITE (NO,019) (IEV1, JC, X(IEV1,JC), JC=1,N)
    WRITE (NO,021)
    WRITE (NO,022) (I, F(I), I=1,K)
    WRITE (NO,025)
025 FORMAT (/,2X,"COORDENADAS DEL CENTROIDE")
    WRITE (NO,026) (JC, XC(JC), JC=1,N)
026 FORMAT (/,2X,3(2HX(,I2,6H,C) = ,1PE14.6,4X))
228 IT = IT + 1
    IF (IT = ITMAX) 80, 80, 240
240 RETURN
    END
```



```
SUBROUTINE CENTR (N,M,K,IEV1,I,XC,X,K1)
DIMENSION X(K,M), XC(N)
```

C

```
DO 20 J=1,N
XC(J) = 0.0
DO 10 IL=1,K1
10 XC(J) = XC(J) + X(IL,J)
RK = K1
20 XC(J) = (XC(J) - X(IEV1,J))/(RK - 1.0)
RETURN
END
```

```

SUBROUTINE CHECK (N,M,K,X,G,H,I,KODE,XC,DELTA,K1)
C
C   LISTA DE ARGUMENTOS
C   TODOS LOS ARGUMENTOS SE ENCUENTRAN DEFINIDOS EN EL PROGRAMA PRIN=
C   CIPAL Y EN LA SUBROUTINA COMPLX
C   DIMENSION X(K,M), G(M), H(M), XC(M)
10  KT = 0
    CALL CONST (N,M,K,X,G,H,I)
C
C   CHEQUEO CONTRA LIMITES EXPLICITOS.
    DO 50 J=1,N
      IF (X(I,J)-G(J)) 20, 20, 30
20  X(I,J) = G(J) + DELTA
      GO TO 50
30  IF (H(J) - X(I,J)) 40, 40, 50
40  X(I,J) = H(J) - DELTA
50  CONTINUE
      IF (KODE) 110,110,60
C   CHEQUEO CONTRA LIMITES IMPLICITOS.
60  NN = N + 1
      DO 100 J=NN,M
        CALL CONST (N,M,K,X,G,H,I)
        IF (X(I,J)-G(J)) 80, 70, 70
70  IF (H(J)-X(I,J)) 80, 100, 100
80  IEV1 = I
          KT = 1
          CALL CENTR (N,M,K,IEV1,I,XC,X,K1)
          DO 90 JJ=1,N
            X(I,JJ) = (X(I,JJ) + XC(JJ))/2.0
90  CONTINUE
100 CONTINUE
      IF (KT) 110, 110, 10
110 RETURN
    END

```

```

SUBROUTINE FUNC(N,M,K,X,F,BPR,I,RET,BPRNIP)
DIMENSIONX(3,3),F(3),BPR(3),RET(3,21),BPRNIP(3,21)
DIMENSION SFUNC(3), XFUNC(3,3), BPFUNC(3), DECIS(3,1,21)
COMMON /ZZ/ NUMST,S(3,21),IP,SO,FDP(20,3,21),NTDTN,NTDTM1,IOPT
COMMON /KKKKK/ KKK

```

C

```

CALL RETURS(X,I,RET,BPRNIP)
IF (NUMST = 1) 10, 10, 20
10 F(I) = RET(NUMST,IP)
   BPR(I) = BPRNIP(NUMST,IP)
   GO TO 99
20 NUMT = NUMST - 1
   KODE = 0
   FSTM1 = S(NUMT,NTDTM1)
   SFUNC(NUMST) = S(NUMST,IP)
   BPFUNC(NUMST) = BPRNIP(NUMST,IP)
   DO 30 JF=1,N
30 XFUNC(NUMST,JF) = X(I,JF)
   CALL TRANS (NUMT,SFUNC,XFUNC,BPFUNC,SO,KODE)
   DO 50 IS = 1,NTDTM1
   IF (S(NUMT,IS) = SFUNC(NUMT)) 50, 40, 40
40 FSTM1 = FDP(KKK,NUMT,IS)
   GO TO 60
50 CONTINUE
60 GO TO (1,2), NUMT
   1 F(I) = RET(NUMST,IP) + FSTM1
   BPR(I) = BPRNIP(NUMST,IP)
   GO TO 99
   2 F(I) = RET(NUMST,IP) + FSTM1
   BPR(I) = BPRNIP(NUMST,IP)
99 F(I) = IOPT * F(I)
   RETURN
   END

```

