

**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA**

CURSO ABIERTO

TOPICOS DE ACTUALIDAD DE LA INVESTIGACION

DE OPERACIONES Y LA SIMULACION

DIAS 23, 24 y 25 DE SEPTIEMBRE

**(EN COLABORACION CON EL INSTITUTO MEXICANO DE SISTEMAS
E INVESTIGACIONES DE OPERACIONES A.C.)**

**PROGRAMACION ESTOCASTICA
EN OPTIMIZACION**

**PROFESOR
DR. SERGIO DE LOS COBOS
PALACIO DE MINERIA**

1 9 9 6

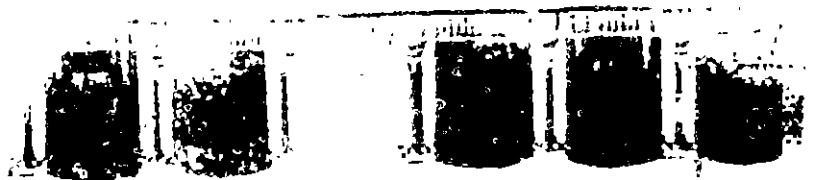
1088

TRAVIO DE VIMBIV
FONDE FOR CORCE
SCORPUS

REK

1110

51



PROGRAMACION ESTOCASTICA EN OPTIMIZACION.

Sergio G. de los Cobos Silva

Profesor del Departamento de Matemáticas, UAM-Iztapalapa.

Blanca R. Pérez Salvador

Profesor del Departamento de Matemáticas, UAM-Iztapalapa.

Miguel A. Gutiérrez Andrade

Profesor del Departamento de Sistemas, UAM-Azcapotzalco.

Contenido.

- 0. Introducción. 1
- 1. Programación combinatoria. 5
 - 1.1. Planteamiento del problema combinatorio. 5
 - 1.2. Problemas P y NP. 7
 - 1.3. Optimización, satisfacibilidad y semioptimización. 9
- 2. Búsqueda tabú. 13
 - 2.1. Introducción a la búsqueda tabú. 13
 - 2.2. Ejemplo: problema de calendarización. 16
 - 2.3. Memoria de término intermedio y largo. 30
 - 2.4. Intensificación y diversificación regional. 31
 - 2.5. Criterios de aspiración. 36
 - 2.6. Problema del agente viajero. 38
 - 2.6.1. Disposición de desechos sólidos. 39
 - 2.6.1.1. Planteamiento del problema. 39
 - 2.6.1.2. Objetivo. 39
 - 2.6.1.3. Solución y resultados. 42
 - 2.6.1.4. Conclusiones del ejemplo. 42
- 3. Recocido Simulado. 45
 - 3.1. Problema de localización de plantas. 46
 - 3.2. Algoritmo exacto. 47
 - 3.2.1. Descripción del algoritmo. 47
 - 3.2.2. Experiencia computacional. 50
 - 3.3. Algoritmo heurístico. 51
 - 3.3.1. Descripción del algoritmo. 51
 - 3.3.2. Experiencia computacional. 52
 - 3.4. Recocido simulado. 53
 - 3.4.1. Proceso de recocido de un sólido. 53
 - 3.4.2. Algoritmo de recocido simulado. 56
 - 3.4.3. Aspectos generales del algoritmo. 59
 - 3.4.4. Descripción del algoritmo. 60
 - 3.4.5. Experiencia computacional. 62
 - 3.5. Algoritmo de búsqueda local. 62
 - 3.5.1. Experiencia computacional. 63
 - 3.6. Análisis compartivo de los métodos. 64
- 4. Algoritmos genéticos. 67

4.1. Algunos experimentos AG.	68
4.2. Componentes básicos.	69
5. Análisis de objetivos.	75
6. Bondades y alcances de la programación estocástica.	79
6.1. Inventarios multiproducto.	79
6.1.1. Inventarios multiproducto: caso determinístico.	80
6.1.2. Marco de mejoramiento por Búsqueda Tabú.	83
6.1.3. Manejo dinámico de la lista tabú.	84
6.1.4. Resultados computacionales.	85
Referencias.	89
Glosario.	93

0.1 Introducción.

En las últimas décadas se ha presentado una variedad importante de problemas de optimización combinatoria de gran escala. Las aplicaciones van desde la localización de centros de emergencia hasta los más sofisticados diseños de modelos industriales, financieros y militares. dichos problemas presentan gran complejidad en la determinación de soluciones óptimas o cercanas a la solución óptima.

Hoy en día, el reto para varias disciplinas es proponer técnicas eficientes de solución, que por un lado garanticen soluciones "buenas", y por otro lado que sean rápidos y fáciles de implantar.

En situaciones prácticas, uno de los problemas computacionales es la *explosión combinatoria*. La explosión combinatoria se encuentra en aquellas situaciones donde las elecciones están compuestas secuencialmente, permitiendo una vasta cantidad de posibilidades, por ejemplo, en la elección de rutas alternativas en un laberinto. Situaciones de este tipo ocurren en problemas de inversiones financieras, operaciones de manufactura, manejo de inventarios, localización de recursos, presupuestación de capitales, etc. Algunos otros ejemplos incluyen la explotación de minerales, análisis de datos meteorológicos, diseño de circuitos integrados, manejo de recursos hidráulicos, operaciones de monitoreo de satélites, y mantenimiento de sistemas.

Los intentos por tratar con el problema de la explosión combinatoria han encontrado muchos obstáculos. No es suficiente contar con un "conocimiento experto" para manejarlos de manera efectiva, de igual manera no es suficiente confiar en el poder computacional de alta velocidad de las super computadoras. Algunos problemas clásicos donde la explosión combinatoria prevalece, muestran que un intento por generar todas las alternativas relevantes por computadora no es una tarea factible. Así por ejemplo, en el problema del agente viajero, en el cual se tiene que salir de una ciudad y regresar a la misma después de haber visitado (con costo de viaje mínimo) todas las demás ciudades, si se tienen n ciudades en total que recorrer entonces dada una ciudad inicial existen $(n-1)!$ soluciones factibles, y si una computadora

que pudiera ser programada para examinar soluciones a razón de un billón de soluciones por segundo; la computadora terminaría su tarea para $n = 25$ ciudades (que es un problema pequeño para muchos casos prácticos) en alrededor de 19,674 años.

Aunque para algunos problemas de optimización como son los problemas de programación lineal, los problemas del transporte, los problemas de asignación, existen algoritmos rápidos y eficientes¹ para muchos otros problemas de optimización combinatoria no los hay, como son los casos del problema de asignación cuadrático, el problema de localización de plantas, el problema del agente viajero, etc. Así, desde hace varios años éstos últimos problemas han sido atacados por algoritmos desarrollados especialmente para el problema específico y usando una diversidad de técnicas tales como planos de corte, ramificación y acotamiento, enumeración implícita, relajación Lagrangeana, partición de Benders, etc. o por combinaciones de las técnicas antes mencionadas. Sin embargo no pueden resolverse de manera exacta usando tiempo y espacio de computadora razonables, aún cuando se tenga sólo un número moderado de variables. En la actualidad existe un gran esfuerzo por parte de la comunidad investigadora para el diseño de buenas heurísticas, i.e., algoritmos eficientes con respecto al tiempo de cómputo y al espacio de memoria, y con cierta verosimilitud de entregar una solución "buena" i.e., relativamente cercana a la óptima mediante el examinar sólo un pequeño subconjunto del número total de permutaciones posibles.

El principal problema de algunos algoritmos heurísticos es su inhabilidad para enfrentarse con puntos de optimalidad local, o en otras palabras, su inhabilidad de continuar la búsqueda hacia el óptimo global después de que un óptimo local se ha alcanzado.

Lo anterior ha propiciado que el enfoque de la inteligencia artificial

¹Para instancias grandes, no es posible encontrar solución óptima en tiempos de cómputo razonables, incluso, por problemas de programación lineal, se han encontrado instancias "patológicas" donde se observa que el método simplex no tiene la propiedad de ser un algoritmo de tiempo polinomial a diferencia del método de Karmarkar.

haya revivido como solución de problemas que requieren de la búsqueda heurística. intrínsecamente en la base del conocimiento de las operaciones, especialmente para mecanismos de razonamientos analógicos y lógicos.

Recientemente varias aproximaciones han surgido del manejo de problemas de decisión complejos, como son: *algoritmos genéticos, redes neuronales, recocido simulado, búsqueda tabú, análisis de objetivos y búsqueda dispersa* entre otros.

La búsqueda tabú, junto con la técnica del recocido simulado y los algoritmos genéticos, han sido singularmente calificados por el Committee on Next Decade of Operations Research (Condor(1988)) como "extremadamente promisorios" para el tratamiento futuro de aplicaciones prácticas.

El propósito de este trabajo es presentar las generalidades de algunos de los métodos de programación estocástica para problemas de optimización combinatoria denominados también como *métodos heurísticos*.²

Una característica importante de los métodos presentados es que proporcionan marcos generales de solución a una amplia variedad de problemas, no sólo de tipo combinatorio.

²La palabra heurística proviene de la palabra griega *heuriskein* que significa encontrar o descubrir.

Capítulo 1

Programación combinatoria.

Los problemas de optimización se dividen naturalmente en dos categorías: problemas de optimización con variables continuas y problemas de optimización con variables discretas. A estos últimos se les llama problemas de *optimización combinatoria*. En problemas de tipo continuo se busca la solución sobre un conjunto con ciertas propiedades de continuidad y generalmente de convexidad; en los problemas de optimización combinatoria se busca la solución sobre un conjunto finito o infinito numerable. Estas dos clases de problemas tienen diferentes retos y los métodos para resolverlos son distintos.

1.1 Planteamiento del problema combinatorio.

El problema de optimización combinatoria tiene la siguiente forma:

$$(P) \text{ Minimizar } C(x) : x \in X \subset R^n.$$

La función objetivo puede ser lineal o no lineal, y la condición $x \in X$ se asume restringida a componentes específicos de x con valores discretos, las restricciones pueden ser incluso de tipo lógico.

Definición 1.1 *Se define un movimiento s como un mapeo definido sobre un subconjunto $X(s)$ de X de la siguiente forma:*

$$s : X(s) \rightarrow X.$$

Es decir, se considera que un *movimiento* es una transición de una solución factible a otra solución factible (transformada), el cual se puede describir mediante un conjunto de uno o varios *atributos*. Este mecanismo lo utilizan diferentes métodos, en general es utilizado por los métodos de optimización clásica como son los de programación lineal, programación entera, programación no lineal, etc.

Una **instancia** de un problema de optimización combinatoria puede formalizarse como una pareja (\mathcal{S}, f) , donde \mathcal{S} denota el conjunto finito de todas las soluciones posibles y f la **función de costo**, mapeo definido por

$$f : \mathcal{S} \rightarrow R \quad (1.1)$$

En el caso de minimización, el problema es encontrar $i_{opt} \in \mathcal{S}$ que satisfaga

$$f(i_{opt}) \leq f(i) \quad \forall i \in \mathcal{S} \quad (1.2)$$

en el caso de maximización, la i_{opt} que satisfaga

$$f(i_{opt}) \geq f(i) \quad \forall i \in \mathcal{S} \quad (1.3)$$

A la solución i_{opt} se le llama una **solución globalmente óptima** y $f_{opt} = f(i_{opt})$ denota el costo óptimo, mientras que \mathcal{S}_{opt} denota el conjunto de soluciones óptimas.

Un problema de **optimización combinatoria** es un conjunto I de instancias de un problema de optimización combinatoria.

Definición 1.2 Sea (\mathcal{S}, f) una instancia de un problema de optimización combinatoria. Entonces una **estructura de vecindades** es un mapeo

$$\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}, \quad (1.4)$$

que define para cada solución $i \in \mathcal{S}$ un conjunto $\mathcal{S}_i \subset \mathcal{S}$ de soluciones que son "cercañas" a i en algún sentido. El conjunto \mathcal{S}_i se llama **vecindad** de la solución i y cada $j \in \mathcal{S}_i$ se llama un **vecino** de i . Además se supone que $j \in \mathcal{S}_i \Leftrightarrow i \in \mathcal{S}_j$.

Definición 1.3 Sea (S, f) una instancia de un problema de optimización combinatoria y \mathcal{N} una estructura de vecindades. Entonces un **mecanismo de generación** es un medio para seleccionar una solución j de la vecindad S_i de la solución i

Definición 1.4 Sea (S, f) denote una instancia de un problema de optimización combinatoria y sea \mathcal{N} una estructura de vecindades, entonces $i \in S$ se llama una **solución óptima local** o simplemente un **óptimo local** con respecto a \mathcal{N} si i es mejor que o igual a, todas sus soluciones vecinas con respecto al costo. Específicamente, en el caso de minimización, i se llama **solución mínima local** o simplemente un **mínimo local** si

$$f(i) \leq f(j), \forall j \in S_i, \quad (1.5)$$

y en el caso de maximización, i se llama una **solución máxima local** o simplemente un **máximo local** si

$$f(i) \geq f(j), \forall j \in S_i, \quad (1.6)$$

Definición 1.5 Sea (S, f) denote una instancia de un problema de optimización combinatoria y sea \mathcal{N} una estructura de vecindades. Entonces \mathcal{N} se llama **exacta** si para cada $i \in S$ que es localmente óptimo con respecto a \mathcal{N} , i también es globalmente óptimo.

Por lo que un problema de *optimización combinatoria* es un conjunto digamos I de instancias de un problema de optimización combinatoria, de manera informal, una instancia está dada por los "datos de entrada" y la información suficiente para obtener una solución mientras que un problema es una colección de instancias del mismo tipo.

Existe un amplio rango de procedimientos heurísticos y optimales para resolver problemas que pueden escribirse en la forma (P). Dichos procedimientos se pueden caracterizar a través de sucesiones de movimientos, los cuales permiten pasar de un punto a otro.

1.2 Problemas P y NP.

Un aspecto importante de los problemas de optimización en general, es el *esfuerzo computacional*. Por ejemplo, el esfuerzo computacional para

resolver el problema de asignación por medio del método Húngaro está en relación polinomial respecto al crecimiento del tamaño del problema. en cambio, para el problema del agente viajero. el esfuerzo computacional está en relación exponencial.

Aquellos problemas para los que se conoce un algoritmo con esfuerzo computacional de tipo polinomial, se dice que pertenecen a la clase P , en cambio, aquellos problemas de esfuerzo computacional no polinomial. por ejemplo de tipo exponencial, se dice que están en la clase NP (non-deterministic polynomial).

Un concepto importante es el de *reducibilidad o transformabilidad*, para el cual, suponga que se tiene un problema p_1 con un algoritmo de solución conocido digamos a . Si se puede transformar cada instancia de otro problema p_2 dentro de una instancia de p_1 en tiempo polinomial, entonces se puede utilizar el algoritmo a para resolver a este último problema. Entonces podemos decir que p_1 es al menos tan *difícil* de resolver que p_2 .

Si cada problema NP es polinomialmente reducible a un problema p , entonces se dice que el problema p es *NP-difícil* (NP-hard). Si además, el problema p es un problema NP , entonces se dice que p es un problema *NP-completo* (NP-complete). Esto es muy importante puesto que nos lleva a un problema abierto en la actualidad: ¿ $P=NP$?, puesto que, si se puede encontrar un algoritmo polinomial para un problema NP -completo, entonces se habrá encontrado un algoritmo polinomial para todos los problemas NP .

Es muy importante el conocimiento de las propiedades y estructura matemática del problema en el estudio de la existencia de algoritmos que permitan encontrar la solución buscada en un tiempo polinomial y la construcción de ellos cuando es posible. En particular, la teoría de Gráficas permite, en muchos casos, el estudio de esta estructura y al aprovechar sus propiedades es posible construir los algoritmos buscados. Pero para otros problemas, no siempre resulta fácil aprovechar la estructura. Hay problemas lineales, como los de flujo en redes y acoplamiento que pueden resolverse de manera muy eficiente. Sin em-

bargo estos problemas están aparentemente muy relacionados con otros problemas que se consideran "intratables". Como ejemplo, los problemas de trayectoria más corta en una gráfica y de acoplamiento, en donde se conocen algoritmos $O(n^2)$ ¹ que los resuelven; en contraste con el problema del agente viajero, el cual resuelve la trayectoria más corta para visitar el conjunto de todos los nodos de una red exactamente una vez, que como es bien conocido, es un problema en la denominada clase NP-completa cuyos problemas son ampliamente considerados irresolubles por algoritmos polinomiales. Este fino límite entre problemas "fáciles" y "difíciles" es un fenómeno recurrente en los problemas de optimización combinatoria.

Para los problemas de optimización en la clase NP-completa, a la fecha, no se conocen algoritmos que los resuelvan en tiempo polinomial. Es por ello, que se han desarrollado algoritmos de tipo heurístico para resolver instancias grandes de problemas que pertenecen a esta clase. Estos algoritmos, no necesariamente encuentran la mejor solución al problema, pero en general se pueden obtener "buenas" soluciones cuando se aplican, aquí se entiende "buena" solución en términos de cercanía al valor óptimo. La ventaja principal de este tipo de algoritmos es que son rápidos y corren en un tiempo polinomial.

1.3 Optimización, satisfacibilidad y semioptimización .

Por ejemplo, en el problema del agente viajero se requiere de un trabajo de *optimización*: encontrar la trayectoria o ruta que sea tan barata como cualquier otra trayectoria o ruta factible. En otros problemas de optimización, ahora bien, el objetivo no es sólo el exhibir un objeto formal que satisfaga un conjunto de criterios establecidos sino también averiguar que cualidades no marcadas posee en el espacio de candidatos. Ejemplo de este estilo es el problema de las N-reinas, (véase por ejemplo

¹La notación $O(n^r)$ se utiliza para indicar que en el peor de los casos el esfuerzo computacional requerido por el algoritmo para resolver una instancia de tamaño n , crece de la forma αn^r donde α y r son independientes del tamaño n y de los datos de la instancia.

Laguna (1993)) donde se requiere de la *satisfacibilidad* de que las reinas no se maten entre sí.

La mayoría de los problemas pueden poseer ambos tipos de trabajos. Por ejemplo, el problema de las N-reinas se puede declarar como sigue: encontrar el emplazamiento más barato de las N reinas en el tablero de ajedrez donde cada reina capturada introduce un costo de una unidad. Puesto de esta manera, el problema llega a ser muy difícil puesto que no se conoce *a priori* cuando una solución de costo cero existe.

Siempre la diferencia en complejidad entre un trabajo de optimización y su contraparte de satisfacibilidad es substancial. El problema del agente viajero es un ejemplo de tal situación: encontrar una ruta a través de un conjunto de ciudades es trivial en el caso de que existan todas los posibles caminos entre cualesquiera dos ciudades, pero encontrar una ruta optimal es un problema NP-complejo. En tales casos, uno se ve forzado a relajar los requerimientos de optimalidad y moderar para encontrar una “buena” solución utilizando sólo un razonable esfuerzo de búsqueda. Se dice que se tiene un problema de *semioptimización* si hay un criterio de aceptación que tolera una vecindad alrededor de la solución optimal.

La mayoría de los problemas combinatorios prácticos son del tipo de semioptimización, por lo que se requiere de un balance razonable entre la calidad de la solución encontrada y el costo de búsqueda de tal solución. Más aún, dado que el esfuerzo requerido para muchos problemas de optimización combinatoria pueden alcanzar fácilmente configuraciones astronómicas, la relajación de la optimalidad es una necesidad económica. El problema básico en manejar un trabajo de semioptimización es el de idear algoritmos que garanticen cotas tanto en el esfuerzo de búsqueda como en su extensión para el cual el objetivo de optimización está comprometido. Un trabajo aún más ambicioso es el de equipar a tales algoritmos con un conjunto de parámetros ajustables que el usuario pueda cambiar para negociar entre la calidad de la solución y el monto de esfuerzo requerido.