```
SUBROUTINE SCONST (N, SCON, KODE)
COMMON /ZZ/ NUMST,S(3,21),IP,SO,FDP(20,3,21),NTOTN,NTOTM1,IQPT
IF (KODE) 100, 100, 200
100 GO TO (1,2,3), N
1 SCON = 150.00
GO TO 99
2 SCON = 175.00
GO TO 99
3 SCON = 225.00
GO TO 99
200 GO TO (4,5,6), N
4 SCON = 170.00
GO TO 99
5 SCON = 215.00
GO TO 99
6 SCON = 252.00
99 RETURN
END
```

```
SUBROUTINE CONST (N, M, K, X, G, H, I)
DIMENSION X(K,M), G(M), H(M)
COMMON /ZZ/ NUMST, S(3,21), IP, SD, FDP(20,3,21), NTOTN, NTOTH1, IOPT
GO TO (1,2,3), NUMST
1 G(1) = S(NUMST,IP) * SD = 12.00
H(1) = S(NUMST,IP) * SD = 11.00
GO TO 99
2 G(1) = 25.00
H(1) = 35.00
GO TO 99
3 G(1) = 25.00
H(1) = 35.00
99 RETURN
END
```

SUBROUTINE TRANS (IBAC,SMAX,DECMAX,BPRMAX,SO,KODE)  
DIMENSION SMAX(3), DECMAX(3,1), BPRMAX(3)

C

IF (KODE) 10, 10, 20  
10 IBACP1 = IBAC + 1  
GO TO (1,2), IBAC  
1 SMAX(IBAC) = SMAX(IBACP1) - DECMAX(IBACP1,1) - BPRMAX(IBACP1)  
GO TO 99  
2 SMAX(IBAC) = SMAX(IBACP1) - DECMAX(IBACP1,1) - BPRMAX(IBACP1)  
GO TO 99  
20 SO = SMAX(1) - DECMAX(1,1) - BPRMAX(1)  
99 RETURN  
END

```

SUBROUTINE RETURS(X,I,RET,BPRNIP)
DIMENSION X(3,1), RET(3,21), BPRNIP(3,21)
COMMON /ZZ/ N,S(3,21),IP,SO,FDP(20,3,21),NTOTN,NTOTM1,IOP
COMMON /KKKKK/ KKK
COMMON /TTT/ S1, NTUBOS
REAL M, NRE, L
GO TO (1,2,3), N
1 CONTINUE
CALL CALSI1(X,I,WO,CO,TO,CF,PT)
CALL MODEVA(WO,CO,TO,CF,PT,TF,A,M,TE,NRE,L,G,DI,WS,BPR)
BPRNIP(N,IP) = BPR
CALL CALPOT(NRE,TE,PT,L,CF,TF,G,DI,POTHPB,POTHPH,CONPOA)
C FUNCION OBJETIVO EN PESOS MEXICANOS.
RET(N,IP) = A*1500.00*0.10 + CONPOA*0.20 + 160000.00
RETURN
2 CONTINUE
CALL CALSI2(X,I,WO,CO,TO,CF,PT)
CALL MODEVA(WO,CO,TO,CF,PT,TF,A,M,TE,NRE,L,G,DI,WS,BPR)
BPRNIP(N,IP) = BPR
CALL CALPOT(NRE,TE,PT,L,CF,TF,G,DI,POTHPB,POTHPH,CONPOA)
C FUNCION OBJETIVO EN PESOS MEXICANOS.
RET(N,IP) = A*1500.00*0.10 + CONPOA*0.20 + 160000.00
RETURN
3 CONTINUE
CALL CALSI3(X,I,WO,CO,TO,CF,PT)
CALL MODEVA(WO,CO,TO,CF,PT,TF,A,M,TE,NRE,L,G,DI,WS,BPR)
BPRNIP(N,IP) = BPR
CALL CALPOT(NRE,TE,PT,L,CF,TF,G,DI,POTHPB,POTHPH,CONPOA)
RET(N,IP) = A*1500.00*0.10 + CONPOA*0.20 + 160000.00 +
1WS*0.008125*8000.00
RETURN
END

```

```

SUBROUTINE CALSI1(X,I,WD,CD,TD,CF,PT)
DIMENSION X(3,1)
COMMON /ZZ/ N, S(3,21), IP, SD, FDP(20,3,21), NTOTN, NTOTH1, IOPT
COMMON /AA/ PTSTAR(65), TWSTAR(65), BTW(56), CSTAR(56), TSTAR(56),
1HPRIMV(65), HFG(65)
COMMON /CC/ BPSOL(64), CSOL(64), TWATER(64)
COMMON /V1V2V3/ V1, V2, V3
LOGICAL FLAG
REAL L1, L2, L3, L4, LAHTW2
L4 = 250000.00
C4 = 0.10
C1 = 0.20
TI = 110.00
L1 = (L4 * C4)/C1
L2 = L1 + V1
C2 = (L1 * C1)/L2
TW2 = S(N,IP)
T2 = FUN2(TW2,C2,56,8,BTW,CSTAR,TSTAR)
T1 = S(N,IP) * X(I,1)
TW1 = FUN2(T1,C1,64,8,BPSOL,CSOL,TWATER)
WD = L2
CD = C2
TD = T2
CF = C1
PT = TLULIN(TWSTAR,PTSTAR,TW1,65,FLAG)
RETURN
END

```