La característica sobresaliente de las técnicas presentadas en este

trabajo es su aplicación general y la habilidad para obtener soluciones arbitrariamente cercanas a la óptima. Sin embargo, el obtener soluciones de alta calidad puede requerir de mucho esfuerzo computacional. Una comparación importante que puede hacerse con respecto a si una técnica es general o no es la que se presenta con respecto a la programación lineal y la programación dinámica. La programación lineal es un modelo muy específico para solucionar problemas en donde la función objetivo y las restricciones deben ser lineales; si el problema no cae dentro de este esquema restringido, la programación lineal no puede aplicarse. Por otro lado, la programación dinámica se puede usar para resolver muchos problemas de optimización y su aplicabilidad depende de la habilidad que se tenga para definir las etapas del problema, las ecuaciones recursivas, las variables de estado y las variables de decisión para un problema específico. En este sentido, las técnicas aquí presentadas resultan ser análogas a la programación dinámica. El uso y la calidad de sus resultados dependen del arte y la habilidad con que se definan sus diferentes componentes como son: las evaluaciones de la función objetivo de la nueva solución con respecto a la anterior, la estructura de vecindades y el grado de refinamiento de la implantación en la computadora.

Capítulo 2

Búsqueda tabú.

Muchos métodos para problemas combinatorios consisten de dos fases: construcción y mejoramiento. En este estudio se incorpora a la búsqueda tabú dentro de la fase de mejoramiento del proceso en orden a continuar la búsqueda cuando un óptimo local se encuentra. En este capítulo se da una introducción sobre el método de la búsqueda tabú y su implantación en computadora.

Otro aspecto interesante del este capítulo es que se presenta mediante el ejemplo numérico el concepto de “valle profundo” o “agujero negro” (ver Figura 2.2), el cual es uno de los elementos más importantes a resolver para evitar en algunos casos el ciclado y/o la suboptimalidad.

2.1 Introducción a la búsqueda tabú.

La Búsqueda Tabú (BT) es un procedimiento heurístico de “alto nivel” introducido y desarrollado en su forma actual por Fred Glover (1989) y (1990a), el cual se utiliza con gran éxito para resolver problemas de optimización cuya característica principal es la de “escapar” de la optimalidad local. Para una lista de aplicaciones véase por ejemplo Glover y Laguna (1993).

“La filosofía de la BT es la de manejar y explotar una colección de principios para resolver problemas de manera inteligente. Uno de los elementos fundamentales de la BT es el uso de la memoria flexible, desde el punto de vista de la BT, la memoria flexible envuelve el proceso dual de crear y explotar estructuras para tomar ventaja mediante combinar actividades de adquisición, evaluación y mejoramiento de la información de manera histórica” (Glover y Laguna (1993)).

En términos generales, el método BT puede esbozarse consistente en:

Se desea moverse paso a paso desde una solución factible inicial de un problema de optimización combinatoria hacia una solución que proporcione el valor mínimo de la función objetivo C . Para esto, se puede representar a cada solución por medio de un punto s (en algún espacio) y se define una vecindad $N(s)$ de cada punto s .

El paso básico del procedimiento consiste en empezar desde un punto factible s y generar un conjunto de soluciones en $N(s)$; entonces se escoge al mejor vecino generado s^* y se posiciona en ese nuevo punto ya sea que $C(s^*)$ tenga o no mejor valor que $C(s)$.

Hasta este punto se está cercano a las técnicas de mejoramiento local a excepción del hecho de que se puede mover a una solución peor s^* desde s .

La característica importante de la búsqueda tabú es precisamente la construcción de una lista tabú T de movimientos: aquellos movimientos que no son permitidos (movimientos tabú) en la iteración presente. La razón de esta lista es la de excluir los movimientos que nos pueden regresar a algún punto de una iteración anterior. Ahora bien, un movimiento permanece como tabú solo durante un cierto número de iteraciones, de forma que se tiene que T es una lista cíclica donde para cada movimiento $s \rightarrow s^*$ el movimiento opuesto $s^* \rightarrow s$ se adiciona al final de T donde el movimiento más viejo en T se elimina.

Las condiciones tabú tienen la meta de prevenir ciclos e inducir la exploración de nuevas regiones. La necesidad del significado de eliminar ciclos se debe a que, al moverse desde un óptimo local, una elección irrestricta de movimientos (persiguiendo aquellos con evaluaciones altas) permite igualmente regresar al mismo óptimo local.

Hay que apuntar, sin embargo, que la eliminación de ciclos no es la última meta en el proceso de búsqueda. En algunos casos, una buena trayectoria de búsqueda resultará al visitar una solución encontrada antes. El objetivo de manera amplia es el de continuar estimulando el descubrimiento de nuevas soluciones de alta calidad como se verá más adelante.

Ahora bien, las restricciones tabú no son inviolables bajo toda circunstancia. Cuando un movimiento tabú proporciona una solución mejor que cualquier otra encontrada, su clasificación tabú puede eliminarse. La condición que permite dicha eliminación se llama *criterio de aspiración*.

Es así como las restricciones tabú y el criterio de aspiración de la BT, juegan un papel dual en la restricción y guía del proceso de búsqueda. Las restricciones tabú, permiten que un movimiento se observe como admisible si no se aplican, mientras que el criterio de aspiración permite que un movimiento se observe como admisible si se satisface.

La búsqueda tabú en una forma simple descubre dos de sus elementos claves: La de restringir la búsqueda mediante la clasificación de ciertos movimientos como prohibidos (es decir, tabú) y el de liberar la búsqueda mediante una función de memoria de término corto que proporciona una "estrategia de olvido".

Tres aspectos merecen énfasis:

- (1). El uso de T proporciona la "búsqueda restringida" de elementos de la aproximación y por lo tanto las soluciones generadas dependen críticamente de la composición de T y de la manera como se actualiza.

(2). El método no hace referencia a la condición de optimalidad local, excepto implícitamente cuando un óptimo local mejora sobre la mejor solución encontrada previamente.

(3). Un “mejor” movimiento (más que un movimiento de mejora) se elige en cada paso.

Para problemas grandes, donde las vecindades pueden tener muchos elementos, o para problemas donde esos elementos son muy costosos para examinar, es de importancia el aislar a un subconjunto de la vecindad, y examinar este conjunto en vez de la vecindad completa. Esto puede realizarse en estadios permitiendo al subconjunto de candidatos expandirse si los niveles de aspiración no se encuentran.

2.2 Problema de calendarización.

A fin de describir diferentes aspectos de BT, se revisita el ejemplo de Barnes y Vanston(1981). Considere el problema de calendarización de una máquina con costos de penalización por retraso y costos de actualización ambos de tipo lineal.

En el tiempo cero, N trabajos llegan a una máquina de capacidad continua. Cada trabajo i ($i = 1, 2, \dots, N$) requiere de t_i unidades de tiempo en la máquina y tiene una penalización de retraso por cada unidad de tiempo de p_i a partir del tiempo cero; s_{ij} es el costo de actualización de calendarizar al trabajo j inmediatamente después del trabajo i . Dos trabajos falsos 0 y $N+1$, se incluyen en cada calendario, donde $t_0 = t_{N+1} = 0$ y $p_0 = p_{N+1} = 0$. Los costos $s_{0,j}$ y $s_{i,N+1}$ se consideran como los costos de la puesta inicial y de limpieza respectivamente. Un calendario tiene la forma:

$$\pi = (0, \pi(1), \pi(2), \dots, \pi(N), N + 1),$$

donde $\pi(i)$ es el índice del trabajo en la posición i del calendario. El objetivo es el de minimizar la suma de los costos de actualización y de retraso para todos los trabajos. En términos matemáticos, se desea:

$$(P) \quad \text{Minimizar} \quad F(\pi) = D(\pi) + S(\pi),$$

donde:

$$D(\pi) = \sum_{i=1}^N d_{\pi(i)} P_{\pi(i)},$$

$$S(\pi) = S_{0,\pi(1)} + \sum_{i=1}^{N-1} S_{\pi(i),\pi(i+1)} + S_{\pi(N),N+1},$$

$$d_{\pi(i)} = \sum_{j=1}^{i-1} t_{\pi(j)}, \quad i = 2, \dots, N, \quad \text{y} \quad d_{\pi(1)} = 0.$$

La clase de movimientos a tomarse dentro de la BT consiste en el cambio común por pares es decir, se intercambia la posición de dos trabajos para transformar un calendario a otro. Suponga que dado un calendario el trabajo $\pi(i)$ precede, pero no necesariamente es adyacente al trabajo $\pi(j)$. Un *movimiento de intercambio* es un rearrreglo de sólo los trabajos $\pi(i)$ y $\pi(j)$ de forma tal que el trabajo $\pi(i)$ se mueve a la posición j y el trabajo $\pi(j)$ se mueve a la posición i . El *valor del movimiento* es la diferencia entre el valor de la función objetivo después del movimiento, $F(\tilde{\pi})$, y el valor de la función objetivo antes del movimiento, $F(\pi)$, i.e.,

$$\text{valor_movimiento} = F(\tilde{\pi}) - F(\pi).$$

Los valores de movimiento por lo general proporcionan una base fundamental para evaluar la calidad de un movimiento, aunque otros criterios también son importantes como se verá más adelante.

Un criterio de paro utilizado en este tipo de métodos, es el de considerar un tiempo límite de cómputo prefijado.

El movimiento que se realiza en la iteración es el de menor valor de de todos los movimientos *candidatos* actuales. Un movimiento se considera que es un candidato si pertenece a la vecindad, en este ejemplo, se considera que se está en la vecindad, si los trabajos a intercambiarse están dentro de una distancia específica (número de posiciones). Dado que se está minimizando, el mejor movimiento candidato es aquel que posee el menor valor. De manera más precisa, el mejor movimiento se selecciona del conjunto de movimientos *admisibles*. Un movimiento es admisible si es no tabú o si su estatus tabú puede eliminarse por medio

del criterio de aspiración. El mejor movimiento entonces se realiza y la estructura de datos tabú se actualiza.

El proceso fundamental mediante el que BT busca trascender la optimalidad local es el de introducir mecanismos para que ciertos movimientos sean prohibidos (ver Figuras 2.1 y 2.2).

En la solución del problema (P), la preocupación principal es el de crear un estatus tabú que prevenga que algún movimiento se invierta bajo la jurisdicción de la memoria de término corto, la cual se escoge para que (P) tenga un número específico de movimientos futuros.

La memoria de término corto de la BT constituye una forma de exploración agresiva que busca realizar el mejor movimiento posible (Esquema 2.1), para satisfacer ciertas restricciones. Esas restricciones están diseñadas para prevenir el regresarse o la repetición de cierto número de veces de ciertos movimientos mediante la ejecución de atributos seleccionados de esos movimientos prohibidos(tabú).

Otra manera de identificar atributos de un movimiento de intercambio es el de introducir información adicional, mediante no sólo hacer referencia de los elementos intercambiados sino también de las posiciones ocupadas por esos elementos en el momento de su cambio. Ahora bien, no existe una forma que se pueda decir que es la mejor, sólo se puede identificar mediante pruebas.

Paso 1. $i :=$ movimiento con mejor valor actual. **Repite Paso 2.** $j_i :=$ movimiento candidato. **Paso 3.** Si j_i es admisible entonces **Paso 4.** sino **Paso 6.** **Paso 4.** Si $F(j_i) < F(i)$ entonces **Paso 5.** sino **Paso 6** **Paso 5.** $i := j_i$. **Paso 6.** Ir a **Paso 2.** hasta(último movimiento candidato). Ir a **Paso 1.**

Esquema 2.1 Selección del mejor candidato admisible.

La meta principal de las restricciones tabú es el permitir al método ir a puntos más allá de la optimalidad local mientras se permita la realización de movimientos de alta calidad en cada paso al mismo tiempo de que exista una negociación balanceada con respecto al esfuerzo computacional al examinar muestras muy grandes, por lo que, en ocasiones

es deseable incrementar el porcentaje de movimientos posibles para que reciban una clasificación tabú. Esto se puede lograr ya sea mediante el aumento en la pertenencia tabú o mediante el cambio de la restricción tabú.

Además se requiere de una estructura de datos para guardar el seguimiento de los movimientos que son clasificados como tabú y para liberar aquellos movimientos de su condición tabú cuando su pertenencia a la memoria de término corto expire. El acompañamiento de la memoria basada en la pertenencia junto con la memoria basada en la frecuencia adicionan el efecto de que se puede estipular una historia selectiva de los estados encontrados durante la búsqueda, y reemplazando la vecindad actual por una vecindad modificada que depende de este proceso histórico.

El último elemento en el procedimiento básico es el *criterio del nivel de aspiración*, cuyo propósito es el de permitir que movimientos tabúes “buenos” se seleccionen si el nivel de aspiración se alcanza. El apropiado uso de tal criterio puede ser muy importante para posibilitar que un método de BT alcance sus mejores niveles de realización. El criterio de aspiración estándar, es el que permite que el estatus tabú se elimine si una solución mejor que la alcanzada hasta el momento se puede obtener, *i.e.*, a un movimiento tabú se le permite ejecutarse si:

$$F(\pi) + \text{valor movimiento} < F(\pi^*)$$

Ahora bien, otros criterios de aspiración pueden también proporcionar efectividad para mejorar la búsqueda, como se verá más adelante.

Una base para uno de esos criterios proviene de introducir el concepto de *influencia* (Glover y Laguna (1993)), la cual mide el grado de cambio inducido en la estructura de solución o de factibilidad. La influencia por lo general se asocia con la idea de distancia del movimiento, *i.e.*, donde un movimiento de gran distancia se concibe como de mayor influencia.

El método inicia con una solución heurística factible inicial, la cual

se guarda como la mejor encontrada.

Un paso crítico, el cual envuelve la orientación agresiva de la memoria de término corto, es la elección del mejor candidato admisible. La función *mejor_movimiento* es una que identifica a un movimiento para el cual el valor del movimiento es el más pequeño. El dominio de la función es el conjunto de todos los movimientos admisibles. El *mejor_movimiento* no tiene que ser necesariamente uno que mejore. Primero, cada uno de los movimientos de la lista de candidatos se evalúa en turno (Esquema 2.1).

Ahora bien, conforme la búsqueda progresa, la forma de la evaluación empleada por la búsqueda tabú llega a ser más adaptativa, incorporando referencias concernientes para la *intensificación* y la *diversificación* regional de búsqueda. Cabe aclarar que en las estrategias basadas en consideraciones de término corto la clasificación tabú sirve para identificar elementos de la vecindad del movimiento actual, mientras que en las estrategias de término intermedio y largo pueden no contener soluciones en esta vecindad, sino que por lo general consisten de seleccionar soluciones *élites* (óptimos locales de alta calidad) encontrados en varios puntos en el proceso de solución. Dichas soluciones *élites* se identifican como elementos de un conglomerado regional en las estrategias de intensificación de término intermedio, y como elementos de diferentes conglomerados en las estrategias de diversificación de término largo.

La esencia del método depende de cómo el registro de la historia (ver Tabla 2.1) se define y se utiliza, y de cómo los candidatos y la función de evaluación se determinan.

Revisar el estatus tabú es el primer paso en la admisibilidad. Si el movimiento no es tabú, es inmediatamente aceptado como admisible; de otra forma, el criterio de aspiración da una oportunidad para eliminar el estatus tabú, proporcionando al movimiento una segunda oportunidad para clasificarse como admisible.

En algunos casos, si las restricciones tabú y el criterio de aspiración

son suficientemente limitados, ninguno de los movimientos posibles serán clasificados como admisible. Un movimiento "menos inadmisibles" se toma para manipular tal posibilidad y se elige si no emergen alternativas admisibles.

La longitud de la lista tabú es un parámetro, si es demasiado pequeño el ciclado ocurrirá, pero si es demasiado grande, restringirá bastante la búsqueda para poder saltar "valles profundos" (i.e. el mejor mínimo local) (ver Figuras 2.1 y 2.2) del espacio de valores de la función objetivo. Una faceta importante de la BT es la habilidad de localizar un rango robusto de longitudes de la lista tabú mediante pruebas empíricas preliminares para identificar para una clase de problemas los tipos de atributos y de restricciones tabú que se realizan de manera más efectiva. Acerca de esto, existe el uso de listas tabú múltiples, cada una desarrollada para un tipo particular de atributo.

Cuando diferentes tipos de atributos se manejan de esta forma, pueden estar dados con pesos variantes, dependiendo de su clasificación y permanencia, para determinar el estatus tabú de los movimientos que lo contienen.

En muchas aplicaciones, el componente de corto término por sí mismo ha producido soluciones superiores a aquellas encontradas mediante procedimientos alternativos, y el uso de memoria de mayor término en esos casos se ha eludido. Ahora bien, la memoria de término intermedio y largo puede ser importante para obtener mejores resultados para problemas difíciles, como se verá en las siguientes secciones.

La memoria de término intermedio y largo opera primariamente como una base de las estrategias de intensificar y diversificar la búsqueda.

En este ejemplo los trabajos se indexan de acuerdo al siguiente orden: $i < j$ implica que $\frac{t_i}{p_i} < \frac{t_j}{p_j}$.

Se conoce la solución óptima para este problema y que es 13,500 para $\pi = (0, 2, 1, 4, 3, 5, 6)$, recuérdese que los trabajos 0 y 6 son sólo trabajos falsos. En este ejemplo se considerará el criterio que impide cualquier movimiento que resulte en un calendario donde cualquiera de

los trabajos ya sea a $\pi(i)$ ocupe la posición i o el trabajo $\pi(j)$ ocupe la posición j , además se tienen las siguientes condiciones:

Trabajo i	Duración t_i	Penalización p_i	Radio t_i/p_i	Orden Natural
1	3	700	.004286	$\pi(5)$
2	4	800	.005	$\pi(4)$
3	1	100	.01	$\pi(3)$
4	4	300	.0133	$\pi(2)$
5	5	200	.025	$\pi(1)$

S_{ij} MATRIZ DE ACTUALIZACION DE COSTOS						
i/j	1	2	3	4	5	6
0	1100	600	1200	2000	1400	∞
1	∞	1300	700	1200	1100	1000
2	900	∞	1100	1300	600	1200
3	900	1000	∞	2000	700	1500
4	1000	700	800	∞	600	1200
5	1400	1300	1200	1300	∞	900

Para iniciar nuestro proceso de búsqueda tabú consideraremos:

$$\left\{ \begin{array}{l} \text{Punto Inicial } \pi_0 = (0, 5, 4, 3, 2, 1, 6). \\ F(\pi_0) = 26600. \\ \text{longitud_tabu} = 7. \\ \text{Distancia máxima} = 1. \end{array} \right.$$

A continuación se presentarán la tablas de iteraciones, en donde se indicará el número de iteración del proceso, los calendarios generados y los correspondientes valores de la función objetivo, así como si el calendario propuesto es un movimiento tabú y el mejor movimiento admisible para continuar con el proceso.

En este ejemplo las vecindades completas se examinan, es decir, se realizan las evaluaciones completas de los cambios de los pares hacia adelante a una distancia de uno. Entonces el mejor cambio se realiza, en este caso el que minimice la función objetivo y no sea movimiento tabú o en el caso de que lo sea para que sea admisible debe de satisfacer el criterio de aspiración que en este caso se considera que se satisface si el valor de la función objetivo mejora sobre todos los valores anteriormente encontrados.

La matriz tabú se construye al inicio del procedimiento, donde las filas de la matriz representan las posiciones y las columnas a los trabajos y se actualiza en cada iteración durante la fase de mejoramiento del algoritmo. Si un elemento (i, j) pertenece a esta matriz en una iteración dada, no se le permite realizar el cambio del trabajo i a la posición j , y se actualiza en cada iteración. Recuérdese que es posible vencer la restricción tabú en el caso de que se satisfaga el criterio de aspiración

La matriz de frecuencias es la que lleva la "historia" del procedimiento y es la que se utiliza para la formación de la función de memoria término largo, la cual permite la diversificación de la búsqueda, es decir, es posible el dirigir la búsqueda "más cercana" ó "más alejada" de las regiones exploradas.

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	$F(\pi)$	TABU = 1	MEJOR Admisible
0	(0,5,4,3,2,1,6)	26600		*
1	(0,4,5,3,2,1,6)	26200		*
	(0,5,3,4,2,1,6)	27300		
	(0,5,4,2,3,1,6)	26200		
	(0,5,4,3,1,2,6)	26700		

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
0	0	0	0	7	0	0	0	1	0
0	0	0	7	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Observe que existen dos movimientos admisibles que nos proporcionan los mejores valores de la función objetivo, por lo que, se puede escoger cualquiera de los dos, en este caso se elige el primero *i.e.*, el movimiento que nos proporciona el calendario $\pi = (0, 4, 5, 3, 2, 1, 6)$ como punto inicial para la siguiente iteración.

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	$F(\pi)$	TABU = 1	MEJOR Admisible
1	(0,4,5,3,2,1,6)	26200		*
2	(0,5,4,3,2,1,6)	26600	1	*
	(0,4,3,5,2,1,6)	25900		
	(0,4,5,2,3,1,6)	26000		
	(0,4,5,3,1,2,6)	26300		

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
0	0	0	0	6	0	0	0	1	0
0	0	0	6	7	0	0	1	0	1
0	0	7	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

2.2. Problema de calendarización. _____ 25

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
2	(0,4,3,5,2,1,6)	25900		*
3	(0,3,4,5,2,1,6)	26100	1	*
	(0,4,5,3,2,1,6)	26200	1	
	(0,4,3,2,5,1,6)	22800		
	(0,4,3,5,1,2,6)	26200		

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
0	0	0	0	5	0	0	0	1	0
0	0	0	5	6	0	0	1	0	1
0	0	6	0	7	0	1	0	0	1
0	7	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0

Observe que en este caso el movimiento que nos proporciona el calendario $\pi = (0, 4, 3, 2, 5, 1, 6)$ es el movimiento que toma el valor objetivo más pequeño en la vecindad por lo que, se toma como punto inicial para la siguiente iteración.

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
3	(0,4,3,2,5,1,6)	22800		*
4	(0,3,4,2,5,1,6)	22800	1	*
	(0,4,2,3,5,1,6)	22500	1	
	(0,4,3,5,2,1,6)	25900	1	
	(0,4,3,2,1,5,6)	19800		

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
0	0	0	0	4	0	0	0	1	0
0	0	0	4	5	0	0	1	0	1
0	0	5	0	6	0	1	0	0	1
0	6	0	0	7	1	0	0	0	1
7	0	0	0	0	0	0	0	0	1

En esta iteración el único movimiento admisible es el que proporciona el calendario $\pi = (0, 4, 3, 2, 1, 5, 6)$ con un valor objetivo de 19800.

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
4	(0,4,3,2,1,5,6)	19800		
5	(0,3,4,2,1,5,6)	19800	1	
	(0,4,2,3,1,5,6)	19400	1	
	(0,4,3,1,2,5,6)	19200	1	*
	(0,4,3,2,5,1,6)	22800	1	

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
0	0	0	0	3	0	0	0	1	0
0	0	0	3	4	0	0	1	0	1
0	7	4	0	5	1	1	0	0	1
7	5	0	0	6	1	1	0	0	1
6	0	0	0	0	0	0	0	0	1

Note que en esta iteración los movimientos que proporcionan los calendarios $\pi = (0, 4, 2, 3, 1, 5, 6)$ y $\pi = (0, 4, 3, 1, 2, 5, 6)$ son movimientos tabú pero ambos satisfacen el criterio de aspiración por lo que se toma como punto inicial para la siguiente iteración el calendario con valor más pequeño en la vecindad.

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
5	(0,4,3,1,2,5,6)	19200		*
6	(0,3,4,1,2,5,6)	19600	1	*
	(0,4,1,3,2,5,6)	18500	1	
	(0,4,3,2,1,5,6)	19800	1	
	(0,4,3,1,5,2,6)	23200	1	

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
0	0	0	0	2	0	0	0	1	0
0	0	7	2	3	1	0	1	0	1
7	6	3	0	4	1	1	1	0	1
6	4	0	0	5	1	1	0	0	1
5	0	0	0	0	0	0	0	0	1

En esta iteración el movimiento que proporciona el calendario $\pi = (0, 4, 3, 2, 1, 5, 6)$ es tabú, pero satisface el criterio de aspiración por lo que se toma como punto inicial para la siguiente iteración. De manera análoga se sigue el procedimiento hasta la iteración 10 la cual tiene las siguientes tablas:

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
9	(0,2,1,4,3,5,6)	13500		***
10	(0,1,2,4,3,5,6)	14100	1	*
	(0,2,4,1,3,5,6)	15500	1	
	(0,2,1,3,4,5,6)	14000	1	
	(0,2,1,4,5,3,6)	14700		

MATRIZ TABU					MATRIZ DE FRECUENCIAS				
7	0	0	4	0	1	1	0	1	0
4	7	3	5	0	2	1	1	1	1
3	6	4	0	0	1	2	1	1	1
2	5	0	0	1	1	1	1	0	1
1	0	0	0	0	0	0	0	0	0

Si en alguna iteración ya no existen puntos admisibles por lo que se tendría que utilizar ahora las funciones de memoria de término intermedio (intensificación) y de término largo (diversificación).

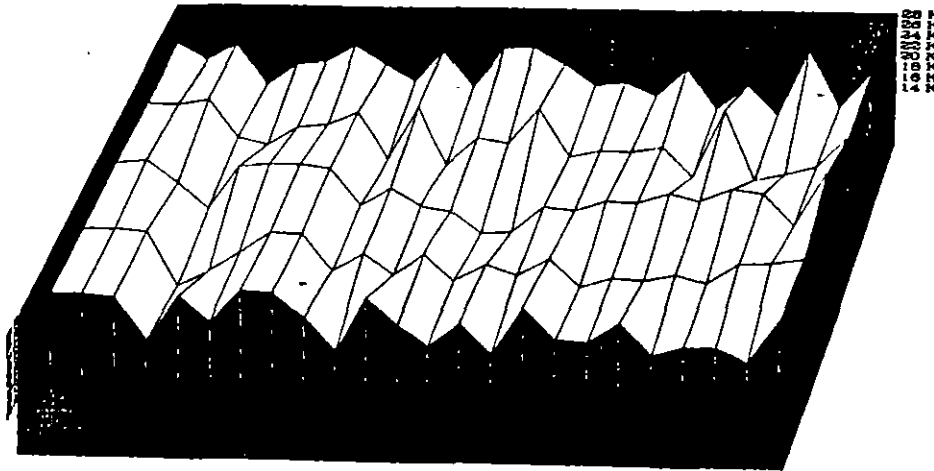
El contador de frecuencia muestra la distribución de los movimientos a través de las iteraciones. Se utiliza ese contador para diversificar la búsqueda, maniobrando dentro de nuevas regiones. Esta influencia de diversificación se restringe para operarse sólo en ocasiones particulares. En este caso, donde ningún movimiento de mejora admisible existe. El uso de la información de la frecuencia se utiliza por lo general para penalizar movimientos que no mejoran mediante el asignar una penalización grande a los pares intercambiados con mayor frecuencia provocando con esto que se pierda lo atractivo de tales intercambios.

En suma, las frecuencias definidas sobre diferentes subconjuntos de soluciones anteriores, particularmente subconjuntos de soluciones élites consistentes de óptimos locales de alta calidad encontrados en varios puntos en el proceso de solución, proporcionan las estrategias complementarias de intensificación.

Una aproximación que está cercanamente unida a los orígenes de la BT y que proporciona un interjuego efectivo entre la intensificación y la diversificación es la *estrategia de oscilación*.

La estrategia de oscilación opera mediante el moverse hasta pegarle a una frontera, representada por la factibilidad o un estado de construcción que normalmente puede representarse por un punto donde el método puede parar. En vez de parar, ahora bien, la definición de vecindad se extiende o el criterio de evaluación para seleccionar movimientos

se modifica, para permitir que se pueda cruzar esa frontera. La aproximación entonces procede para una profundidad específica más allá de la frontera y entonces se regresa. En este punto la frontera de nuevo se aproxima y se cruza, esta vez desde la dirección opuesta, procediendo a un nuevo punto en turno. El proceso de aproximar y cruzar repetidamente la frontera desde diferentes direcciones crea una forma de oscilación que es la que le da el nombre a la estrategia. El control sobre esta oscilación se establece mediante el generar evaluaciones y reglas de movimientos modificadas, dependiendo de la región en la cual se está actualmente navegando y de la dirección de la búsqueda.

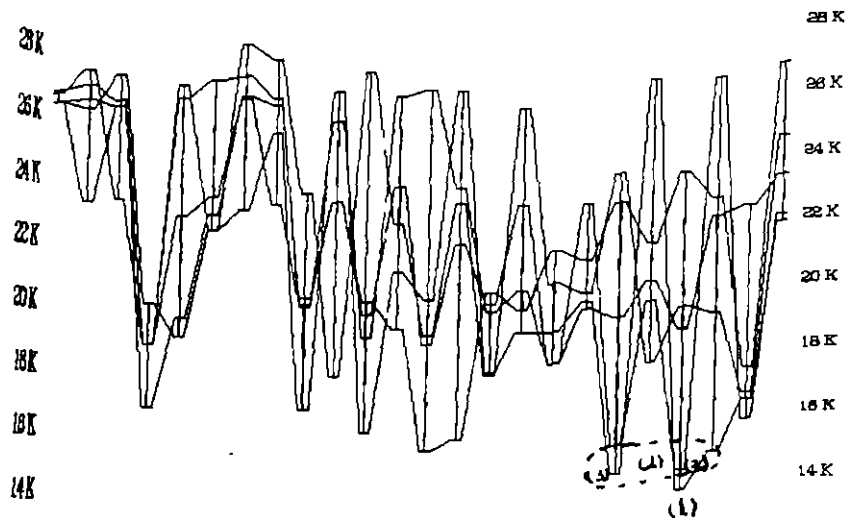


Gráfica de los valores objetivos del ejemplo.

Figura 2.1

Un ejemplo simple de esta aproximación ocurre para el problema de la mochila multidimensional donde los valores de las variables cero-uno se cambian de 0 a 1 hasta alcanzar la frontera de factibilidad. El método entonces continúa dentro de la región infactible utilizando el mismo tipo de cambios, pero con un evaluador modificado. Después de un número seleccionado de pasos, la dirección se invierte mediante el cambiar las variables de 1 a 0. El criterio de evaluación se maneja

para mejorar y varia de acuerdo a cuando el movimiento es de más a menos infactible o de menos a más infactible y se acompañan mediante restricciones asociadas sobre cambios admisibles para los valores de las variables.



Valle profundo(2), óptimo global(1), óptimos locales(3).

Figura 2.2

Ahora bien, para incorporar la estrategia de oscilación, no necesariamente se tiene que definir en términos de factibilidad, sino que puede definirse donde la búsqueda parece gravitar. La oscilación consiste en forzar la búsqueda a movimientos fuera de esta región y el permitir regresarse a la región, ofreciendo de esta manera una forma efectiva para eliminar entrapamientos suboptimales en las búsquedas estándares.

2.3 Memoria de término intermedio y largo.

Como se ha visto, el método de la BT empieza con una solución factible inicial y en el proceso de ejecución, el procedimiento actualiza a los

arreglos y elementos de la función memoria. Entonces el proceso se repite hasta que el criterio de terminación se encuentra.

En el método de BT descrito en el ejemplo, el “mejor” movimiento que se realiza en cada iteración se especifica como el movimiento admisible con el menor valor objetivo. Ahora bien, esta estrategia no garantiza que el movimiento seleccionado permita la búsqueda en la dirección de la solución optimal, por lo que, se requiere de técnicas que nos permitan integrar las estrategias de intensificación y diversificación de manera efectiva, basándose sobre las funciones de memoria de término intermedio y largo de la BT. En otras palabras, es de importancia vital el “mirar” la *dependencia regional* de buenos criterios de decisión, no sólo en términos de movimientos de mejoramiento y no mejoramiento.

2.4 Intensificación y diversificación regional.

La fase de intensificación proporciona una forma simple para enfocar la búsqueda al rededor de la mejor solución (o conjunto de soluciones élites) hasta el momento.

Para entender la importancia de estos recursos de la BT, consideremos dos corridas del ejemplo numérico, pero variando los valores de algunos de los parámetros iniciales.

CORRIDA 1

$$\left\{ \begin{array}{l} \text{Punto Inicial } \pi_0 = (0, 5, 4, 3, 2, 1, 6). \\ F(\pi_0) = 26600. \\ \text{longitud_tabu} = 7. \\ \text{Distancia máxima} = 2. \end{array} \right.$$

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	$F(\pi)$	TABU = 1	MEJOR Admisible
0	(0,5,4,3,2,1,6)	26600		*
1	(0,5,2,3,4,1,6)	25500		*
	(0,3,4,5,2,1,6)	26100		
	(0,5,4,1,2,3,6)	26600		
2	(0,3,2,5,4,1,6)	20700	1	*
	(0,3,2,1,4,5,6)	14900		
	(0,3,4,5,2,1,6)	26100	1	
3	(0,1,2,3,4,5,6)	14900	1	
	(0,3,4,1,2,5,6)	19600	1	
	(0,3,2,5,4,1,6)	20700	1	

Como se puede observar, en la iteración 2 se llegó a un óptimo local, el cual está dado por el calendario $\pi = (0, 3, 2, 1, 4, 5, 6)$ con un valor objetivo de 14900, y en la iteración 3 también se llega a otro óptimo local dado por el calendario $\pi = (0, 1, 2, 3, 4, 5, 6)$ con igual valor objetivo, pero en éste último caso se tiene que el punto es tabú y además no se tienen soluciones admisibles. Se puede entonces pensar en escoger a éste último óptimo local para intensificar la búsqueda dentro de la región por lo que se toma como punto de arranque y se inicializa la tabla tabú para continuar la búsqueda, pero se observa que en la iteración 10 se cae en un ciclo, y no existe mejoramiento por lo que se puede considerar que se está en un valle profundo (o agujero negro, ver Figura 2.2).

Ahora bien, en este ejemplo la búsqueda pudo estar muy restringida debido al valor de la longitud tabú que es muy alto, por lo que se realizó una segunda corrida:

CORRIDA 2

$$\left\{ \begin{array}{l} \text{Punto Inicial : } \pi_0 = (0, 5, 4, 3, 2, 1, 6) \\ F(\pi_0) = 26600 \\ \text{longitud_tabu} = 3 \\ \text{Distancia máxima} = 2 \end{array} \right.$$

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	$F(\pi)$	TABU = 1	MEJOR Admisible
0	(0,5,4,3,2,1,6)	26600		*
1	(0,5,2,3,4,1,6)	25500		*
	(0,3,4,5,2,1,6)	26100		
	(0,5,4,1,2,3,6)	26600		
2	(0,3,2,5,4,1,6)	20700	1	*
	(0,5,4,3,2,1,6)	26600		
	(0,5,2,1,4,3,6)	22600		
3	(0,5,2,3,4,1,6)	25500	1	
	(0,3,4,5,2,1,6)	26100	1	
	(0,3,2,1,4,5,6)	14900		
4	(0,1,2,3,4,5,6)	14900	1	*
	(0,3,4,1,2,5,6)	19600		
	(0,3,2,5,4,1,6)	20700	1	

Se puede observar que en la iteración 3 se alcanza un óptimo local y en la iteración 4 se alcanza el otro óptimo local que constituyen un agujero negro, pero en esta corrida a diferencia de la anterior se tiene un punto admisible el cual se toma para proseguir con la búsqueda pero en las iteraciones 13 y 14 se vuelven a alcanzar los óptimos locales y se inicia un ciclo a partir de la iteración 20.

Las dos corridas anteriores indican la necesidad de contar con elementos que nos permitan salir de este tipo de entrampamientos subóptimos, por lo que se debe recurrir a un análisis retrospectivo del proceso de búsqueda que nos pueda proporcionar un reconocimiento de patrones de comportamiento para poder identificar regiones no visitadas.

A manera de ejemplo se considera la matriz de frecuencias de la corrida 2 después de 20 iteraciones, donde las filas indican las posiciones y las columnas los trabajos:

2	0	3	0	2
0	4	0	3	0
4	0	4	0	5
0	3	0	4	0
2	0	2	0	2

Tabla 2.1 Matriz de Frecuencias de la Corrida 2.

Una forma de “aprender” qué regiones no se han visitado, es el observar la frecuencia con la cual un cierto trabajo no se ha seleccionado para una posición particular, así por ejemplo, se tiene que el trabajo 1 no se ha localizado en las posiciones 2 y 4, que el trabajo 3 se ha localizado más frecuentemente en la posición 3 de los calendarios, etcétera.

A partir de la matriz histórica de la búsqueda se pueden generar calendarios que se utilizan como puntos de arranque para nuevos procesos de búsqueda, a manera de ilustración se tienen de manera empírica los siguientes ejemplos:

$$\left\{ \begin{array}{l} \text{Punto Inicial : } \pi_0 = (0, 2, 1, 5, 3, 4, 6). \\ F(\pi_0) = 16300. \\ \textit{longitud_tabu} = 3. \\ \text{Distancia máxima} = 2. \end{array} \right.$$

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	$F(\pi)$	TABU = 1	MEJOR Admisible
0	(0,2,1,5,3,4,6)	16300		*
1	(0,2,1,4,3,5,6)	13500		*
	(0,5,1,2,3,4,6)	23400		
	(0,2,3,5,1,4,6)	18500		

$$\left\{ \begin{array}{l} \text{Punto Inicial : } \pi_0 = (0, 2, 3, 5, 1, 4, 6). \\ F(\pi_0) = 18500. \\ \textit{longitud_tabu} = 3. \\ \text{Distancia máxima} = 2. \end{array} \right.$$

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
0	(0,2,3,5,1,4,6)	18500		*
1	(0,2,1,5,3,4,6)	16300		*
	(0,5,3,2,1,4,6)	23100		
	(0,2,1,5,3,4,6)	17300		
2	(0,5,1,2,3,4,6)	23400	1	*
	(0,2,3,5,1,4,6)	18500		
	(0,2,1,4,3,5,6)	13500		

{ Punto Inicial : $\pi_0 = (0, 5, 1, 4, 3, 2, 6)$.
 $F(\pi_0) = 24500$.
 $longitud_tabu = 3$.
 Distancia máxima = 2.

ITERACION	$\pi = (0, \pi(1), \pi(2), \pi(3), \pi(4), \pi(5), 6)$	F(π)	TABU = 1	MEJOR Admisible
0	(0,5,1,4,3,2,6)	24500		*
1	(0,4,1,5,3,2,6)	23300		*
	(0,5,3,4,1,2,6)	27800		
	(0,5,2,1,3,4,6)	23400		
2	(0,5,1,4,3,2,6)	24500	1	*
	(0,4,3,5,1,2,6)	26200		
	(0,4,1,2,3,5,6)	18900		
3	(0,2,1,4,3,5,6)	13500	1	*
	(0,4,3,2,1,5,6)	19800		
	(0,4,1,5,3,2,6)	23300	1	

Ahora bien, la diversificación se restringe para operarse sólo en ocasiones particulares. En este caso, se seleccionan las ocasiones donde ningún movimiento de mejora admisible existe. Por lo general, se utiliza la información de la frecuencia para penalizar a los movimientos que

no mejoran la búsqueda mediante el asignar una penalización grande a intercambios de pares con mayores contadores de frecuencia.

Una implantación sencilla de tal técnica se puede realizar de la siguiente manera: Primero, se cuenta el número de veces que cada movimiento digamos m se ha realizado en orden a calcular su frecuencia $f(m)$. Entonces una penalización $p(m)$ se asocia a cada movimiento de la siguiente manera:

$$p(m) = \begin{cases} 0 & \text{si } m \text{ alcanza un criterio de aspiración.} \\ wf(m) & \text{de otra forma.} \end{cases}$$

donde w es una constante. Entonces el valor del movimiento puede ser:

$$F(x + m) - F(x) + p(m).$$

El peso w depende del problema, del tipo de movimiento y de la vecindad. Glover y Laguna (1993) indican que en muchas aplicaciones se ha observado que este peso es aproximadamente proporcional a la raíz cuadrada del tamaño de la vecindad multiplicada por la desviación estándar del valor (sin penalización) de cada movimiento ejecutado durante la búsqueda.