```

SUBROUTINE CALSI2(X,I,WD,CO,TD,CF,PT)
DIMENSION X(3,1)
COMMON /ZZ/ N, S(3,21), IP, SD, FDP(20,3,21), NTOTN, NTOTM1, IDPT
COMMON /AA/ PTSTAR(65), THSTAR(65), BTH(56), CSTAR(56), TSTAR(56),
1HPRIMV(65), HFG(65)
COMMON /CC/ BPSOL (64), CSOL (64), TWATER (64)
COMMON /V1V2V3/ V1, V2, V3
LOGICAL FLAG
REAL L1, L2, L3, L4, LAHTW2
L4 = 250000.00
C4 = 0.10
C1 = 0.20
TI = 110.00
L1 = (L4 * C4)/C1
L3 = L1 + V1 + V2
C3 = (L4 * C4)/L3
TW3 = S(N,IP)
T3 = FUN2(TW3,C3,56,8,BTH,CSTAR,TSTAR)
T2 = S(N,IP) - X(I,1)
L2 = L1 + V1
C2 = (L1 * C1)/L2
TW2 = FUN2(T2,C2,64,8,BPSOL,CSOL,TWATER)
PT = TLULIN (THSTAR,PTSTAR,TW2,65,FLAG)
WD = L3
CO = C3
TD = T3
CF = C2
RETURN
END

```

```
SUBROUTINE CALS13 (X,I,WD,CD,TO,CF,PT)
DIMENSION X(3,1)
COMMON /ZZ/ N, S(3,21), IP, SO, FDP(20,3,21), NTOIN, NTOH1, IOPT
COMMON /AA/ PTSTAR(65), TWSTAR(65), BTH(56), CSTAR(56), TSTAR(56),
1HPRIMV(65), HFG(65)
COMMON /CC/ BPSOL(64), CSOL(64), TWATER(64)
COMMON /V1V2V3/ V1,V2,V3
LOGICAL FLAG
REAL L1, L2, L3, L4, LAMTW2
L4 = 250000.00
C4 = 0.10
C1 = 0.20
TI = 110.00
L3 = L4 - V3
C3 = (L4 * C4)/L3
T3 = S(N,IP) - X(I,1)
TW3 = FUN2(T3,C3,64,8,BPSOL,CSOL,TWATER)
PT = TLULIN(TWSTAR,PTSTAR,TW3,65,FLAG)
WD = L4
CD = C4
TO = TI
CF = C3
RETURN
END
```

```

SUBROUTINE MUDEVACHD,CO,TD,CF,PT,TF,AREA,H,TE,NRE,LANGTU,G,DI,MS,
1BPR)
COMMON /ZZ/ N,S(3,21),IP,SO,FDP(20,3,21),NTOTN,NTOTH1,IOPT
COMMON /AA/ PTSTAR(65), TWSTAR(65), BTH(56), CSTAR(56), TSTAR(56),
1HPRIMV(65), HFG(65)
COMMON /CC/ BPSOL (64), CSOL (64), TWATER (64)
COMMON /TTT/ S1, VTUBOS
LOGICAL FLAG
REAL LAMBDA
REAL H, NRE, LANGTU
TS = S(N,IP)
PS = TLULIN(TWSTAR,PTSTAR,TS,65,FLAG)
WF = (WU + CD)/CF
VV = WD - WF
C WRITE (6,008)
008 FORMAT ( "CALCULO DE LOS GASTOS DE LA SOLUCION DE ENTRADA WU Y DEL
1 VAPOR GENERADO EN EL EVAPORADOR VV")
C WRITE (6,009) WF,VV
009 FORMAT ("0" / " WF = ",F12.4,10X,"VV = ",F12.4)
C PT ES LA PRESION DE TRABAJO A LA CUAL SE VA A TRABAJAR Y TW ES LA
C TEMPERATURA DE EBULLICION DEL AGUA PURA CORRESPONDIENTE A ESA PRE--
C SION DE TRABAJO
C WRITE(6,200)
C USO DE LA FUNCION TLULIN PARA OBTENER LA TEMPERATURA DE EBULICION
C DEL AGUA POR RASTRO EN TABLA E INTERPOLACION LINEAL.
C TW = TLULIN (PTSTAR,TWSTAR,PT,65,FLAG )
C IMPRESION DE LOS RESULTADOS
C WRITE (6,203) PT,TW
C IF (FLAG) WRITE (6,204)
200 FORMAT ("TEMPERATURA DE EBULLICION DEL AGUA PURA POR RASTRO EN
1TABLA E INTERPOLACION LINEAL")
203 FORMAT ( "0" / "LOS RESULTADOS DADOS POR TLULIN SON" / "0",15X,"
1PT = ",F9.4,10X,"TW = ",F10.3)
204 FORMAT (55X,"(EXTRAPOLADO)")
C DETERMINACION DE LA TEMPERATURA DE EBULLICION DE LA SOLUCION COMO
C UNA FUNCION DE SU CONCENTRACION Y DE LA TEMPERATURA DE EBULLICION
C DEL AGUA PURA A LA MISMA PRESION DE TRABAJO.
C BTH,CSTAR Y TSTAR SON VECTORES QUE REPRESENTAN LA TEMPERATURA DE---
C EBULLICION DEL AGUA PURA, LA CONCENTRACION DE LA SOLUCION Y LA TEM-
C PERATURA DE EBULLICION DE LA SOLUCION RESPECTIVAMENTE,LOS VALORES--
C DE BTH NO COINCIDEN CON LOS DE TWSTAR PORQUE SE FORMARON ESTOS ARR E
C GLOS CON OBJETIVOS DIFERENTES.
C TF = FUN2 (TW,CF,56,8,BTH,CSTAR,TSTAR)
C IMPRESION DE LOS RESULTADOS
C WRITE (6,500)
C WRITE (6,503) TW,CF,TF
500 FORMAT ("0 TEMPERATURA DE EBULLICION DE LA SOLUCION PUR RASTRO EN
1TABLA E INTERPOLACION DOBLE")
503 FORMAT("0" / " LOS RESULTADOS DADOS POR FUN2 SON"/ "0",5X,"PARA TW
1 = ",F10.3,3X," Y CF = ",F7.4,5X,"TF = ",F10.3)
C DETERMINACION DE LAS ENTALPIAS DEL VAPOR GENERADO EN EL EVAPORADOR-
C CORRESPONDIENTES A LA TEMPERATURA DE SATURACION DEL MISMO.
C TWSTAR Y HPRIMV SON VECTORES QUE REPRESENTAN LA TEMPERATURA DE EBU-
C LLICION DEL AGUA PURA Y LA ENTALPIA DEL VAPOR DE AGUA SATURADO A LA
C MISMA TEMPERATURA
C HG = TLULIN (TWSTAR,HPRIMV,TW,65,FLAG)

```