Como se puede observar, la función de penalización depende directamente del criterio de aspiración, por lo que ahora nos enfocaremos a este concepto.

2.5 Criterios de aspiración.

Los criterios de aspiración se introducen en la BT para determinar cuando las restricciones tabú pueden sobrellevarse para remover una clasificación tabú que de otra manera se aplicaría a un movimiento. El uso apropiado de tales criterios puede ser muy importante para que un método BT alcance sus mejores niveles de realización.

En las primeras aplicaciones de la BT se aplicó tan solo un tipo sencillo de criterio de aspiración, consistente de remover una clasificación

tabú a un movimiento si éste permitía una solución mejor que la mejor encontrada hasta el momento, tal regla se ilustró en los ejemplos de los capítulos anteriores.

En Glover y Laguna (1993), se indica que las aspiraciones son de dos clases: *aspiraciones de movimiento* y *aspiraciones de atributo*. Una aspiración de movimiento, cuando se satisface, revoca la clasificación tabú del movimiento. Una aspiración de atributo, cuando se satisface revoca el estatus tabú del atributo. En éste último caso el movimiento puede o no cambiar su clasificación tabú, dependiendo de si la restricción tabú puede activarse por más de un atributo. Así entonces se pueden tener los siguientes criterios de aspiración:

Aspiración por Default: Si todos los movimientos posibles son clasificados como tabú, entonces el movimiento “menos tabú” se selecciona.

Aspiración por Objetivo: Una aspiración de movimiento se satisface, permitiendo que un movimiento x sea un candidato para seleccionarse si $F(x) < \text{mejor costo}$.

Aspiración por Dirección de Búsqueda: Un atributo de aspiración para e se satisface si la dirección en e proporciona un mejoramiento y el actual movimiento es un movimiento de mejora.

2.6 Problema del agente viajero

Considere n ciudades y una matriz $(c_{i,j})$ de orden $n \times n$, cuyos elementos denotan los costos entre cada par i, j de ciudades. Se define un recorrido como una trayectoria cerrada que visita cada ciudad exactamente una vez. El problema es encontrar el recorrido de costo mínimo.

Una formulación del problema es:

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{i,j} X_{i,j}$$

sujeto a:

$$\sum_i X_{i,j} = 1, \quad i = 1, 2, \dots, n, \quad \sum_j X_{i,j} = 1, \quad j = 1, 2, \dots, n, \quad X_{i,j} = 0, 1 \quad \forall i, j,$$

$$c_i - c_j + nX_{i,j} \leq n - 1, \quad \forall i \neq j, \quad \forall c_i, c_j \in \mathbb{R},$$

donde:

$$X_{i,j} = \begin{cases} 1 & \text{si se va al lugar } j \text{ después del lugar } i, \\ 0 & \text{si no se va al lugar } j \text{ después del lugar } i. \end{cases}$$

Por otra parte, en este problema se puede observar que una solución está dada por una permutación cíclica $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ donde $\pi(k)$ denota la ciudad a visitar después de la ciudad k , con $\pi^l(k) \neq k, l = 1, 2, \dots, n - 1$ y $\pi^n = k, \forall k$. Aquí $\pi^l(k)$ se entiende como la aplicación de l veces la permutación k . Cada solución corresponde a un recorrido. El espacio de soluciones está dado por

$$S = \{\text{todas las permutaciones } \pi \text{ cíclicas de las } n \text{ ciudades}\}$$

por lo que el problema se puede plantear como:

$$(P1) \quad \text{Minimizar } f(\pi) = \sum_{i=1}^n c_{i,\pi(i)}, \quad \forall \pi \in S.$$

2.6.1 Disposición de desechos sólidos

Planteamiento del problema

La región marina¹ representa en la actualidad una zona de gran importancia para la economía del país, ya que de ella se extrae el 75% de la producción nacional de hidrocarburos.

Toda esta producción se lleva a cabo mediante una importante infraestructura marina y submarina formada principalmente por las plantas de producción, perforación, medición, rebombeo e inyección, así como de una gran red de ductos submarinos para la conducción de hidrocarburos.

La actividad en las plataformas marinas se realiza durante las 24 hrs. del día de los 365 días del año, generándose una gran cantidad de desechos industriales y domésticos, clasificándose éstos en biodegradables (restos de alimentos) y no biodegradables (chatarra, plásticos, madera, etc.).

Para el tratamiento de estos desechos se cuenta con equipos anti-contaminantes abordo de las plataformas como son: trituradores de alimentos, compactadores e incineradores de basura.

Solamente los restos de alimentos son vertidos al mar después de ser triturados, y el volumen de desechos restante se envía actualmente a la terminal marítima de Dos Bocas, Tabasco, contando para ello con el apoyo de barcos abastecedores de gran capacidad.

Objetivo

De acuerdo con la política de Pemex para optimizar el aprovechamiento de sus recursos, se plantea la elaboración de un estudio en función del costo de transportación marina de todo el material que se genera en las plataformas localizadas en la sonda de Campeche. Se desea conocer

¹de los Cobos Silva S.G., "Optimización combinatoria en la industria del petróleo", notas de curso: VI Escuela nacional de optimización y análisis numérico, abril 1996, Puebla México.

cuales son las rutas óptimas de recolección de desechos y los resultados nos permitirán considerar la posibilidad de construir un centro de acopio de desechos sólidos en Cd. del Carmen, Campeche.

Información

Por motivos de seguridad nacional, muchos datos son ficticios pero se ha tratado de que sean congruentes entre sí para que proporcionen una idea de la magnitud del problema real, el resto de la información ha sido obtenida de las diversas publicaciones de PEMEX.

- De acuerdo con las actividades que se realizan, los complejos de producción generan actualmente en promedio 440 toneladas de desechos sólidos por mes, que representan el 70% del total de desechos generados en el área marina. La recolección de chatarra y desechos sólidos en cada complejo de producción se lleva a cabo dos veces al mes, por un barco dedicado exclusivamente a esa actividad. Para cumplir con las normas internacionales, la capacidad de recolección debe ser 1.1 veces el volumen de desechos generados, por lo que los barcos recolectores de 250 tons. cumplen con esta norma.
- El 30% restante de los desechos sólidos, se genera en plataformas de tipo exploratorio y de perforación que son plataformas que se desplazan de acuerdo a un programa elaborado en base a la explotación de los campos, la recolección se lleva por medio de los barcos abastecedores que arriban a las plataformas transportando material, equipo y combustible.
- El tipo de barco empleado, se desplaza a una velocidad promedio de 12 nudos (22.22 kms.) por hora y supongamos que su arrendamiento es de N\$ 1,117 por hora.
- El tiempo promedio de carga en cada plataforma es de aproximadamente 3 horas, la descarga de residuos en el puerto es en promedio de 5 horas.

- Las distancias entre las plataformas y los centros de acopio se muestran en las Tablas 2.2 y 2.3.

	1	2	3	4	5	6	7	8	9	10	11
1	0	89.04	78.78	78.37	81.07	79.59	79.57	84.33	87.90	103.2	166.9
2	89.04	0	51.50	56.35	59.00	61.50	80.40	78.80	78.25	78.15	150.3
3	78.78	51.50	0	5.60	8.50	10.80	29.90	28.23	28.00	31.90	104.7
4	78.37	56.35	5.60	0	3.80	5.40	24.60	22.80	23.00	28.90	101.4
5	81.07	59.00	8.50	3.80	0	2.50	21.80	19.60	19.10	25.11	98.30
6	79.59	61.50	10.80	5.40	2.50	0	19.60	17.70	17.60	25.30	96.70
7	79.57	80.40	29.90	24.60	21.80	19.60	0	4.60	9.40	25.80	86.50
8	84.33	78.80	28.23	22.80	19.60	17.70	4.60	0	4.80	21.10	83.80
9	87.90	78.25	28.00	23.00	19.10	17.60	9.40	4.80	0	16.10	81.10
10	103.2	78.15	31.90	28.90	25.11	25.30	25.80	21.10	16.10	0	73.40
11	166.9	150.3	104.7	101.4	98.30	96.70	86.50	83.80	81.10	73.40	0

Tabla 2.1. Distancias de las plataformas a Cd. del Carmen(Kms.).

1: Cd. del Carmen, 2: Rebombero, 3: POL-A, 4: ABKATUN-N, 5: ABKATUN-D
6: ABKATUN-A, 7: NOHOCH-A, 8: AKAL-C, 9: AKAL-J, 10: KU-A, 11: Cayo Arcas.

	1	2	3	4	5	6	7	8	9	10	11
1	0	70.60	120.0	125.2	130.0	133.1	149.0	148.0	148.1	148.7	227.1
2	70.60	0	51.50	56.35	59.00	61.50	80.40	78.80	78.25	78.15	150.3
3	120.0	51.50	0	5.60	8.50	10.80	29.90	28.23	28.00	31.90	104.7
4	125.2	56.35	5.60	0	3.80	5.40	24.60	22.80	23.00	28.90	101.4
5	130.0	59.00	8.50	3.80	0	2.50	21.80	19.60	19.10	25.11	98.30
6	133.1	61.50	10.80	5.40	2.50	0	19.60	17.70	17.60	25.30	96.70
7	149.0	80.40	29.90	24.60	21.80	19.60	0	4.60	9.40	25.80	86.50
8	148.0	78.80	28.23	22.80	19.60	17.70	4.60	0	4.80	21.10	83.80
9	148.1	78.25	28.00	23.00	19.10	17.60	9.40	4.80	0	16.10	81.10
10	148.7	78.15	31.90	28.90	25.11	25.30	25.80	21.10	16.10	0	73.40
11	96.70	150.3	104.7	101.4	98.30	96.70	86.50	83.80	81.10	73.40	0

Tabla 2.3. Distancias de las plataformas a Dos Bocas(Kms.).

1: Dos Bocas, 2: Rebombero, 3: POL-A, 4: ABKATUN-N, 5: ABKATUN-D
6: ABKATUN-A, 7: NOHOCH-A, 8: AKAL-C, 9: AKAL-J, 10: KU-A, 11: Cayo Arcas.

Solución y resultados del ejemplo

Lo que se desea es conocer las rutas óptimas (trayectorias de costo mínimo), por lo que se plantearon dos problemas del agente viajero en la forma (P1) descrito anteriormente y se resolvieron usando la técnica de la búsqueda tabú con longitud tabu igual a 5 y una estructura de vecindades formada por las permutaciones hacia adelante de al menos una posición, se utilizó una PC386 sin coprocesador así como un paquete comercial donde se corrieron los métodos de ramificación y acotamiento (branch and bound (B.B)) y de la ruta del ángulo mayor (greatest angle tour (G.A.T.)), encontrándose los resultados dados en la Tabla 2.4.

Cabe observar que el número de iteraciones utilizadas por la B.T. para encontrar los valores óptimos globales fueron de 4 para el problema de Cd. del Carmen y de 9 para el problema de Dos Bocas, además para el problema de Cd. del Carmen el método de la ruta del ángulo mayor fue del 97.7% de eficiencia respecto a la solución óptima global.

Conclusiones del problema ejemplo

Como se puede observar, si se saliera de Cd. del Carmen se tendría un ahorro al año de N\$ 54279.75, pero esta diferencia no justificaría la construcción de un centro de acopio en Cd. del Carmen en el corto plazo puesto que, este tipo de obra requiere cumplir con condiciones específicas establecidas por la normatividad nacional e internacional en materia ambiental, por tanto, se deberá seguir utilizando la terminal marítima de Dos Bocas como centro de acopio, pero aplicando la ruta óptima encontrada.²

²Deseamos agradecer en especial al Ing. Alejandro González Silva así como a la A.M.I.M.E. Cd. del Carmen, y a los Ingenieros: Flores Kauffman J., Lara Vences L.M., Peralta Rodríguez J.M., Olvera Sánchez F. y Bernal González E., por su apoyo proporcionado para este ejemplo.

	Recorrido para recolección desde:	
	Cd. del Carmen	Dos Bocas
Distancia Total(Kms.)	421.21	464.15
Costo Total (N\$) a	60269.23	62530.88
B.T. Segundos	0.55	1.21
B.B. Segundos	3.0	33.61
G.A.T. Segundos	1.47	4.78

Tabla 2.4. Tabla de Totales, B.T: Búsqueda Tabú,

Capítulo 3

Recocido simulado.

El objetivo fundamental de esta sección es presentar la técnica y bondad de recocido simulado, además se presenta el problema de localización de plantas. Para tal propósito¹ se presentan tres métodos alternativos de solución del problema: un método exacto, un método heurístico y un método de búsqueda local. El método exacto y el heurístico que se seleccionaron, se reportan en la literatura como los más eficientes, vea [LOV88]. El estudio con el método de búsqueda local está enfocado a la investigación de si es bueno aceptar una solución más mala, a partir de la solución que actualmente se tenga.

El análisis comparativo de los cuatro algoritmos se hizo en cuanto a los valores que se obtienen en la función objetivo. Para cada algoritmo se reporta la experiencia computacional y en la última sección se hace un análisis comparativo de los cuatro métodos.

¹Gutierrez Andrade M. A., de los Cobos S. S. G., Pérez S. B. R., “*El problema de localización de plantas: Análisis de técnicas de solución*”, Memorias, XI Simposio Métodos Matemáticas Aplicados a las Ciencias,(Turrialba, Costa Rica, 1995) pp. 23-32.

3.1 Problema de localización de plantas.

El problema de localización de plantas tiene la siguiente estructura: hay n sitios en una región que requiere un producto. La demanda para el producto en el sitio i es d_i unidades, para $i = 1, \dots, n$. La demanda tiene que satisfacerse manufacturando el producto dentro de la región. Se necesitan m o menos plantas para manufacturar el producto que satisfaga la demanda, donde m se especifica. El costo por construir una planta en el sitio i es f_i pesos. Si una planta se construye en el sitio i , k_i unidades es la capacidad de producción.

Si existe una ruta de transporte entre el sitio i al j , k_{ij} es la capacidad sobre el horizonte de planeación y c_{ij} es el costo de transporte de una unidad entre el sitio i al j . El problema es determinar un subconjunto óptimo de sitios de localización de las plantas y un plan de transporte que minimice el costo total de construcción de plantas y el transporte de productos. El problema de determinar un subconjunto de sitios para localizar plantas y los costos fijos de transporte entre sitios es un problema de optimización combinatoria. Una vez que se conoce una solución de este problema, es decir, se ha decidido donde colocar las plantas, el problema de determinar la cantidad óptima transportada a lo largo de las rutas es un simple problema de transporte. Así, cada vez que se asignan las plantas a sitios, se debe resolver un problema de transporte para obtener la distribución óptima del producto.

Este problema puede formularse como un problema entero usando variables de decisión con las siguientes interpretaciones:

$y_i = 1$ si la planta se localiza en el sitio i .

$y_i = 0$ en otro caso.

$y_{ij} = 1$ si existe una ruta de transporte del sitio i al j .

$y_{ij} = 0$ en otro caso.

x_{ij} = cantidad (en unidades) del producto transportado del sitio i al sitio j .

Entonces el problema de localización de plantas puede formularse en términos matemáticos como:

3.2. Algoritmo exacto para el problema de localización de plantas 47

$$\min \sum_i \sum_j c_{ij} x_{ij} + \sum_i f_i y_i + \sum_i \sum_j f_{ij} y_{ij} \quad (3.1)$$

$$\begin{aligned} \sum_j x_{ij} - k_i y_i &\leq 0 \quad \forall i \\ x_{ij} - k_{ij} y_{ij} &\leq 0 \quad \forall i, j \\ \sum_i x_{ij} &\geq d_j \quad \forall j \\ \sum_i y_i &\leq m \\ x_{ij} &\geq 0 \quad \forall i, j; \quad y_i, y_{ij} = 0 \text{ o } 1 \quad \forall i, j \end{aligned}$$

El problema de localización de plantas también está clasificado dentro de la familia NP-completa y por lo tanto los algoritmos conocidos para resolver el problema de manera exacta, son de enumeración implícita o de ramificación y acotamiento.

3.2 Algoritmo exacto para el problema de localización de plantas

En esta sección se introduce un algoritmo exacto para resolver el problema de localización de plantas. Este algoritmo se reporta como una algoritmo eficiente para resolver el problema de manera exacta (vea [LOV88]) y originalmente lo propone Khumawala ([KHU72]).

3.2.1 Descripción del algoritmo exacto

El algoritmo de Khumawala es un algoritmo de ramificación y acotamiento y resuelve el siguiente problema: Sea m el número de plantas potenciales a instalar (sin límite de capacidad) y n el número de consumidores:

$$\min z = \sum_i \sum_j c_{ij} x_{ij} + \sum_i f_i y_i \quad (3.2)$$

sujeto a

$$\sum_{i \in N_j} x_{ij} = 1 \quad j = 1, 2, \dots, n,$$

$$0 \leq \sum_{j \in P_i} x_{ij} \leq n_i y_i \quad i = 1, 2, \dots, m,$$

$$y_i = 0 \text{ o } 1 \quad i = 1, 2, \dots, m,$$

donde, $c_{ij} = t_{ij} d_j$.

t_{ij} : Costo unitario del producto de la planta i al consumidor j .

d_j : demanda del consumidor j .

f_i : Costo fijo asociado a la planta i .

x_{ij} : Porcentaje de d_j abastecido desde la planta i .

y_i : Es igual a 1 o 0, dependiendo de si la planta i está abierta o cerrada.

N_j : Conjunto de almacenes que pueden abastecer al consumidor j .

P_i : Conjunto de aquellos consumidores que pueden ser abastecidos por la planta i .

n_i : Número de elementos de P_i .

Primero se resuelve el problema 3.2 de manera lineal, relajando las restricciones de integrabilidad. Sea z_0 la solución a este problema. Si las restricciones de integrabilidad se cumplen, entonces el problema está resuelto, de otra manera, se elige una variable y_k que viola estas condiciones y se formulan dos nuevos problemas. (a) La restricción $y_k = 0$ se adiciona al problema y este es nuevamente resuelto en forma lineal, sea z_1 la solución; claramente $z_1 \geq z_0$. (b) La restricción $y_k = 1$ se adiciona al problema original y este es nuevamente resuelto en forma lineal, sea z_2 la solución; claramente $z_2 \geq z_0$.

Entonces $\bar{z} = \min\{z_1, z_2\}$ es una cota inferior para z . Este procedimiento resulta en la construcción del árbol de búsqueda en donde cada una de las ramas se corta hasta que se obtiene una mejor solución del problema 3.2 o cuando la cota inferior asociada al subproblema es mayor o igual al valor de la mejor solución encontrada. El algoritmo para cuando ya no existen nodos que explorar sobre el árbol de búsqueda. En este momento, la mejor solución encontrada, es la solución óptima del problema.

En cada nodo del árbol de búsqueda existen tres conjuntos de índices, K_0 , K_1 y K_2 , donde K_0 es el conjunto de índices asociados a plantas

3.2. Algoritmo exacto para el problema de localización de plantas 49

que están abiertas en forma fija; K_1 es el conjunto de índices asociados a plantas que están cerradas en forma fija y K_2 es el conjunto de índices asociados a plantas que están libres.

La solución óptima del problema lineal en este nodo es:

$$\begin{aligned} x_{ij} &= 1 \text{ si } c_{ij} + g_i/n_i = \min_{k \in K_1 \cup K_2} [c_{kj} + g_k/n_k], \\ &= 0 \text{ en otro caso,} \\ y_i &= 0 && i \in K_0, \\ &= \sum_{j \in P_i} x_{ij}/n_i, && i \in K_2, \\ y_i &= 1, && i \in K_1, \end{aligned}$$

donde

$$\begin{aligned} g_k &= f_k, && k \in K_2, \\ &= 0, && k \in K_1. \end{aligned}$$

Existen algunas simplificaciones para reducir el número de ramas sobre el árbol de búsqueda. Son las siguientes:

1.- Cota mínima para abrir una planta. Para $i \in K_2, j \in P_i$, calcule

$$\begin{aligned} \nabla_{ij} &= \min_{k \in N_j \cap (K_1 \cup K_2); k \neq i} [\max(c_{kj} - c_{ij}, 0)], \\ \Delta_i &= \sum_{j \in P_i} \nabla_{ij} - f_i. \end{aligned}$$

si $\Delta_i > 0$, entonces $y_i = 1$ para todas las ramas que salgan del nodo. ∇_{ij} mide el costo mínimo que el consumidor j puede tener si la planta i se abre, cuando se considera con respecto a todas las plantas no cerradas en este nodo. Claramente, si la suma de este mínimo para la planta i sobre todos los consumidores que puede proveer, excede su costo fijo f_i , entonces la planta i siempre debe estar abierta en este nodo.