```

C IMPRESION DE LOS RESULTADOS
C WRITE (6,400)
C WRITE (6,403) TH, HG
C IF (FLAG) WRITE (6,404)
400 FORMAT ("ENTALPIA DEL VAPOR DE AGUA SATURADO POR RASTREO EN TABLA
1E INTERPOLACION LINEAL")
403 FORMAT ("0" / "LOS RESULTADOS DADOS POR TLULIN SON" / "0", 10X, "
1TH = ", F10.3, 10X, "HG = ", F10.4)
404 FORMAT ("55X", "EXTRAPOLADO")
BPR = TF - TH
C DETERMINACION DE LA ENTALPIA DEL VAPOR DE AGUA GENERADO EN EL EVAPO
C RADOR ( VAPOR SOBRECALENTADO ) CORRESPONDIENTE A LA TEMPERATURA DE
C EBULLICION DE LA SOLUCION.
HV = HG + 0.46*(TF - TH)
C WRITE (6,600) HV
600 FORMAT (" LA ENTALPIA DEL VAPOR SOBRECALENTADO ES HV = ", F10.4)
C DETERMINACION DE LAS ENTALPIAS DE LAS SOLUCIONES QUE ESTAN ENTRANDO
C Y SALIENDO DEL EVAPORADOR.
C CALCULO DE LA ENTALPIA DE LA SOLUCION DILUIDA EN FUNCION DE SU TEM
C PERATURA Y CONCENTRACION.
CALL ENTAL (TO, CO, HO)
C WRITE (6,800)
C WRITE (6,803) TO, CO, HO
800 FORMAT (" ENTALPIA DE LA SOLUCION DILUIDA")
803 FORMAT ("0" / " LOS RESULTADOS DADOS POR ENTAL SON"/ "0", 5X, "PARA
1 TO = ", F8.3, 5X, " Y CO = ", F7.4, 5X, "HO = ", F12.6)
C CALCULO DE LA ENTALPIA DE LA SOLUCION CONCENTRADA EN FUNCION DE SU
C TEMPERATURA Y CONCENTRACION.
CALL ENTAL (TF, CF, HF)
C WRITE (6,900)
C WRITE (6,903) TF, CF, HF
900 FORMAT (" ENTALPIA DE LA SOLUCION CONCENTRADA")
903 FORMAT ("0"/ " LOS RESULTADOS DADOS POR ENTAL SON"/ "0", 5X, "PARA
1 TF = ", F10.3, 3X, "Y CF = ", F7.4, 5X, " HF = ", F12.6)
C DETERMINACION DEL CALOR LATENTE DE EVAPORACION DEL AGUA POR RASTREO
C EN TABLA E INTERPOLACION LINEAL, EN FUNCION DE LA PRESION DEL VAPOR
C DE CALENTAMIENTO.
LAMBDA = TLULIN (PTSTAR, HFG, PS, 65, FLAG)
C WRITE (6,700)
C WRITE (6,703) PS, LAMBDA
700 FORMAT (" CALOR LATENTE DE EVAPORACION DEL AGUA POR RASTREO EN
1TABLA E INTERPOLACION LINEAL")
703 FORMAT (" LOS RESULTADOS OBTENIDOS DE LA INTERPOLACION SON"/"0",
15X, "PARA PS = ", F8.4, 5X, "LAMBDA = ", F10.3)
C IF (FLAG) WRITE (6,704)
704 FORMAT ("EXTRAPOLADO")
C CALCULO DE LA CANTIDAD DE CALOR DESARROLLADA EN EL EVAPORADOR.
Q = WF * HF + VV * HV - WD * HD
C WRITE (6,35) Q
35 FORMAT (" LA CANTIDAD DE CALOR QUE NECESITA SUMINISTRARSE AL EVA
1PORADOR ES"/"0", 10X, "Q = ", E12.3, "BTU/HR")
C WS = VAPOR DE CALENTAMIENTO NECESARIO PARA SUMINISTRAR Q
WS = Q/LAMBDA
C WRITE (6,45) WS
45 FORMAT (" VAPOR REQUERIDO PARA EL CALENTAMIENTO"/"0", 10X, "WS = ",
1F12.3, "LBS/HR")
C SE TRABAJARA CON UN APPROACH DE 5 GRADOS FARENHEIT EN EL CANBIADOR
C DE CALOR.
TE = TS = 5.00
TAV = (TF + TE)/2

```

```

CALL CALCCP (TAV,CF,CPAV)
M = MF + Q/(CPAV*(TE - TF))
G = MASA VELOCIDAD EN LOS TUBOS DEL CAMBIADOR DE CALOR.
CALL CALDEN (TAV, CF, RHOAV)
QVOL = FLUJO A TRAVES DE LA BOMBA EN GALONES POR MINUTO.
QVOL = (7.48*M)/(60*RHOAV)
C WRITE (6,553) QVOL
553 FORMAT (2X,"QVOL = ",E12,6,"GPH.")
C VELOCIDAD EN FT/HR.
VEL = 36000.00
G = VEL * RHOAV
C S = AREA PARA EL FLUJO DEL FLUIDO EN LOS TUBOS.
S1 = M/G
C CALCULO DEL AREA PARA LA TRANSFERENCIA DE CALOR.
C SE USARAN TUBOS DE NIKEL 3/4 DE PULGADA NOMINAL IPS ESTANDARD
DI = 0.824 /12
DO = 1.05/12
KM = 34
: SI = AREA PARA EL FLUJO DE FLUIDOS DE UN TUBO.
: AUEXT = AREA EXTERNA PARA LA TRANSFERENCIA DE CALOR POR PIE LINEAL
SI = 0.534/144
AUEXT = 0.275
RDI = 0.000
RDO = 0.000
HD = 1500.00
CALL UD (TAV, CF, VEL, DI, DO, HI, RDI, KM, RDO, HD, U,NRE)
: WRITE (6,998) U
998 FORMAT (" U = ",E12,6,"BTU/HR.SQ.FT.F.DEG.")
: DETERMINACION DE LA TEMPERATURA MEDIA LOGARITMICA.
DTML = ((TS - TE) * (TS - TF))/(CALOG((TS - TE)/(TS - TF)))
CALCULO DEL AREA PARA LA TRANSFERENCIA DE CALOR.
AREA = Q/(U * DTML)
WRITE (6,888) AREA
888 FORMAT (" EL AREA PARA LA TRANSFERENCIA DE CALOR ES " / " ",10X,"
1AREA = ",E12,6,2X,"SQ.FT.")
NTUBOS = NUMERO DE TUBOS EN EL CAMBIADOR DE CALOR.
LONGTU = LONGITUD DE LOS TUBOS.
DE TUBO.
NTUBOS = S1/SI
LONGTU = AREA/(NTUBOS * AUEXT)
RETURN
END

```

FORM

```

C      FUNCION TLULIN (X,F,XVAL,N,FLAG)
C
C      FUNCION QUE EFECTUA EL RASTREO EN LA TABLA Y LA INTERPOLACION LINEAL EN LOS PARES DE VALORES TABULADOS X(1),...,X(N) Y F(1),...,F(N).
C      DADO X = XVAL EL CORRESPONDIENTE FXVAL ES RETORNADO.
C      LOGICAL FLAG
C      DIMENSION X(1), F(1)
C
C      FLAG = .FALSE.
C      PONER A FLAG COMO CIERTA SI LA EXTRAPOLACION ES NECESARIA.
C      IF ( XVAL.LT.X(1) .OR. XVAL.GT.X(N) ) FLAG = .TRUE.
C
C      EFECTUAR UN RASTREO PARA VER SI XVAL CUMPLE
C      NM1 = N - 1
C      DO 1 I=1,NM1
C      IF (XVAL.LT.X(I+1) .OR. I.EQ.NM1 ) GO TO 2
C      1 CONTINUE
C
C      EFECTUAR LA INTERPOLACION LINEAL CON LOS PUNTOS BASE I E I+1
C      2 TLULIN = F(I) + (XVAL - X(I))*(F(I+1)-F(I))/(X(I+1) - X(I))
C      RETURN
C      END

```



APLICACION DE UNA SUBROUTINA ARBITRARIA DE Y EN FUNCION DE X AND Z

```

FUNCTION FUN2 (A,B,N,M,X,Z,Y)
DIMENSION X(2), Z(2), Y(2)
IF ((A .LE. X(1)) .AND. (B .LE. Z(1))) GO TO 13
IF ((A .GE. X(N)) .AND. (B .GE. Z(H))) GO TO 14
IF ((A .LE. X(1)) .AND. (B .GE. Z(M))) GO TO 15
IF ((A .GE. X(N)) .AND. (B .LE. Z(N=H+1))) GO TO 16
IF (A .LE. X(1)) GO TO 19
IF (A .GE. X(N)) GO TO 23
MP = M + 1
GO TO 17
23 I = (N=H+1)
I2 = N
GO TO 22
19 I = 1
I2 = I + M - 1
22 DO 20 J = I,I2
20 IF (B .LE. Z(J)) GO TO 21
21 J = J - 1
FUN2 = Y(J) + (B-Z(J))/(Z(J+1)-Z(J)) * (Y(J+1)-Y(J))
RETURN
13 FUN2 = Y(1)
RETURN
14 FUN2 = Y(N)
RETURN
15 FUN2 = Y(M)
RETURN
16 FUN2 = Y(N=H+1)
RETURN
17 DO 3 I=MP,N,M
IF (A .LT. X(I)) GO TO 4
3 CONTINUE
4 IF (B .LT. Z(I=H)) GO TO 9
IF (B .GT. Z(I=1)) GO TO 11
GO TO 12
9 YT1 = Y(I=H)
YT2 = Y(I)
GO TO 18
11 YT1 = Y(I=1)
YT2 = Y(I+M=1)
GO TO 18
12 J1 = I=H
J2 = I=1
DO 5 J=J1,J2
IF (B .LT. Z(J)) GO TO 6
5 CONTINUE
6 YT1 = Y(J=1) + (B-Z(J=1))/(Z(J)-Z(J=1)) * (Y(J)-Y(J=1))
I2 = I+M=1
DO 7 J=I,I2
IF (B .LT. Z(J)) GO TO 8
7 CONTINUE
8 YT2 = Y(J=1) + (B-Z(J=1))/(Z(J)-Z(J=1)) * (Y(J)-Y(J=1))
18 FUN2 = YT1 + (A-X(I=1))/(X(I)-X(I=1)) * (YT2 -YT1)
RETURN
END

```

C

SUBROUTINE ENTAL (X1,X2,Y)

Y = (-6.40874852E+01) + (1.21317199E+00) \* (X1) +  
1(2.11861379E+02) \* (X2) - (2.68619112E+00) \* (X1) \* (X2) +  
2(4.07586535E+00) \* (X1) \* (X2\*\*2)  
RETURN  
END



```
SUBROUTINE CALCCP (T, C, CCAL)
CCAL = (9.49188779E-01) + (2.01900577E-05) * (T) -
1(3.28659389E-01) * (C)
RETURN
END
```

```

SUBROUTINE UOCTAV, CF, VEL, DI, DO, HI, RDI, KH, RDO, HO, U, NRE)
REAL NRE
CALL CALHI (TAV, CF, VEL, DI, HI, NRE)
WRITE (6,990) HI
C 990 FORMAT (10X,"HI = ",E12.6)
      XW = DO - DI
      DL = (DO - DI)/ALOG(DO/DI)
      UO1 = (DO)/(HI * DI) + (DO * RDI)/(DI) + (DO) * (XW)/(KH * DL)
      1+RDO + 1/HO
      U = 1/UO1
      RETURN
      END

```

70  
FOPM

yo