2.- La segunda simplificación se refiere a la reducción de n_i . Si para $i \in K_2, j \in P_i$

$$\min_{k \in K_1 \cap N_j} (c_{kj} - c_{ij}) < 0,$$

entonces n_i se reduce en uno. De hecho, si la desigualdad se cumple para toda $j \in P_i$, entonces $P_i = \emptyset$, $n_i = 0$ y $y_i = 0$ para todas las ramas que salgan de este nodo.

Claramente, si una planta que ya este abierta puede suministrar al consumidor j más barato que cualquiera de las que están libres en el nodo, entonces este consumidor debe ser abastecido por cualquiera de las plantas abiertas del nodo. Por lo tanto, este consumidor, no se debe considerar como consumidor potencial de las plantas libres en este nodo.

3.- La tercera simplificación determina una cota máxima en la reducción de costo por abrir una planta. Si esta cota es negativa, la planta debe permanecer fija en cerrada.

Para $i \in K_2, j \in P_i$

$$\omega_{i,j} = \min_{k \in K_1 \cap N_j} [\max(c_{kj} - c_{ij}, 0)],$$

$$\Omega_i = \sum_{j \in P_i} \omega_{i,j} - f_i.$$

si $\Omega_i < 0$, entonces $y_i = 0$ para todas las ramas que salgan de este nodo. La $\omega_{i,j}$ es similar a la $\nabla_{i,j}$ de la primera simplificación, excepto que aquí las comparaciones se hacen únicamente sobre todas las plantas fijas en estar abiertas; es decir, $\omega_{i,j}$ es el ahorro mínimo que puede hacerse por suministrar al consumidor j si la planta i está abierta cuando se considera sobre todas las plantas abiertas fijas en este nodo. Es claro que si la suma de estos ahorros para la planta i sobre todos los consumidores que puede proveer es menor que el costo fijo f_i , entonces esta planta debe estar cerrada y eliminarla para consideraciones adicionales.

Estas consideraciones deben hacerse a cada nodo tantas veces como sea necesario hasta que ya no se puedan hacer operaciones de cerrar o abrir plantas en forma definitiva.

3.2.2 Experiencia Computacional

La prueba del algoritmo se llevó a cabo generando de manera aleatoria instancias del problema de localización de plantas. Para valores preestablecidos del número m de plantas y el número n de consumidores, se generaron 100 instancias aleatorias del problema. Para cada

3.3. Algoritmo Heurístico para el problema de localización de plantas 51

planta se tomó como costo fijo un número aleatorio con distribución uniforme entre (100-500); para cada consumidor se tomó como costo de transporte un número aleatorio con distribución uniforme entre (40-200) y finalmente para la demanda de consumidores, se tomó un número aleatorio con distribución uniforme entre (10-100).

La Tabla 3.1 reporta el tiempo promedio (en segundos) y los valores del número de plantas y de consumidores considerados. Para cada valor del número de plantas, se consideraron cuatro casos (en cuanto al número de consumidores).

3.3 Algoritmo Heurístico para el problema de localización de plantas

El algoritmo descrito en la subsección anterior es un método exacto para la solución al problema de localización de plantas. Como todos los métodos de ramificación y acotamiento, tiene el problema que cuando el número de plantas en la instancia crece, el tamaño del árbol de búsqueda crece de manera no polinomial con respecto a este número. Esto da como consecuencia que para instancias grandes, el algoritmo no sea aplicable. Una alternativa para encontrar “buenas” soluciones al problema es la de desarrollar algoritmos heurísticos que resuelvan el problema dando una solución subóptima lo más cercana posible del óptimo. En esta subsección se describe una modificación al algoritmo heurístico propuesto por Grange (vea [SHA75]).

3.3.1 Descripción del Algoritmo heurístico

El algoritmo es sencillo y se puede describir de la siguiente manera. Suponga que el número de sitios en donde se pueden construir plantas potenciales es n y se requiere construir m o menos plantas entonces, para cada consumidor j seleccione de las n plantas potenciales que pueden cubrir su demanda, la planta que le suministre con el costo mínimo de transporte. Una vez hecho esto, se tendrá un número k de plantas abiertas.

Para cada planta abierta i , calcule su costo de eliminación; este costo de eliminación es la suma de las diferencias entre el costo menor

de suministrar su demanda por alguna otra de las $k - 1$ plantas abiertas y el costo de transporte a sus consumidores asignados. Si este costo es menor que el costo fijo f_i de la planta, entonces ciérrela y reasigne sus demandas conforme se hayan obtenido los costos menores.

Este proceso se sigue hasta que todas las plantas tengan un costo fijo menor que su costo de eliminación.

Una vez terminado este proceso se checa si $k \leq m$. Si esta condición se cumple pare con una solución subóptima. En caso contrario, seleccione la planta con el valor mínimo del costo de eliminación menos su costo fijo. Ciérrela y reasigne demanda.

Repita este proceso hasta que $k \leq m$.

3.3.2 Experiencia computacional

La prueba del algoritmo se llevó a cabo generando de manera aleatoria instancias del problema de localización de plantas. Para valores preestablecidos del número m de plantas y el número n de consumidores, se generaron 100 instancias aleatorias del problema. Para cada planta se tomó como costo fijo un número aleatorio con distribución uniforme entre (100-500); para cada consumidor se tomó como costo de transporte un número aleatorio con distribución uniforme entre (40-200) y finalmente para la demanda de consumidores, se tomó un número aleatorio con distribución uniforme entre (10-100).

La Tabla 3.2 reporta el tiempo promedio (en segundos) y los valores del número de plantas y de consumidores considerados. Para cada valor del número de plantas, se consideraron cuatro casos (en cuanto al número de consumidores).

No. Pla.	EXACTO			
	No. Arcos / Tiempo(seg.)			
	5	10	15	20
5	0.95	1.16	1.28	1.36
	10	15	25	30
10	2.20	2.68	3.93	5.05
	15	25	35	40
15	4.64	9.24	15.52	17.33
	20	25	35	40
20	15.89	17.89	21.23	30.41
	25	30	35	40
25	32.27	76.26	84.48	95.47
	30	40	50	60
30	60.14	99.55	164.97	235.60
	35	40	50	60
35	209.48	214.28	237.80	437.75
	40	45	50	60
40	396.18	436.76	1050.76	1504.63

Tabla 3.1. Tiempos promedio de ejecución del algoritmo exacto.

No. Pla.	HEURISTICO			
	No. Arcos / Tiempo(seg.)			
	5	10	15	20
5	0.01	0.01	0.02	0.02
	10	15	25	30
10	0.02	0.03	0.04	0.04
	15	25	35	40
15	0.04	0.05	0.07	0.09
	20	25	35	40
20	0.05	0.07	0.09	0.09
	25	30	35	40
25	0.07	0.10	0.12	0.13
	30	40	50	60
30	0.12	0.13	0.15	0.16
	35	40	50	60
35	0.13	0.16	0.19	0.18
	40	45	50	60
40	0.16	0.21	0.21	0.22

Tabla 3.2. Tiempos promedio de ejecución del algoritmo heurístico.

3.4 Recocido simulado

El algoritmo de recocido simulado puede usarse para atacar este problema y obtener soluciones que no necesariamente son óptimas pero se puede mostrar un comportamiento muy favorable del método con respecto a instancias que se conoce la solución.

3.4.1 El proceso de recocido de un sólido

El algoritmo de recocido simulado está basado en una analogía entre la simulación de recocido de sólidos y la problemática de resolver problemas de optimización combinatoria de gran escala. Por esta razón el algoritmo se conoce como **recocido simulado**. Recocido denota un proceso de calentamiento de un sólido a una temperatura en la

que sus granos deformados recristalizan para producir nuevos granos. La temperatura de recocido o de recristalización, depende del tipo de material, del grado de deformación del mismo, además de su uso futuro. Seguida a la fase de calentamiento, viene un proceso de enfriamiento en donde la temperatura se baja poco a poco. De esta manera, cada vez que se baja la temperatura, las partículas se acomodan en estados de más baja energía hasta que se obtiene un sólido con sus partículas acomodadas conforme a una estructura de cristal. Si se comienza con un valor máximo de la temperatura, en la fase de enfriamiento del proceso de recocido, para cada valor de la temperatura T debe permitirse que se alcance su equilibrio térmico. Sin embargo, si el proceso de enfriamiento es demasiado rápido y no se alcanza en cada etapa el equilibrio térmico, el sólido congelará en un estado cuya estructura será amorfa en lugar de la estructura cristalina de más baja energía. La estructura amorfa está caracterizada por una imperfecta cristalización del sólido.

El equilibrio térmico está caracterizado por la distribución de Boltzmann [TOD83]. De acuerdo a esta distribución, la probabilidad que el sólido esté en un estado i con energía E_i a la temperatura T , viene dada por

$$P_T\{X = i\} = \frac{1}{Z(T)} \exp\left(\frac{-E_i}{k_B T}\right), \quad (3.3)$$

donde X es una variable aleatoria que denota el estado actual del sólido. $Z(T)$ es una constante de normalización llamada la **función partición**, que esta definida como

$$Z(T) = \sum_j \exp\left(\frac{-E_j}{k_B T}\right), \quad (3.4)$$

donde la sumatoria se extiende sobre todos los posibles estados y k_B es una constante física conocida como la constante de Boltzmann. El factor $\exp\left(\frac{-E_j}{k_B T}\right)$ se conoce como el factor de Boltzmann. Obviamente 3.3 es una función de densidad de probabilidad ya que siempre es mayor o igual a cero y la suma sobre todos los valores es igual a la unidad. Se puede observar en 3.3 que cuando el valor de T disminuye, la distribución de Boltzmann se concentra en los estados de menor energía mientras que si la temperatura se aproxima a cero, únicamente los esta-

dos con mínima energía tienen una probabilidad de ocurrencia diferente de cero.

Por lo dicho anteriormente, el proceso de recocido consta de dos pasos fundamentales que son:

- Incrementar la temperatura del baño térmico a un valor máximo.
- Decrementar **cuidadosamente** la temperatura del baño térmico hasta que las partículas se reacomoden por sí mismas en un estado de mínima energía, denominado el estado fundamental del sólido.

El proceso físico de recocido puede modelarse exitosamente usando métodos de simulación (vea [BIN78] o [MET53]), el algoritmo introducido para tal propósito se basa en técnicas Monte Carlo y genera una sucesión de estados del sólido de la siguiente manera. Dado un estado i del sólido con energía E_i , se genera un estado subsecuente j aplicando un mecanismo de perturbación que transforma el estado actual en el siguiente estado por medio de una pequeña distorsión, por ejemplo, por el desplazamiento de una partícula. La energía del siguiente estado es E_j . Si la **diferencia de energía**, $E_j - E_i$, es menor o igual a cero, el estado j se acepta como el estado actual. Si la diferencia de energía es mayor que cero, el estado j se acepta con una probabilidad que está dada por

$$\exp\left(\frac{E_i - E_j}{k_B T}\right), \quad (3.5)$$

donde T denote la temperatura del baño térmico y k_B es la constante de Boltzmann. La regla de decisión descrita arriba se conoce como el **criterio de Metropolis** y al algoritmo se le conoce como **algoritmo de Metropolis**.

Si la temperatura se baja poco a poco, el sólido puede alcanzar su equilibrio térmico en cada temperatura. En el algoritmo de Metropolis esto se lleva a cabo generando un número grande de transiciones para un valor dado de la temperatura.

3.4.2 El Algoritmo de Recocido Simulado

La simulación del proceso de recocido puede usarse para describir un proceso de generación de sucesiones de soluciones de un problema de optimización combinatoria en donde se vaya obteniendo, conforme el proceso avanza, mejores soluciones al mismo. Para este propósito, se puede observar una analogía entre el sistema físico y un problema de optimización combinatoria en donde cada solución del problema de optimización combinatoria puede verse como un estado del sólido y los valores de la función objetivo para cada solución del problema de optimización combinatoria como los niveles de energía del sólido. En resumen, se puede pensar en las siguientes equivalencias.

- Las soluciones de un problema de optimización combinatoria son equivalentes a los estados de un sistema físico.
- El costo de una solución es equivalente a la energía de un estado.

Además, se introduce un parámetro que juega el papel equivalente de la temperatura. Este parámetro se llama el **parámetro de control**. El algoritmo de recocido simulado puede verse como una iteración del algoritmo de Metropolis, evaluado en valores decrecientes del parámetro de control.

Como en los algoritmos de búsqueda local se supone la existencia de una estructura de vecindades y un mecanismo de generación. Se introducen las siguientes definiciones.

Definición 3.1 Sea (S, f) denote una instancia de un problema de optimización combinatoria y denote por i y j dos soluciones con costo $f(i)$ y $f(j)$, respectivamente. Entonces el criterio de aceptación determina si j se acepta de i a partir de aplicar la siguiente probabilidad de aceptación:

$$P_c\{\text{aceptar } j\} = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ \exp\left(\frac{f(i)-f(j)}{c}\right) & \text{si } f(j) > f(i), \end{cases} \quad (3.6)$$

donde $c \in R^+$ denota el parámetro de control.

Claramente, el mecanismo de generación corresponde al mecanismo de perturbación en el algoritmo de Metropolis, mientras que el criterio de aceptación corresponde al criterio de Metropolis.

Definición 3.2 *Una transición es una acción combinada que transforma la solución actual en una subsecuente. Esta acción consiste de los siguientes dos pasos: (i) aplicación del mecanismo de generación, (ii) aplicación del criterio de aceptación.*

Sea c_k denote el valor del parámetro de control y L_k el número de transiciones generadas en la k -ésima iteración del algoritmo de Metropolis. Entonces el algoritmo de recocido simulado puede describirse en pseudo-código como se muestra en la Figura 3.1

El algoritmo de la Figura 3.1 comienza llamando a un procedimiento de inicialización donde se definen la solución inicial, la temperatura inicial y el número inicial de generaciones necesarias para alcanzar el equilibrio térmico para la temperatura inicial. La parte medular del algoritmo consta de dos ciclos. El externo **Repite...hasta** y el interno **para...finpara**. El ciclo interno mantiene fijo el parámetro de control hasta que se generan L_k soluciones y se acepta o se rechaza la solución generada conforme los criterios de aceptación ya discutidos. El ciclo externo disminuye el valor de la temperatura mediante el procedimiento **CALCULA-CONTROL** y calcula el número de soluciones a generar para alcanzar equilibrio térmico mediante el procedimiento **CALCULA-LONGITUD**. Este ciclo finaliza cuando la condición de paro se cumple.

Un rasgo característico del algoritmo de recocido simulado es que, además de aceptar mejoramientos en el costo, también acepta soluciones más malas en costo. Inicialmente, para valores grandes de c , puede aceptar soluciones con un valor objetivo mucho mayor a la solución actual; cuando c decrece, únicamente pequeñas desviaciones serán aceptadas y finalmente, cuando el valor de c se aproxima a cero, no se aceptarán desviaciones. Este hecho significa que el algoritmo de recocido simulado, en contraste con el algoritmo de búsqueda local, puede escapar de mínimos locales además de exhibir los rasgos favorables de los algoritmos de búsqueda local; es decir, simplicidad y aplicabilidad general.

Note que la probabilidad de aceptar desviaciones está implementada al comparar el valor de $\exp((f(i) - f(j))/c)$ con un número aleatorio generado de una distribución uniforme en el intervalo $[0, 1)$. Además, debe ser obvio que la velocidad de convergencia del algoritmo está determinada al escoger los parámetros L_k y c_k , $k=0,1,\dots$. Si los valores c_k decrecen rápidamente o los valores de L_k no son grandes, se tendrá una convergencia más rápida que cuando los valores de c_k decrecen lentamente o los valores de L_k son grandes.

ComienzaINICIALIZA ($i_{inicial}, c_0, L_0$) $k := 0$ $i := i_{inicial}$ **Repite**para $l := 1$ a L_k haz**Comienza**GENERA (j de S_i)si $f(j) \leq f(i)$ entonces $i := j$

sino

si $\exp\left(\frac{f(i)-f(j)}{c_k}\right) >$ numero aleatorio en $[0, 1)$ entonces $i := j$ **finpara** $k := k + 1$ CALCULA-LONGITUD (L_k)CALCULA-CONTROL (c_k)

hasta criterio de paro

fin

Figura 3.1 Algoritmo de Recocido Simulado en pseudo-código

Comparando el algoritmo de recocido simulado con el algoritmo de búsqueda local, resulta evidente que recocido simulado puede verse como una generalización de búsqueda local y viene a ser idéntico a búsqueda local, en el caso que el parámetro de control c se tome igual a cero. En lo que respecta a la ejecución de ambos algoritmos, generalmente el algoritmo de recocido simulado obtiene soluciones de mejor calidad que el algoritmo de búsqueda local.

3.4.3 Aspectos generales del algoritmo

En las aplicaciones del algoritmo de recocido simulado comúnmente se desea implementarlo de manera que la sucesión de soluciones estén generadas a partir de valores decrecientes del parámetro de control. Las soluciones se generan continuamente tratando de transformar la solución actual en una subsecuente por medio de aplicar los mecanismos de generación y el criterio de aceptación. Las aplicaciones del algoritmo de recocido simulado requieren de la especificación de los siguientes puntos: (i) Una representación concisa del problema, (ii) un mecanismo de transición y (iii) un programa de enfriamiento. Cada uno de estos puntos se pueden enumerar con más detalle.

1. Una descripción concisa de la **representación del problema** consiste de una representación del espacio de soluciones y una expresión de la función de costo. La función de costo debe escogerse de manera que represente la efectividad de las soluciones con respecto al objetivo de optimización. La representación del problema y la función de costo deben darse por medio de expresiones simples que permitan una fácil manipulación.
2. La generación de ensayos para transformar la solución actual en una subsecuente consiste de tres pasos. Primero, se debe generar una nueva solución aplicando un mecanismo de generación. Enseguida se debe calcular la diferencia de costo de las dos soluciones, por último, se hace una decisión de aceptar o no, la nueva solución.

La evaluación de la nueva solución es lo que más tiempo consume en el algoritmo de recocido simulado y por lo tanto debe hacerse lo más eficientemente posible. El mecanismo de generación usualmente se escoge de tal manera que se obtengan modificaciones simples para que puedan ser ejecutadas rápidamente, por ejemplo permutaciones, cambios o inversiones. El cálculo de la diferencia de costo se hace tomando en cuenta las diferencias entre ambas soluciones. Para muchas aplicaciones éste es el camino más rápido para calcular las diferencias en el costo. La decisión de aceptar la nueva solución se basa en un criterio de aceptación (vea definición 3.1).

3. Ejecutar el proceso de recocido, requiere de la especificación de los parámetros que determinan el programa de enfriamiento. Estos parámetros son el valor inicial del parámetro de control, una función que especifique el decremento del parámetro de control, la longitud de cada bloque donde permanece constante el parámetro de control y el criterio de paro.

En [AAR89] se muestra que si se selecciona el mecanismo de generación, el criterio de aceptación como en la definición 3.1, una estructura de vecindades y un programa de enfriamiento adecuado; el algoritmo de recocido simulado converge con probabilidad 1 al conjunto de soluciones óptimas.

3.4.4 Descripción del Algoritmo

En este caso el algoritmo de recocido simulado puede aplicarse de la siguiente manera:

- El espacio de soluciones \mathcal{S} , para el problema combinatorio, está representado por todos los vectores binarios (y_1, y_2, \dots, y_n) tal que $y_i = 0$ o 1 y $\sum_{i=1}^n y_i \leq m$. Donde $y_i = 0$ si no se coloca planta en el sitio i y $y_i = 1$ si se coloca planta en el sitio i .
- La función de costo a minimizar, se escoge como:

$$f = \sum_i \sum_j c_{ij} x_{ij} + \sum_i f_i y_i \quad (3.7)$$

donde x_{ij} es la cantidad (en unidades) del producto transportado del sitio i al sitio j y f es el costo de la localización más transporte.