```
SUBROUTINE CALHI (TAV, CF, VEL, DI, HI, NRE)
REAL MU, K, NRE, NPR
CALL CALDEN (TAV, CF, RHO)
WRITE (6,885) RHO
C
885 FORMAT (" RHO = ",E12.6)
CALL CALVIS (TAV, CF, MU)
WRITE (6,886) MU
C
886 FORMAT (" MU = ",E12.6)
NRE = DI * VEL * RHO/MU
WRITE (6,950) NRE
C
950 FORMAT (" NUMERO DE REYNOLDS = ",F12.0)
CALL CALCCP (TAV, CF, CP)
WRITE (6,887)CP
C
887 FORMAT (" CP = ",E12.6)
CALL CALCK (TAV, CF, K)
WRITE (6,889) K
C
889 FORMAT (" K = ",E12.6)
NPR = CP * MU/K
WRITE (6,951) NPR
C
951 FORMAT (" NUMERO DE PRANDTL = ",F9.4)
HI = ((K)/(DI)) * (NRE**0.8) * (NPR**0.4) * (0.0255)
RETURN
END
```

FORM:

SUBROUTINE CALDEN (T, C, DEN)

C

DEN = 62.3 \* ((9.67806874E-01) + (1.46770847E-04)\*(T) +  
1(1.22827104E+00) \* (C) - (1.45233716E-03) \* (T) \* (C))  
RETURN  
END

```
SUBROUTINE CALVIS (T, C, VIS)
VISC = EXP((3.18176205E-01) - (7.56133555E-03) * (T) +
1(1.11472513E+01) * (C) - (3.19721153E-02) * (T) * (C))
VIS = 2.42 * VISC
RETURN
END
```

```
SUBROUTINE CALCK (T, C, CONTER)
CONTER = (0.32693236) + (3.54145430E-04) * (T) +
1(1.42345771E-01) * (C) - (1.55951488E-01) * (C**2) -
2(1.04770982E-04) *(T) *(C)
RETURN
END
```

```

SUBROUTINE CALPOT (NRE,TE,P1,L,CF,TF,G,DI,POTHPB,POTHPM,CONPOA)
COMMON /AA/ PTSTAR(65), TWSTAR(65), BTW(56), CSTAR(56), TSTAR(56),
IHPPIMV(65), HFG(65)
COMMON /TTT/ S1, NTUBOS
LOGICAL FLAG
REAL NRE, L
C
C NRE = NUMERO DE REYNOLDS EN LOS TUBOS DEL CAMBIADOR DE CALOR.
C P2 = PRESION QUE DEBE HABER EN LA LINEA PARA QUE LA SOLUCION NO
C HIERVA.
C SE CONSIDERARA QUE LA PRESION DE VAPOR DE LA SOLUCION ES IGUAL A
C LA PRESION DE VAPOR DEL AGUA PURA A LA MISMA TEMPERATURA AUNQUE EN
C REALIDAD ES MENOR.
C P21 = TLULIN(TWSTAR, PTSTAR, TE, 65, FLAG)
C P2 = P21 + 0.10 * P21
C DE LA GEOMETRIA DEL SISTEMA PUEDE SUPONERSE QUE Z2 = Z1, ES DECIR
C QUE EL CAMBIO DE NIVEL ENTRE LA SUCCION Y LA DESCARGA DE LA BOMBA
C ES MINIMO Y QUE POR LO TANTO EL CAMBIO DE ENERGIA POTENCIAL ES
C DESPRECIABLE.
C IGUALMENTE SE SUPONDRÁ QUE U2 = U1 Y QUE EL CAMBIO DE ENERGIA
C CINETICA ES TAMBIEN DESPRECIABLE.
C DHFGCC = PERDIDAS POR FRICCION (FT.) EN LOS TUBOS DEL CAMBIADOR DE
C CALOR SEGUN FORMULA 3.45 DEL KERN.
C AGR = ACELERACION DEBIADA A LA FUERZA DE GRAVEDAD.
C AGR = 418000000.00
C F = 0.0035 + 0.264/(NRE**.42)
C CALL CALDEN (TE,CF,DENTE)
C CALL CALDEN (TF,CF,DENTF)
C DEN = (DENTE + DENTF)/2
C DE = DIAMETRO EQUIVALENTE.
C RH = RADIO HIDRAULICO.
C S1 = AREA DE FLUJO DEL FLUIDO PARA TODOS LOS TUBOS.
C PERMOJ = PERIMETRO DE HOJADO
C PERMOJ = 3.1416 * NTUBOS * DI
C RH = S1/PERMOJ
C DE = 4 * RH
C DHFGCC = 4 * F*(G**2)*L/(2*AGR*(DEN**2)*DE)
C W = CABEZA DE LA BOMBA EN FT.
C W = (P2/DENTE) - (P1/DENTF) + DHFGCC
C WRITE (6,557) W
557 FORMAT(2X,"W = ",E12.6,"FT.")
C POTENB = POTENCIA TEORICA DE LA BOMBA LB.FUERZA*FT./LB.MASA
C POTENB = W * DEN
C EFBOMB = EFICIENCIA DE LA BOMBA.
C EFBOMB = 0.40
C POTREA = POTENCIA REAL DE LA BOMBA LB.FUERZA * FT./LB.MASA
C POTREA = POTENB/EFBOMB
C POTHPB = POTENCIA REAL DE LA BOMBA EN HP.
C POTHPM = POTREA/550.00
C EFMOTO = EFICIENCIA DEL MOTOR.
C EFMOTO = 0.65
C POTHPM = POTENCIA EN HP MOTOR.
C POTHPM = POTHPB/EFMOTO
C EL SISTEMA TRABAJARA ANUALMENTE 8000 HRS.
C POTMKW = CONSUMO DE POTENCIA EN KILOWATTS DEL SISTEMA.
C POTMKW = POTHPM * 0.745
C CONPOA = CONSUMO ANUAL DE POTENCIA DEL SISTEMA.
C CONPOA = POTMKW * 8000.00
RETURN
END

```

B I B L I O G R A F I A .

- 1.- FORMULATION AND OPTIMIZATION OF MATHEMATICAL MODELS.  
Cecil L. Smith and Ralph W. Pike  
INTERNATIONAL TEXTBOOK COMPANY
- 2.- INTRODUCTION TO DYNAMIC PROGRAMMING.  
George L. Nemhauser  
JOHN WILEY AND SONS, INC.
- 3.- FUNDAMENTAL OF OPERATION RESEARCH.  
Russell L. Ackoff and W. Sasieni.  
JOHN WILKY AND SONS, INC.
- 4.- MODELING AND SIMULATION IN CHEMICAL ENGINEERING.  
Roger G. E. Franks.  
JOHN WILEY INTERNATIONAL EDITION.
- 5.- STRATEGY OF PROCESS ENGINEERING.  
Dale F. Rudd and Charles C. Watson  
JOHN WILEY INTERNATIONAL EDITION.
- 6.- THEORY AND TECHNIQUES OF OPTIMIZATION FOR PRACTICING.  
ENGINEERS  
Raymond L. Zahradnig.  
BARNES AND NOBLE.
- 7.- OPTIMIZATION: THEORY AND PRACTICE.  
Gordon S. G. Beveridge and Robert S. Schechter.  
MC GRAW HILL SERIES IN CHEMICAL ENGINEERING.
- 8.- ITAHARA SEIJI, STIEL, L.I. "OPTIMAL DESIGN OF MULTIPLE.  
EFFECT EVAPORATORS BY DYNAMIC PROGRAMMING",  
IND. ENG. CHEM. PROCESS DESIGN DEVELOP. 5 309 (1966).



- 9).- ITAHARA SEIJI, STIEL, L.I. "OPTIMAL DESIGN OF MULTIPLE EFFECT EVAPORATORS WITH VAPOR BLEED STREAMS",  
IND. ENG. CHEM. PROCESS DESIGN DEVELOP. 7. 11 (1968).
- 10).- BOX, M.J. " A NEW METHOD OF CONSTRAINED OPTIMIZATION AND A COMPARISON WITH OTHER METHODS", COMPUTER J.,  
8 , 42-52, 1965.
- 11).- OPTIMIZATION TECHNIQUES WITH FORTRAN  
James L. Kuester and Joe H. MIZE  
MC GRAW HILL BOOK COMPANY.
- 12).- DIGITAL COMPUTING AND NUMERICAL METHODS,  
Brice Carnahan and James O. Wilkes  
JOHN WILEY AND SONS INC.
- 13).- INTRODUCTION TO CHEMICAL ENGINEERING  
Walter L. Badger and Julius T. Banchemo.  
MC GRAW HILL INTERNATIONAL STUDENT EDITIONS
- 14).- PROCESS HEAT TRANSFER  
D. Q. Kern  
MC GRAW HILL INTERNATIONAL STUDENT EDITIONS.
- 15).- CHEMICAL ENGINEER'S HANDBOOK, 4 th. Ed.  
John H. Perry  
MC GRAW HILL INTERNATIONAL STUDENT EDITIONS.