- Las nuevas soluciones se generan a partir de dos mecanismos: uno que selecciona aleatoriamente

un sitio i , si existe planta en el sitio i se quita y si no existe, entonces se coloca una planta. El otro mecanismo genera aleatoriamente un sitio i , si existe una planta en i se cambia a otro sitio j donde no exista, tomado en forma aleatoria y si no existe planta en el sitio i , se busca aleatoriamente un sitio j donde exista una planta y se intercambia.

La selección de uno u otro mecanismo se hace de manera aleatoria. Con probabilidad p se hace intercambio de plantas y con probabilidad $1 - p$ se abre una planta o se cierra. El valor de p puede variarse dependiendo del parámetro de control.

- La actualización de la función de costo entre la solución actual y la nueva solución puede hacerse fácilmente en la parte de costos fijos únicamente al quitar, agregar o intercambiar los costos de plantas que se den de alta y de baja. La parte del costo de transporte debe calcularse para cada nueva configuración usando la solución actual y modificando los suministros variados por medio de penalizaciones.

El algoritmo que describe el procedimiento de recocido puede enunciarse como aparece abajo.

Algoritmo de Recocido Simulado para el Problema de Localización de Plantas

Propósito: Obtener los sitios donde colocar las plantas y la asignación de plantas a consumidores.

Descripción:

Paso 1: Tome c un valor grande apropiado.

Paso 2: Asigne valores 0 o 1 a y_i con $i \in \{1, 2, \dots, n\}$ de manera arbitraria.
Obtenga el valor correspondiente de la función objetivo f .

Paso 3: Asigne el valor verdadero a la variable lógica CHANGE.

Paso 4: Si la variable CHANGE tiene valor falso, entonces pare.

Paso 5: Haga CHANGE igual a falso.

Paso 6: Repita los puntos 1-11 r veces.

- 1.- Seleccione aleatoriamente un número $u \in \{1, 2, \dots, n\}$.
- 2.- Genere un número aleatorio x , uniformemente distribuido en $(0, 1)$.
- 3.- Si $x \geq p$ abra (o cierre) la planta u , en caso contrario intercámbiela por una cerrada (abierta).
- 4.- Evalúe el valor Δf .
- 5.- Si $\Delta f < 0$ entonces vaya a 9.
- 6.- Sea $P(\Delta f) = \exp(-\Delta f/c)$.
- 7.- Genere un número aleatorio x , uniformemente distribuido en $(0, 1)$.
- 8.- Si $x \geq P(\Delta f)$ entonces vaya a 11.

9.- Acepte el cambio hecho y reactualice el valor de f .

10.- Si $\Delta f \neq 0$ entonces CHANGE — verdadero.

11.- Fin del paso 6.

Paso 7 Reduzca c e incremente p y r .

Paso 8 Vaya al paso 4.

3.4.5 Experiencia computacional

La prueba del algoritmo se llevó a cabo generando de manera aleatoria instancias del problema de localización de plantas. Para valores preestablecidos del número m de plantas y el número n de consumidores, se generaron 100 instancias aleatorias del problema. Para cada planta se tomó como costo fijo un número aleatorio con distribución uniforme entre (100-500); para cada consumidor se tomó como costo de transporte un número aleatorio con distribución uniforme entre (40-200) y finalmente para la de manda de consumidores, se tomó un número aleatorio con distribución uniforme entre (10-100).

La Tabla 3.3 reporta el tiempo promedio (en segundos) y los valores del número de plantas y de consumidores considerados. Para cada valor del número de plantas, se consideraron cuatro casos (en cuanto al número de consumidores).

3.5 Algoritmo de búsqueda local

Uno de los puntos importantes del algoritmo de recocido simulado es el hecho de que cada que se ejecuta el **Paso 6** del mismo, no solamente acepta soluciones con menor costo sino que también se aceptan con cierta probabilidad soluciones con mayor costo. Esta situación cada vez es menos frecuente conforme el valor del parámetro de control c se decrementa. Sin embargo queda en duda el hecho de si el aceptar soluciones “malas” da algún beneficio. Con el objeto de poder comparar los resultados del algoritmo de recocido simulado con el algoritmo de búsqueda local correspondiente, también se implantó este último en la computadora.

No. Pla.	RECOCIDO SIMULADO			
	No. Arcos / Tiempo(seg.)			
	5	10	15	20
5	0.01	0.78	0.75	0.83
	10	15	25	30
10	2.04	2.73	2.20	2.67
	15	25	35	40
15	4.82	5.82	9.15	9.55
	20	25	35	40
20	6.92	7.30	13.73	10.34
	25	30	35	40
25	12.85	10.83	12.46	12.58
	30	40	50	60
30	12.53	12.74	11.92	13.44
	35	40	50	60
35	18.66	26.87	27.80	29.86
	40	45	50	60
40	20.33	30.54	32.75	32.32

Tabla 3.3. Tiempos promedio de ejecución del algoritmo de recocido simulado.

No. Pla.	BÚSQUEDA LOCAL			
	No. Arcos / Tiempo(seg.)			
	5	10	15	20
5	0.00	0.07	0.07	0.08
	10	15	25	30
10	0.21	0.23	0.22	0.27
	15	25	35	40
15	0.42	0.58	0.93	0.95
	20	25	35	40
20	0.69	0.73	1.53	1.04
	25	30	35	40
25	1.15	1.18	1.58	1.66
	30	40	50	60
30	1.25	1.26	1.92	1.97
	35	40	50	60
35	1.17	2.77	2.78	2.96
	40	45	50	60
40	2.05	3.17	3.25	4.49

Tabla 3.4. Tiempos promedio de ejecución del algoritmo de búsqueda local.

Siguiendo la misma notación que la ya dada para el algoritmo de recocido simulado y tomando en cuenta que si en el algoritmo de recocido simulado se toma el parámetro de control c desde el principio cercano a cero, el algoritmo que nos queda es un algoritmo de búsqueda local, ya que, la probabilidad de aceptar soluciones más malas que la actual prácticamente se reduce a cero.

3.5.1 Experiencia computacional

La prueba del algoritmo se llevó a cabo generando de manera aleatoria instancias del problema de localización de plantas. Para valores preestablecidos del número m de plantas y el número n de consumidores, se generaron 100 instancias aleatorias del problema. Para cada planta se tomó como costo fijo un número aleatorio con distribución

uniforme entre (100-500); para cada consumidor se tomó como costo de transporte un número aleatorio con distribución uniforme entre (40-200) y finalmente para la de manda de consumidores, se tomó un número aleatorio con distribución uniforme entre (10-100).

La Tabla 3.4 reporta el tiempo promedio (en segundos) y los valores del número de plantas y de consumidores considerados. Para cada valor del número de plantas, se consideraron cuatro casos (en cuanto al número de consumidores).

3.6 Análisis comparativo de los métodos

Como ya se mencionó, se generaron para cada valor fijo de m y n 100 instancias en forma aleatoria del problema de localización de plantas. Para cada una de estas instancias generadas, se corrieron los cuatro algoritmos y se calculó el tiempo promedio de solución. Los resultados aparecen en las Tablas 3.1, 3.2, 3.3 y 3.4.

Como se observó, el tiempo de solución del algoritmo de recocido simulado es muy variable, ya que depende en gran medida del valor inicial del parámetro de control c , del número r de veces que se ejecuta el paso 6 (del algoritmo) y de la forma en que se reduzca el parámetro c y se incrementen los parámetros p y r . Para el análisis realizado en este documento, se tomó un valor inicial de c de tal manera que la tasa de aceptación de las soluciones propuestas fuera mayor o igual a 0.95 y los decrementos fueran del 0.1 (paso 7 del algoritmo). El número r inicial que se tomó en cada caso fue $r = n$ y los incrementos fueron de 0.05 (paso 7 del algoritmo). Para el cálculo la eficiencia de

cada uno de los algoritmos con respecto al valor óptimo de la función objetivo Los cálculos que aparecen en la Tabla 3.5 son con base en

los promedios obtenidos en las 100 instancias resultas para cada valor del número m de plantas y del número n de consumidores. En la

Tabla 3.5, puede observarse que las eficiencias del algoritmo de recocido simulado son mejores que las de los otros dos algoritmos heurísticos y que el comportamiento de esta eficiencia se mantiene conforme crecen

los valores del número de plantas y del número de consumidores. Las pruebas computacionales con el problema de localización de plantas revelen un excelente comportamiento del algoritmo. Las soluciones subóptimas difieren en 1-2% de la solución óptima. Además si se ejecuta el procedimiento varias veces con diferentes valores iniciales, se obtienen en todos los casos la solución. Los programas computacionales y las corridas de los mismos; se realizaron en una computadora IBM-PS/70 de 25 mhz con procesador 80386 y sin coprocesador matemático.

m	n	ALGORITMO (eficiencia)			
		EXACTO	HEURISTICO	REC. SIM.	BUS. LOC.
5	5	1.00	0.9504	0.9899	0.9051
	10	1.00	0.9490	0.9966	0.8965
	15	1.00	0.9377	0.9989	0.8679
	20	1.00	0.9371	0.9977	0.8587
10	10	1.00	0.8440	0.9962	0.9058
	15	1.00	0.8563	0.9929	0.8526
	25	1.00	0.8920	0.9957	0.8671
	30	1.00	0.9140	0.9912	0.8623
15	15	1.00	0.8033	0.9924	0.8973
	25	1.00	0.8655	0.9985	0.8631
	35	1.00	0.8690	0.9963	0.8651
	40	1.00	0.8662	0.9796	0.8249
20	20	1.00	0.7971	0.9919	0.8450
	25	1.00	0.8042	0.9847	0.8357
	35	1.00	0.7982	0.9863	0.8126
	40	1.00	0.8243	0.9895	0.7991
25	25	1.00	0.7405	0.9911	0.8552
	30	1.00	0.7959	0.9884	0.8696
	35	1.00	0.8079	0.9943	0.8229
	40	1.00	0.7659	0.9886	0.8027
30	30	1.00	0.7586	0.9891	0.8222
	40	1.00	0.7786	0.9920	0.8149
	50	1.00	0.8056	0.9881	0.7604
	60	1.00	0.8048	0.9866	0.7669
35	35	1.00	0.7753	0.9904	0.8049
	40	1.00	0.7516	0.9906	0.7982
	50	1.00	0.7755	0.9898	0.7890
	60	1.00	0.7755	0.9923	0.7883
40	40	1.00	0.7234	0.9857	0.7814
	45	1.00	0.7945	0.9943	0.7886
	50	1.00	0.7676	0.9917	0.7699
	60	1.00	0.7578	0.9862	0.7621

Tabla 3.5. Comparación de los algoritmos que resuelven el problema de localización de plantas con respecto a su eficiencia.

Capítulo 4

Algoritmos genéticos

“La vida empezó como el resultado de un experimento a gran escala sobre la Tierra, llamado evolución biológica. Esto empezó al rededor de cuatro mil millones de años en el Oceano primitivo en la forma de una masa viviente. De aquí, los peces se desarrollaron. Tiempo más tarde la vida se arrastro sobre la tierra. Lo siguiente, nuestros ancestros colgaron de los árboles. Y finalmente nos observamos a nosotros mismos: El hombre meditaba de cómo el método de la evolución trabaja.” Rechenberg (1989).

Vale la pena reflexionar sobre la efectividad de la estrategia de la evolución biológica. En años anteriores, llegaba a ser obvio que las reglas de la evolución biológica son el resultado de un proceso de evolución en sí mismas. Suponga una población de organismos existentes con reglas hereditarias escasamente modificadas con la norma existente. Si esas modificaciones ayudan a la población para adaptarse más rápidamente a su medio ambiente particular, entonces esta población tiene mayor oportunidad de sobrevivir en el futuro que una población con menos reglas hereditarias efectivas. Por tanto se puede asumir que la evolución, durante su acción de más de tres mil millones de años, da en sí misma un modelo optimal de operación. Al final, la estrategia de la evolución es un modo optimal que ha desarrollado fascinantes sistemas incluyendo al cerebro humano. Esto origina la

idea que los conceptos estratégicos de la evolución biológica pueden utilizarse con éxito para crear máquinas "inteligentes".

Una idea general de los algoritmos genéticos (AG), es que se pueden ver como una búsqueda aleatoria inteligente, en el sentido de obtener un muestreo aleatorio del espacio de soluciones.

Los algoritmos genéticos fueron desarrollados por Holland en la Universidad de Michigan por los 60's. El nombre de algoritmos genéticos se origina de la analogía entre la estructura genética de los cromosomas y de la representación de las soluciones de problemas por medio de vectores.

4.1 Algunos experimentos.

Fue en 1964 cuando Rechenberg (1989) realizó su experimento para imitar el método de la evolución biológica en un laboratorio de mecánica de fluidos. Un disco de aluminio flexible en cinco posiciones fue montado en un tunel de aire. El disco articulado podía alterarse por pasos. Existen 345,025,251 formas posibles. La tarea era el de encontrar la forma con resistencia al avance mínima. Suponiendo que no se conocía cual era, el disco era puesto en una configuración inicial aleatoria. Para producir las alternaciones aleatorias de las cinco bisagras (las mutaciones en Biología), se usaba en este primer experimento un aparato mecánico, en donde cinco pelotitas, representando a las cinco bisagras del disco, pasaban de manera aleatoria hacia diferentes cajas. La caja marcada determinaba el ángulo de alternación.

El arreglo experimental hace posible medir la conveniencia de las formas mutadas. La conveniencia técnica es la resistencia al avance del disco, la cual tiene que llegar a ser mínima. Al inicio del experimento del tunel de aire el disco fue plegado de una forma de zig zag de alta resistencia al avance. El esquema experimental para imitar las reglas de la evolución biológica se discutirán más adelante. La idea básica es rechazar todas las mutaciones con resistencia al avance creciente (conveniencia decreciente). Pero si una forma generada aleatoriamente

tiene una resistencia al avance pequeña, entonces se tiene la forma nodriza para la siguiente generación. La resistencia al avance se midió contra el número de generaciones. Encontrando que la forma plana se tenía después de 300 generaciones.

El término técnico de *estrategia de evolución* para un programa, tiene el objetivo de copiar las reglas de la evolución biológica en la forma más exacta. Así de lejos, la imitación de la evolución de un modo simplificado ha probado ser muy exitoso. Con el tiempo se ha llegado a describir diferentes estilos de operaciones de evolución.

4.2 Componentes básicos.

Los algoritmos genéticos (AG) están basados en la noción de la propagación de nuevas soluciones a partir de *soluciones padres*, empleando mecanismos modelados de aquellos que se aplican en genética. La mejor descendencia de las soluciones padres se retiene para una nueva generación, por lo que al proceder en forma de evolución se fomenta la *supervivencia de los más aptos*. Como la calidad de los más aptos (mejor descendencia) eventualmente construye el más alto nivel compatible con el medio ambiente (el problema es gobernado por restricciones), la mejor sobre todas las descendencias se registra y es el candidato propuesto por el método para una solución optimal.

Un algoritmo genético básico tiene tres operadores: *reproducción*, *cruzamiento* y *mutación*. La reproducción es un apareamiento aleatorio de individuos (soluciones muestra) de una población para crear una o más descendencias. El cruzamiento define el resultado como un cambio de gene (plan reproductivo), cuyo valor específico es conocido como *allele*, *i.e.*, los alleles pueden concebirse como instancias en el sentido de sistemas expertos. El cambio de genes (tipo de información y sus atributos) siguen las reglas posicionales tradicionalmente modelada después de la reproducción biológica. Deben modificarse y particularizarse a diferentes tipos de problemas combinatorios, ahora bien, para que tenga sentido se debe permitir ciertas relaciones de restricciones para proporcionar una oportunidad al progénito de realizar un

mejoramiento sobre sus padres.

Finalmente, una mutación es simplemente el introducir un elemento aleatorio, muchas veces usado para enmendar el resultado de un gene cambiado cuando el resultado no se encuentra exitoso bajo las restricciones apropiadas. A este respecto, la mutación está más fuertemente sesgada para ayudar en la aplicación de los algoritmos genéticos que en la genética biológica. La mutación diversifica el espacio de búsqueda y protege de la pérdida de material genético que puede darse en la reproducción y el cruzamiento.

En términos generales podemos considerar que los AG son técnicas de búsqueda aleatoria que imitan a los procesos observados en la evolución natural. Combinan la supervivencia de los más aptos (o mejores) entre estructuras de anillos (soluciones) con un estructurado cambio de información aún aleatorizado.

Los algoritmos genéticos difieren de las técnicas de optimización tradicionales en muchos aspectos. Ellos trabajan con una codificación de las variables (típicamente como anillos) más que con las variables en sí mismas, y utilizan las reglas de transición probabilística para moverse de una población de soluciones a otras, más que de una solución sencilla a otra. La más interesante e importante característica de los AG es que ellos utilizan tan solo evaluaciones de la función objetivo. Esto es, los AG no utilizan ninguna información sobre diferenciabilidad, convexidad o alguna otra característica auxiliar. Esto hace que los AG sean fáciles de utilizar y de implantar en gran variedad de problemas de optimización.

La aplicación de los algoritmos genéticos a problemas de investigación de operaciones se ha limitado debido a los dominios factibles complejos. Dado un problema de optimización, frecuentemente el paso más difícil en la aplicación del AG es el codificar las soluciones como anillos de forma tal que el apareamiento de soluciones factibles de por resultado soluciones factibles.

Las técnicas para la codificación de soluciones varían dado el tipo

de problema, en la mayoría de los casos, involucra cierta cantidad de arte. Por lo general para problemas enteros 0-1, las soluciones se representan típicamente por medio de una cadena de bits de longitud igual al número de variables en el problema, digamos n , donde cada posición de la cadena puede tomar el valor 0 ó 1, (en terminología genética) cada posición es un gene y el valor en esa posición es un allele. Esto resulta en un espacio de búsqueda de cardinalidad 2^n .

Existen muchas variaciones de los algoritmos genéticos debido al uso de diferentes operadores de reproducción, cruzamiento y mutación. A continuación se proporciona una estructura general de este procedimiento dado por Hadj-Alouane y Bean (1993):

0.- Proporcione una generación inicial.

1.- *Copie* las N_c soluciones superiores.- Las soluciones de la generación actual son ordenadas de manera creciente respecto del valor de la función objetivo y entonces las N_c soluciones superiores se copian dentro de la siguiente generación, este procedimiento se llama reproducción.

2.- *Cruzamiento* aleatorio de parejas. Primero seleccione aleatoriamente dos cadenas (padres) de la generación actual. Ahora, cree dos descendencias como sigue: Para cada gene, una moneda sesgada se lanza, y el resultado determina cuando se realiza o no se realiza el intercambio de los allele de los padres. Formalmente, esta operación puede describirse de la siguiente manera. Considere los anillos padres $P_1 = p_{11}p_{12} \dots p_{1m}$ y $P_2 = p_{21}p_{22} \dots p_{2m}$, donde p_{ij} es el allele del j -ésimo gene de la cadena P_i . Similarmente, sean $O_1 = o_{11}o_{12} \dots o_{1m}$ y $O_2 = o_{21}o_{22} \dots o_{2m}$ las descendencias creadas al cruzar a P_1 y a P_2 . Observe que aquí, las letras en negrillas se utilizan dado que se denota al gene y no al allele. Entonces el intercambio probabilístico de los m allele puede modelarse como m variables aleatorias independientes, digamos X_1, X_2, \dots, X_m , donde cada

una de ellas tiene una distribución Bernoulli con parámetros diferentes de 0.5.

Esas variables aleatorias se definen como sigue:

Para $j = 1, \dots, m$,

$$X_j = \begin{cases} 1 & \text{si los alleles } p_{1j} \text{ y } p_{2j} \text{ se intercambian,} \\ 0 & \text{de otra forma.} \end{cases}$$

Para realizarse esos m ensayos independientes, los genes de las dos descendencias se construyen. Los genes de la descendencia son funciones de las variables aleatorias X_j^s , dadas como sigue:

Para $j = 1, \dots, m$,

$$\begin{aligned} o_{1j} &= (1 - X_j)p_{1j} + X_j p_{2j}, \\ o_{2j} &= X_j p_{1j} + (1 - X_j)p_{2j}. \end{aligned}$$

Una vez que las descendencias se crean, se evalúan, y sólo la que tiene mejor valor de la función objetivo se incluye en la nueva generación.

3.- Crear *mutación* mediante la generación aleatoria de un pequeño número de soluciones completamente nuevas e inclúyalas dentro de la nueva generación. Una solución aleatoria consiste de una cadena de m números aleatorios, digamos $r_1 r_2 \dots r_m$ donde r_i está uniformemente distribuido en el intervalo $[1, n_i]$, para toda $i = 1, 2, \dots, m$. Esta operación es diferente de la mutación de gene por gene dado que involucra el traer nuevos miembros a la población.

En general, un algoritmo genético para resolver un problema combinatorio debe contener 5 componentes principales (Davis y Steenstrup (1987)):

1. Una representación cromosómica de las configuraciones del espacio de búsqueda del problema.
2. Una forma de generar la población inicial de soluciones.

3. Una función de evaluación que juega el papel del medio ambiente, calificando las configuraciones en términos de su "valor de ajuste".
4. Una descripción precisa de las operaciones genéticas que alteran la composición de los "hijos" durante la reproducción.
5. Los valores de los parámetros que el algoritmo genético utiliza (tamaño de la población, probabilidades de aplicar las operaciones genéticas, etc.)

Holland y Golberg, desarrollan el concepto de *esquema* o patrón de similaridad para analizar el problema de convergencia para algoritmos que trabajan con representaciones de cadenas de bits. Un esquema es un patrón en el que ciertas posiciones del cromosoma son fijas. Por ejemplo, el esquema $1*0**1$, comprende a todos los cromosomas cuyo primer allele es 1, tercer allele es 0 y su sexto allele es 1. Note que * representa posiciones en que no estamos interesados y que pueden tomar cualquier valor del alfabeto genético utilizado. A través de la noción de esquemas el teorema fundamental de los algoritmos genéticos, proporciona una cota inferior a la evaluación para un tipo de esquema en una generación, por lo que, se pueden mantener "buenos" esquemas después de un cierto número de generaciones. Esta cota depende del número de posiciones fijas del esquema, de la distancia entre las posiciones fijas y del criterio de evaluación de la generación anterior.

Capítulo 5

Análisis de objetivos

El análisis de objetivos (AO) es un procedimiento para desarrollar métodos de mejora para resolver problemas. Laguna y Glover (1993) comentan que una aplicación inicial fue la calendarización de la producción futura de energía de una planta nuclear, donde se descubrió la ventaja de incorporar un componente de inteligencia artificial dentro de la técnica de optimización estándar, *i.e.*, se puede considerar que el (AO) es una integración de la inteligencia artificial con la investigación de operaciones que proporciona una nueva estrategia para resolver problemas de optimización combinatoria mediante el análisis retrospectivo y técnicas de reconocimiento de patrones (*percepción tardía* Glover y Greenberg (1989)), proporcionando a los procedimientos de solución optimal o a heurísticas la habilidad para *aprender* qué reglas son las mejores para resolver una clase particular de problemas.

El análisis de objetivos proporciona un medio para explotar una "visión" amplia, junto con los beneficios de las evaluaciones de los movimientos. En particular, la incorporación de un proceso sistemático de diversificación en la búsqueda proporciona, a largo plazo, información útil para crear una función de evaluación dinámica.

Laguna y Glover (1993) proporcionan las características y pasos principales del análisis de objetivos que son:

- Fase 1.- Se aplican métodos existentes a determinadas soluciones optimales o de alta calidad a problemas repre-

sentativos de una clase dada.

- Fase 2.- Se utilizan las soluciones generadas en la fase 1 como *objetivos*, los que llegan a ser el foco de un nuevo conjunto de *soluciones transmitidas*. Durante esa transmisión, cada problema se vuelve a resolver, con los siguientes propósitos:

- Evaluar la habilidad de las reglas de decisión actuales para identificar “buenos” movimientos y determinar regiones donde esas reglas están operando de manera eficiente.
- Marcar a los movimientos posibles en cada iteración para elección sesgada para seleccionar movimientos con altos puntajes.
- Generar información que pueda utilizarse para inferir los puntajes.

- Fase 3.- Construcción de funciones parametrizadas con la información registrada en la fase 2, con el objetivo de encontrar valores de los parámetros para crear una *regla de decisión maestra*. Esta regla puede también realizarse mediante el integrar funciones que complementan á la regla actual en regiones donde ésta falla para seleccionar movimientos de alto puntaje. En general, el objetivo debe ser el mejorar la habilidad general del método para realizar elecciones “correctas” en cada paso de solución.

- Fase 4.- Generación de un modelo matemático o estadístico (tal como el de la programación de metas generalizada o el modelo de análisis de discriminantes) para determinar los valores de los parámetros efectivos para la regla de decisión maestra.

- Fase 5.- Se concluye el proceso al aplicar la regla de decisión maestra a los problemas representativos originales y a otros problemas de la clase de solución elegida para confirmar su mérito.

En Glover y Greenber (1989) se proporciona el siguiente ejemplo: Suponga una estrategia de enumeración implícita que tiene una estrategia de ramificación hacia adelante donde se elige una *mejor rama* de la forma $x = b$, donde x es una elección de variable y b es una elección de su valor fijo (donde b es 0 ó 1). Suponga que un cierto criterio, digamos k se utiliza en la realización de la ramificación mediante la elección de los valores con pesos lineales w_1, w_2, \dots, w_k . Por simplicidad, suponga que esta rama (x, b) se eligió porque:

$$\sum_i w_i F_i(x, b) = \min[\sum_i w_i F_i(\chi, \beta) : (\chi, \beta) \in B],$$

donde B es el conjunto de ramas admisibles. Después de que el problema se resuelve, la *percepción tardía* se utiliza para examinar las ramas que se cortarán de tal modo que se eliminen las trayectorias muertas. La idea es ajustar los pesos w_i para asegurar al mejor ganador (conocido con percepción tardía) para los *problemas de entrenamiento*.

Para este caso las desigualdades lineales para las w_i , $i = 1, 2, \dots, k$ se generan de la siguiente manera: Sea (u, v) que denote a la rama que puede tomarse (con percepción tardía). Entonces, se ajustan los pesos para hacerlo ganar:

$$\sum_i w_i F_i(u, v) \leq \sum_i w_i F_i(\chi, \beta),$$

para toda $(\chi, \beta) \in B$.

Hasta que exista una solución a esas desigualdades, un nuevo peso se escoge. Por construcción, la elección de los pesos asegura una trayectoria más corta a una solución para todos los problemas de entrenamiento (sin trayectorias muertas). Eventualmente, puede no existir un vector de pesos que lo asegure, por lo que se requiere de algún criterio. Una forma, por supuesto, de definir una proximidad razonable es mínimos

cuadrados, pero una mejor alternativa es empezar a clasificar a los problemas y obtener el mecanismo de aprendizaje de manera simultánea, así como el de tratar simultáneamente al problema para obtener los parámetros del algoritmo optimal dentro de cada clase.

Capítulo 6

Bondades y alcances de la programación estocástica.

En los capítulos anteriores de este libro, se ha podido observar el *excelente* comportamiento de los métodos de programación estocástica para resolver problemas de tipo combinatorio. Sin embargo, existe una pregunta muy importante: ¿qué tan buena es su realización?.

La pregunta anterior es de gran importancia para cualquier método, pero reviste una importancia principal para los métodos de tipo heurístico, puesto que esto nos permitirá, por un lado, considerar qué tan buena es una solución proporcionada por este tipo de métodos, y por otro lado, nos puede proporcionar un criterio de *paro* del método utilizado, o en su defecto, si se utilizan de manera híbrida con otros métodos, el de proporcionarnos procedimientos para mejorar las soluciones.

6.1 Inventarios multiproductos.

Son grandes los retos a los que se debe de enfrentar las empresas, entre los que destacan la competencia creciente, las dificultades de abastecimiento, la sobreabundancia de artículos y la escases de materia prima, por lo que es de importancia cada día mayor la adecuada administración económica de los inventarios.

Una forma de asegurar la continuidad de las operaciones es el alma-

cenamiento, sin embargo, esto desencadena costos suplementarios, lo que por efecto ocasiona una reducción del margen de utilidad, por otro lado, la falta de inventarios adecuados puede propiciar un rompimiento en la continuidad de operación y altos costos de oportunidad por no poder satisfacer la demanda.

En esta sección se desarrolla para caso de los los modelos de inventario multiproducto con demanda determinística y con costos de reaprovisionamiento conjunto, un algoritmo que por la primera vez¹ utiliza un esquema de la búsqueda tabú para este tipo de problemas, y proporciona mayor eficiencia en cuanto a las soluciones obtenidas con respecto a algoritmos reportados como eficientes. Se reporta en la experiencia computacional, la eficiencia de este marco de solución propuesto como la mayor en todas las instancias consideradas, con respecto de los demás algoritmos probados.

La sección del capítulo se desarrolla como sigue: en la subsección 1, se describe el modelo de inventarios multiproducto para el caso de demanda determinística, así como algunos de los algoritmos que resuelven el problema con eficiencia; en la subsección 2, se considera a la búsqueda tabú como un metaprocedimiento para realizar una fase de mejoramiento y se realiza un análisis comparativo contra otros métodos, en la subsección 3, se describen algunos métodos de manejo de la lista tabú de tipo dinámico. Finalmente en la subsección 4, se presenta la experiencia computacional.

6.1.1 Inventarios multiproducto: caso determinístico.

El problema de inventarios multiproductos consiste en general en determinar cuánto y cuándo ordenar de cada artículo de los que se controlan, para que, la demanda sea satisfecha con costos mínimos. En este trabajo se supone que la entrega por parte de los proveedores es inmediata y se realiza en una sola entrega, no se permite déficit y los artículos nunca llegan a ser obsoletos una vez que se inventarean. Además se

¹de los Cobos(1994).

supone que diferentes artículos de un mismo proveedor se pueden entregar en un solo paquete y se realiza a intervalos regulares de tiempo, digamos t_1 .

El problema para n artículos por tanto se traduce en determinar la frecuencia de los ciclos de abasto tanto individual como en conjunto, por lo que se puede observar fácilmente que el cociente de la frecuencia individual del artículo j digamos t_j entre la frecuencia conjunta es un entero positivo β_j mayor o igual a uno, i.e., $t_j = \beta_j t_1$, $j = 1, 2, \dots, n$

Los artículos tienen diferentes ciclos, pero el que marca las oportunidades para ordenar es el menor duración t_1 . Se ordena tantas veces como lo requiera dicho artículo en una unidad de tiempo, $1/t_1$, y cada vez que se ordena se incluyen en la orden los artículos cuyo ciclo así lo indica.

Los costos que intervienen en el modelo son de: ordenación, adquisición e inventario. Un costo fijo K se carga cada vez que se realiza una orden. Un costo por línea k_j se produce cada vez que se carga el artículo j en la orden correspondiente.

El costo debido al inventario se carga sobre el inventario promedio durante un ciclo y está dado por:

$$\frac{1}{2} \sum_{j=1}^n h_j d_j t_j.$$

donde: h_j , d_j , y t_j son el inventario promedio, la demanda promedio y el tiempo del ciclo del artículo j , respectivamente.

De lo anterior se tiene que el costo total por unidad de tiempo es:

$$CT = \frac{1}{t_1} \left(K + \sum_{j=1}^n \frac{k_j}{\beta_j} + \sum_{j=1}^n c_j d_j \right) + \frac{1}{2} t_1 \sum_{j=1}^n h_j d_j \beta_j.$$

el cual se desea minimizar. El costo total no es sólo función de t_1 , sino también de las β_j , $j = 1, 2, \dots, n$ y donde las c_j , $j = 1, 2, \dots, n$ son los costos unitarios.

La solución propuesta en Narro(1993) consiste en aproximar los valores de las β_j 's y después encontrar el valor de t_1 que minimiza a $CT(t_1)$,

de la siguiente manera:

El costo correspondiente a cada artículo ordenado se puede expresar como: $\alpha_j K$, con $\alpha_j \geq 0$, $\sum_{j=1}^n \alpha_j = 1$. Si la política óptima para cada artículo j , considerado como independiente, aconseja ordenar x_j unidades en cierto instante, entonces se tiene que:

$$K\delta\left(\sum_{j=1}^n x_j\right) + \sum_{j=1}^n k_j\delta(x_j) \geq \sum_{j=1}^n (\alpha_j K + k_j)\delta(x_j),$$

donde $\delta(x) = 1$ si $x > 0$, y cero de otra forma.

La política óptima es una política factible para el sistema de inventario del artículo j con costo por ordenar $(\alpha_j K + k_j)$. Si el costo mínimo correspondiente es $(CT)_j(\alpha_j)$, una cota inferior para el costo de la política óptima completa está dada por: $\sum_{j=1}^n (CT)_j(\alpha_j)$, para cualquier asignación de las α_j 's, por lo que la mejor cota inferior está dada por:

$$\begin{aligned} & \text{Minimizar}_{\alpha_i} \sum_i (CT)_i(\alpha_i) \\ \text{s.a. } & \alpha_i \geq 0, \quad i = 1, 2, \dots, n, \quad \sum_{i=1}^n \alpha_i = 1. \end{aligned}$$

Ahora bien, las formas en que se calculan los ciclos y se aproximan los múltiplos son diversas. Si l es el real que resulta del cociente del tiempo de duración del ciclo correspondiente y el de menor longitud y β es el entero más próximo, entonces se debe satisfacer que: $\beta(\beta+1) < l^2 < (\beta+1)l$.

El algoritmo propuesto por Narro(1993) se resume en:

Paso 1.- Resolver:

$$\begin{aligned} & \text{Minimizar}_{\alpha_i} \sum_i (CT)_i(\alpha_i) \\ \text{s.a. } & \alpha_i \geq 0, \quad i = 1, 2, \dots, n, \quad \sum_{i=1}^n \alpha_i = 1. \end{aligned}$$

Paso 2.- Calcular para $j=1,2,\dots,n$:

$$l_j'^2 = \frac{k_j}{d_j h_j} \max \left\{ \frac{d_i h_i}{K + k_i}, i = 1, 2, \dots, n \right\},$$

$$\beta_j'^2 = \frac{1 + \sqrt{1 + 4l_j'^2}}{2},$$

$$t_1' = \sqrt{\frac{2(K + \sum_{i=1}^n \frac{k_i}{\beta_i'})}{\sum_{i=1}^n h_i d_i \beta_i'}},$$

$$l_j^2 = \frac{2(\alpha_j K + k_j)}{\frac{d_j h_j}{t_1'^2}},$$

$$\beta_j^2 = \frac{1 + \sqrt{1 + 4l_j^2}}{2}.$$

paso 3.- Resolver:

$$\text{Minimizar } \frac{1}{t_1} \left(K + \sum_{j=1}^n \frac{k_j}{\beta_j} + \sum_{j=1}^n c_j d_j \right) + \frac{1}{2} t_1 \sum_{j=1}^n h_j d_j \beta_j, \text{ s.a. } t_1 \geq 0.$$

6.1.2 Marco de mejoramiento por Búsqueda Tabú.

A fin de iniciar la búsqueda para mejorar los planes maestros de los pedidos a proveedores una orden inicial se requiere. Esta se obtiene de manera natural al resolver el paso 1 y poner $\beta_j = 1, j = 1, 2, \dots, n$. Ahora bien, para utilizar a la búsqueda tabú como un *metaprocedimiento de mejora*, en lugar de utilizar valores fijos para las $\beta_j, j =$

$1, 2, \dots, n$, se propone que cada β esté dentro de un rango de valores dado por:

$$\beta_j \in [1, 2\beta_j^*], \text{ donde } \beta_j^* = \left\lceil \sqrt{\frac{1 + \sqrt{1 + 4l_j^2}}{2}} \right\rceil \text{ para } j = 1, 2, \dots, n,$$

donde $\lceil x \rceil$ es el entero mayor más cercano de x .

Por tanto, lo que se desea es resolver:

$$\text{Minimizar } \frac{1}{t_1} \left(K + \sum_{j=1}^n \frac{k_j}{\beta_j} + \sum_{j=1}^n c_j d_j \right) + \frac{1}{2} t_1 \sum_{j=1}^n h_j d_j \beta_j,$$

$$\text{s.a. } t_1 \geq 0, \quad \beta_j^* \in [1, 2\beta_j^*], \quad j = 1, 2, \dots, n.$$

el cual es un problema de programación no lineal entero mixto, pero como se observa la función objetivo resulta ser una función convexa en t_1 , por lo que, una vez fijos los valores de las β_j^* , el valor óptimo para t_1 se encuentra fácilmente.

Consideraremos que una solución es de la forma:

$$\pi = (\pi(1), \pi(2), \dots, \pi(n)) \text{ donde } \pi(i) \in [1, \beta_i^* + 1], \quad i = 1, 2, \dots, n.$$

entonces, definimos un movimiento de la forma $\pi' = \pi \pm e_j$, $j = 1, 2, \dots, n$, tal que $\pi'(i) \in [1, \beta_i^* + 1]$, $i = 1, 2, \dots, n$. En una implantación sencilla de BT, la lista tabú se mantiene en forma implícita registrando sólo el índice j de la variable entera cuyo valor ha cambiado en la iteración.

6.1.3 Manejo dinámico de la lista tabú.

El *manejo de la lista tabú* significa la actualización de ésta, i.e., la decisión de cuántos y cuáles movimientos se pondrán como tabú para una

iteración de la búsqueda. Muchas de las implantaciones de la búsqueda tabú utilizan un manejo de la lista tabú de tipo estático. De manera más precisa, los movimientos permanecen como tabú durante un número de iteraciones, por lo que la eficiencia del algoritmo depende de la elección de la duración del estatus tabú o, equivalentemente, de la longitud de esta lista.

Este manejo de la lista tabú de tipo dinámico permite el examen más detallado de la región factible, por lo que es posible romper ciclos y diversificar la búsqueda.

Existen varios métodos de lista tabú dinámica, entre los que destacan: el método de la *secuencia de Cancelación* (CSM) y el método de *eliminación inversa* (REM) propuestos en Glover (1990a).

6.1.4 Resultados computacionales.

Esta técnica de la búsqueda tabú para el problema de inventarios multiproducto con demanda determinística y costos de reaprovisionamiento conjuntos, probó su robustez al encontrar mejores soluciones respecto a las reportadas por algoritmos eficientes de la literatura, en particular se compararon los siguientes:

KyR : Kaspi y Rosenblatt (1985), **N-R** : Narro(1993).

GyB : Goyal y Belton (1979), **TB** : Búsqueda Tabú(de los Cobos(1994)).

Las Tablas 6.1 y 6.2 son ejemplos de Narro(1993) donde se muestra la comparación de los algoritmos.

j	d_j	h_j	k_j	GyB (β_j)	KyR (β_j)	N-R (α_j, β_j)		BT (β_j)
1	3000	2	5	1	1	0.065	1	1
2	20000	2	7	1	1	0.0	1	1
3	12500	1.6	4	1	1	0.506	1	1
4	10000	1.45	5	1	1	0.0	1	1
5	4250	2	5	1	1	0.14	1	1
6	2500	3	7	1	1	0.156	1	1
7	5000	1	10	1	2	0.0	2	2
8	3000	1.25	6	1	1	0.089	1	2
9	2000	3	6	1	1	0.042	1	1
10	1500	2	9	2	2	0.0	2	2
t_1				0.0419	0.04	0.026		0.038
CT				4914.5	4895	4895		4892.85

Tabla 6.1 Ejemplo de 10 artículos.

j	d_j	h_j	k_j	KyR (β_j)	N-R (α_j, β_j)	BT (β_j)
1	10000	10	10	1	0.33534	1
2	35000	2	8	1	0.23943	1
3	8000	8	8	1	0.178644	1
4	9000	5	6	1	0.11059	1
5	50000	1	8	1	0.0	1
6	15000	2	5	1	0.05467	1
7	20000	2	7	1	0.6532	1
8	12500	1.6	4.5	1	0.0099	1
9	10000	1.45	8	1	0.0	2
10	4250	2	5	2	0.0	2
11	2500	3	7	2	0.0	2
12	10000	1	10	1	0.0	2
13	4000	1.25	5	1	0.0	2
14	2000	3	6	1	0.0	2
15	1500	2	9	3	0.0	4
CT				11632.35	11772.5	11449.76

Tabla 6.2 Ejemplo de 15 artículos.

Para probar la robustez de la eficiencia del algoritmo propuesto en este trabajo se corrieron para cada instancia los tres algoritmos. Se generaron 100 instancias aleatorias para cada uno de los tamaños $n = 5, 10, 15, 20, 25$ y 30 , con los siguientes rangos de las variables:

$$2000 \leq d_j \leq 50000, d_j \text{ entero}, j = 1, 2, \dots, n,$$

$$1 \leq h_j \leq 8, j = 1, 2, \dots, n,$$

$$1 \leq k_j \leq 12, j = 1, 2, \dots, n.$$

Para resolver el sistema del Paso 1, se utilizó el paquete GINO, los resultados obtenidos fueron los siguientes:

n		KyR		N-R		BT	
		Costo	Tiempo	Costo	Tiempo	Costo	Tiempo
5	min.	4908	0.0297	4995	0.0310	4908	0.0297
	max.	6928	0.0183	6928	0.0183	6928	0.0183
	prom.	5805	0.0237	5835	0.0249	5805	0.0237
10	min.	8743	0.0216	8791	0.0228	8716	0.0213
	max.	10125	0.0174	10176	0.0182	10064	0.0171
	prom.	9632	0.0179	9664	0.0180	9602	0.0177
15	min.	7992	0.0201	7992	0.0210	7896	0.0207
	max.	11993	0.0144	12059	0.0148	11993	0.0144
	prom.	10448	0.0158	10545	0.0166	10446	0.0160
20	min.	12907	0.0134	13196	0.0145	12864	0.0129
	max.	15767	0.0100	16244	0.0110	15732	0.0099
	prom.	14650	0.0122	15080	0.0134	14609	0.0118
25	min.	13497	0.0188	13709	0.0193	13346	0.0172
	max.	17361	0.0131	17924	0.0144	17309	0.0127
	prom.	15035	0.0166	15345	0.0175	14958	0.0158
30	min.	18593	0.0137	19250	0.0148	18354	0.0123
	max.	21038	0.0128	21769	0.0138	20679	0.0106
	prom.	19947	0.0132	20652	0.0142	19615	0.0111

Tabla 6.3 Valores generados por 100 instancias aleatorias.

A continuación se presenta la eficiencia que se calculó tomando como base el valor z_{tabu} , esto es, el valor de la función objetivo obtenido por el algoritmo propuesto en este trabajo y se utilizó la ecuación:

$$eficiencia = 1 - \frac{z - z_{tabu}}{z_{tabu}}$$

Los resultados son los siguientes:

n		KyR	N-R	BT
5	min.	1	0.982	1
	max.	1	1	1
	prom.	1	0.994	1
10	min.	0.996	0.991	1
	max.	0.993	0.988	1
	prom.	0.996	0.993	1
15	min.	0.999	0.999	1
	max.	1	0.994	1
	prom.	0.999	0.990	1
20	min.	0.996	0.974	1
	max.	0.997	0.967	1
	prom.	0.997	0.967	1
25	min.	0.988	0.972	1
	max.	0.996	0.964	1
	prom.	0.994	0.974	1
30	min.	0.986	0.951	1
	max.	0.982	0.947	1
	prom.	0.983	0.947	1

Tabla 6.4 Eficiencia Comparativa.

Se puede observar de esta última Tabla 6.4 que la eficiencia del algoritmo propuesto utilizando un marco de búsqueda tabú es la más elevada. El número de iteraciones bajo el marco de búsqueda tabú² que se realizaron fueron de $2n$ donde n es el número de artículos considerados. Cabe señalar que en el estudio de Kaspi y Rosenblatt compararon la efectividad de varios algoritmos generando 132000 ejemplos y cubriendo 132 clases diferentes.

²Cabe mencionar que existen algoritmos de mayor eficiencia para el problema propuesto.

Referencias.

- Aarts, E. y Korst, J., *Simulated Annealing and Boltzmann Machines*, Wiley, 1989. [AAR89]
- Barnes J. W. and Vanston L. K. (1981), *Scheduling Jobs with Linear Delay Penalties and Sequence Dependent Setup Costs*, ORSA 29:1, 146-159.
- Barnes J. W. and Laguna M. (1993), *A Tabu Search Experience in Production Scheduling*. Annals of Ops. Res. 41. 141-156
- Cerny V. (1985), *Thermodynamical Approach to the Traveling Salesman Problem. An efficient simulation algorithm*, Journal of Optimization Theory and Applications 45,41-52
- Chaudhry, S., McCormick, S.T. y Douglas, M., *Locating Independent Facilities with Maximum Weight: Greedy Heuristics*, Omega Vol. 14 No. 5, 1986, pag. 383-389. [CHA86]
- Colin R.R.(Ed.), (1993), "Modern Heuristic Techniques for combinatorial Problems", John and Willey Sons, N.Y.
- Committee on the Next Decade of Operations Research (Condor 1988), *Operations Research: The Next Decade*. Ops. Res. 36
- Dammeyer F., and Voss S., (1993), *Dynamic Tabu List Management Using the Reverse Elimination Method*, Annal of Ops. Res. 41, 31-46
- Davis L. and Streenstrup, M., (1987), "Genetic Algorithms and Simulated Annealing", L. Davis(ed.), Pitman, London.
- de los Cobos Silva S. (1987), *Un Sistema para la Resolución del Problema de Localización de Facilidades*, Tesis de Maestría, C.E.C.-Colegio de Postgraduados, Chapingo México.
- de los Cobos Silva S. (1994), *La Técnica de la Búsqueda Tabú y sus Aplicaciones*, tesis doctoral, DEP-FI, UNAM.
- de los Cobos Silva S. G., Pérez Salvador B.R. y Gutierréz Andrade. M.A. (1995) "Lineamientos de implantación de la búsqueda tabú para los problemas de calendarización", IX Simposio Métodos Matemáticos Aplicados a las Ciencias, UCR, Turrialba, Costa Rica, J. Trejos(ed.).
- de los Cobos Silva S. G., Gutierréz Andrade. M.A. y Pérez Salvador B.R., (1996)

"Un ejemplo de cómo saltar valles profundos en problemas combinatorios.", *Revista CONTACTOS*, No. 15, pp. 36-41.

Fox B. (1993), Integrating and Accelerating Tabu Search, Simulated Annealing, and Genetic Algorithms, *Annals of Ops. Res.* **41**, 47-67.

Glover F. (1977), Heuristics for Integer Programming using Surrogate Constrains. *Decision Sciences* **8**, 156-166

Glover F. (1987), Tabu Search, *Preliminary draft: Paper presented at the ORSA/TIMS Joint National Meeting, St. Louis. Center For Applied Artificial Intelligence, University of Colorado*

Glover F. (1989), Tabu Search, Part I, *ORSA Journal on Computing* **1:3**, 190-206.

Glover F. and Greenberg H. J. (1989), New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence, *European J. of Opt. Res.* **39**, 119-130.

Glover F. (1990), Tabu Search: A Tutorial, *Interfaces* **20:4**, 74-94.

Glover F., Klingman D. and Phillips N. (1990), Netform Modeling and Applications, *Interfaces* **20:4**, 7-27.

Glover F. (1990a), Tabu Search, Part II, *ORSA Journal on Computing* **2:1**, 4-31.

Glover F. (1993), Genetic Algorithms and Scatter Search: Unsuspected Potentials. *Technical report (August), School of Busines, University of Colorado at Boulder.*

Glover F. and Laguna M. (1993), Tabu Search, *Modern Heuristic Techniques for Combinatorial Problems*, Colin R. Reeves(Ed.), 70-150, *Blackwell Scientific Publications, Oxford.*

Glover F., Tailard E. and de Werra D. (1993), A User's Guide to Tabu Search, *Annals of Ops. Res.* **41**, 3-28.

Goyal S.K. and Belton A.S. (1979), On a Simple Method of Determining Order Qualities under Deterministic Demand, *management Sciences* **25**, 604.

Gutierrez-Andrade M. (1991), La técnica del Recocido Simulado y sus Aplicaciones, *Tesis de Doctorado, D.E.P.F.I.-U.N.A.M., México.*

Hadj-Alouane A., Bean J. C. and Murty K. G. (1993), A Hybrid Genetic/Optimization Algorithm for Task Allocation Problem, *Technical report (November), Dept. of Ind. and Ops. Engineering, The University of Michigan at Ann Arbor.*

Hadj-Alouane A. and Bean J. C. (1993). A Genetic Algorithm for the Multiple-Choice Integer Program, *Technical report (July), Dept. of Ind. and Ops. Engineering, The University of Michigan at Ann Arbor.*

Holland J. (1975). *Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.*

Hooker J.N. (1994), Needed: An Empirical Science Of Algorithms, *Operation Research* 42: 2, 201-212.

Jacobsen, S.K. (1983), Heuristics for the Capacitated Plant Location Model, *European Journal of Operational Research* 12, 1983, pag.253-261. [JAC83]

Kaspi M. and Rosenblantt M.J. (1985), The Effectiveness of Heuristic Algorithm for Multi-Item Inventory Systems with Joint Replenishment Costs, *Int. Journal of production Research* 23, 109-116.

Khumawala, B.M. (1972), An Efficient Branch and Bound Algorithm for the Warehouse Location Problem, *Management Science* 18, 1972, pag. 718-731.[KHU72]

Laarhoven P.J.M. van y Aarts E.H.L. (1988), *Simulated Annealing: Theory and Applications, Reidel, 1988. [LAA88]*

Laguna M., Barnes J. W. and Glover F. (1990), Tabu Search for a Single Machine Scheduling Problem, *Technical report (july), Advanced Knowledge Systems Group of U S, West Advanced Technologies.*

Laguna M. (1993), A Guide to Implementing Tabu Search, *Technical report (October), Graduate School of Business and Administration, University of Colorado, Boulder.*

Laguna M. and Kelly J. (1993), Master Production Scheduling in a Single Facility with Sequence-Dependent Changeover Times, *Technical report (September), Graduate School of Business and Administration, University of Colorado at Bould*

Laguna M. and Glover F. (1993), Integrating Target Analysis and Tabu Search for Improved Scheduling Systems, *Expert Systems with Aplication, Vol. 6, 287-297.*

Love, R.F., Morris, J.G., Wesolowsky, G.O., (1988) *Facilities Location Models & Methods, North-Holland, 1988. [LOV88]*

Mirchandani, P.B. y Francis, R.L. editores, (1990) *Discrete Location Theory, Wiley, 1990. [MIR90]*

Murty K. G. (1992), *Network Programming, Prentice Hall.*

Narro R. A. (1993), "Sistemas de Inventarios", *reporte técnico, DEP-FI, UNAM.*

Narro R. A., and de los Cobos S. S., (1994), "Algorithms for Multiple-Item Inventory Systems", *15th international Symposium on Mathematical Programming, August 15-19, Ann Arbor, Michigan, USA.*

Pearl J. (1985), *Heuristics, Addison-Wesley.*

Rechenberg I. (1964), *Cybernetic Solution Path of an Experimental Problem, Royal Aircraft Establishment, Library Translation 1122, Farnborough.*

Rechenberg I. (1989), *Artificial Evolution and Artificial Intelligence, Evolution in Medicine; Learning; Principles and Techniques, Chapman and Hall.*

Rosenblatt F. (1962), *Principles of Neurodynamics, Spartan, Washington, DC.*

Shannon, R.E. y Ignizio, J.P. (1975) *A heuristic programming algorithms for warehouse location, AIIE Transactions vol. II, No 4, pag. 334. [SHA75]*

Glosario.

- agujero negro, 13, 33
- algoritmo genético, 3, 69
- algoritmo Khumawala, 47
- algoritmo Metropolis, 55
- algoritmo Shanon e Ignizio, 51
- allele, 69
- análisis de objetivos, 3
- atributo, 6

- bondades de programación estocástica, 79
- búsqueda dispersa, 3
- búsqueda tabú, 3, 13

- ciclado, 13
- constante de Boltzmann, 54
- criterio de aspiración, 15, 19, 36, 56
- criterio Metropolis, 55
- cruzamiento, 69

- descendencia, 69
- distribución de Boltzmann, 54
- diversificación regional, 20

- eficiencia de algoritmo, 86
- entrampamiento suboptimal, 33
- enumeración implícita, 47
- esfuerzo computacional, 7
- esquema genético, 73
- estrategia de evolución biológica, 69
- estrategia de oscilación, 28
- estructura de vecindades exacta, 7
- explosión combinatoria, 1

- factor de Boltzmann, 54

función costo, 6
función partición, 54

gene, 69

influencia, 19
inteligencia artificial, 2
intensificación regional, 20
instancia, 6
inventarios multiproducto, 79

lista tabú, 14

manejo de lista tabú, 84
matriz de frecuencias, 23
matriz tabú, 23
máximo local, 7
mecanismo de generación, 7, 18, 20
memoria de término corto, 15
memoria de término intermedio, 21
memoria de término largo, 21
métodos de simulación, 55
métodos heurísticos, 3
métodos híbridos, 79
mínimo local, 7
movimiento, 5
movimiento admisible, 15, 17
movimiento candidato, 17
movimiento de intercambio, 17
movimiento élite, 20
mutación, 68, 69

NP-completo, 47

optimización combinatoria, 5, 6
óptimo global, 6
óptimo local, 7

parámetro de control, 56
penalización, 36
percepción tardía, 77
plan reproductivo, 69
política óptima, 82
problema del agente viajero, 1, 38
problema de entrenamiento, 77
problema de localización de plantas, 2. 46
problema de las N reinas, 9
problema entero, 46
problema del transporte, 46
problemas NP-completos, 8
problemas NP-difíciles, 8
progénito, 69
programación estocástica, 3

ramificación y acotamiento, 47
recocido simulado, 53
redes neuronales, 3
reducibilidad de problemas, 8
reglas hereditarias, 67
representación del problema, 59
reproducción, 69
robustes de algoritmo, 86
rutas óptimas, 42

satisfacibilidad, 10
semioptimización, 10, 13
solución muestra, 69
soluciones padre, 69
suboptimización, 10

técnica Monte Carlo, 55
transición, 57

valle profundo, 13, 32
valor de movimiento, 17



**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA**

**CURSO ABIERTO
TOPICOS DE ACTUALIDAD DE LA INVESTIGACION
DE OPERACIONES Y LA SIMULACION**

**SIMULACION UTILIZANDO
POWERSIM y GPSS/H**

**PROFESOR
M EN I MARCO A. MONTUFAR BENITEZ
PALACIO DE MINERIA
1996**

**TOPICOS DE ACTUALIDAD DE LA INVESTIGACION
DE OPERACIONES
Y LA SIMULACION**

MINITALLER DE SIMULACION

**UTILIZANDO GPSSH, POWERSIM Y PROOF
ANIMATION**

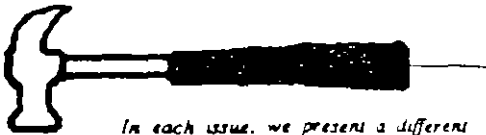
EXPOSITOR MARCO A. MONTUFAR BENITEZ

25 - SEPTIEMBRE - 1996

MEXICO D.F.

4.1

ARQUETIPOS SYSTEMICOS



Toolbox

In each issue, we present a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. See page 10 for a symbol key for the diagrams.

Success to the Successful: Self-Fulfilling Prophecies

By Daniel H. Kim

Imagine you have two new direct reports, Stan and Frank. Both seem equally qualified—a degree from a good school, a couple years of solid business experience, and youthful enthusiasm. You want to fill an upcoming opening in a management position, but you aren't quite sure which one is the best candidate. You want to be as objective as possible in your recommendation, so you decide to encourage both of them and see which one demonstrates the most ability.

After a couple of weeks, Stan has gotten a jump start on the latest assignment and is doing a stellar job. Frank was out with the flu, so when he comes back, he's a little bit behind. You keep your eye on him and continue to encourage him, but you really start to focus on Stan. Before you know it, you're giving him more and more responsibility, and he does exceedingly well each time. Frank is

doing adequately, but for some reason, you just don't feel like he has that extra "umph." Since you already have a "hot one" on your hands, you feel it's not as necessary to invest as much time and energy in him.

In time, you promote Stan into the management position and pat yourself on the back for having picked the right person from the beginning. Frank, in your assessment, turned out to be just an average performer. But is that really the case?

Self-Fulfilling Prophecies

The "Success to the Successful" archetype suggests that success may depend as much on structural forces as innate ability or talent (see "Success to the Successful Template"). The performance of individuals or teams is often the result of the structure they are put in which forces them to compete for a limited resource such as a manager's time, a company's investments, or training facilities.

Assuming both groups (or individuals) are equally capable, if one person or group (A) is given more resources, it has a higher likelihood of succeeding than B. That initial success justifies devoting more resources to A and robs B of further resources (R1). As B gets less resources, its success

diminishes, which further reinforces the "bet on the winner" allocation of resources (R2). The structure continues to reinforce the success of one, and the eventual demise of the other.

"Success to the Successful" is an archetypal case of self-fulfilling prophecies. The outcome of a situation is highly dependent on the initial conditions (or expectations) and whether they favor one party or the other. If B had received more resources in the beginning, the roles would be reversed: B's success would increase, and A would suffer.

In effect, our mental model of what we believe will determine success shapes the very success we seek to assess. In the case of Stan and Frank, the manager may not have had a strong feeling either way in the beginning. But initial events—Stan's success with the first project and Frank's illness—quickly became the shaper of his expectations and actions, reinforcing what he later believed to be an objective assessment that Stan was "right" for the job.

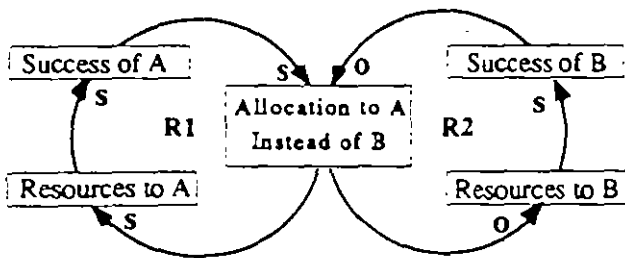
Balancing Work and Family

The tension between work and family is another example of the "Success to the Successful" archetype (see "Balancing Work and Family" diagram). We each have a certain amount of time and attention available. The more we devote to work, the more successful we may become, which fuels the desire to put more time into work (R3). A similar result occurs if we devote our energy to our family (R4). Most of us struggle to maintain a balance between the two.

Suppose, however, that a large project forces you to put in long hours at work for an extended period of time. The time away from the family begins

Continued on next page

Success to the Successful Template



The "Success to the Successful" archetype suggests that success may depend as much on structural forces as talent. If one person or group (A) is given more resources, it has a higher likelihood of succeeding than B (assuming they are equally capable). The initial success justifies devoting more resources to A than B (R1). As B gets less resources, its success diminishes, further justifying more resource allocations to A (R2).

Continued from previous page

to create tension at home. Your family complains that you are never around. But when you do come home, you get hit with all of the problems that have been accumulating. So you withdraw further from your family, devoting yourself even more to the project. Your work on the project is starting to generate interest throughout the company. At the same time that praise at work is building, the complaints at home are piling up, driving you further from your family. The two situations mutually feed each other—the downward spiral of one, and the upward spiral of the other.

Rewriting the Prophecies

The "Success to the Successful" archetype highlights how success can be determined by initial chance and how the structure can systematically eliminate the other possibilities that may have been equally viable (or even superior). If we are not conscious of being in this archetype, we become a victim of its structure, which continually pushes us to do whatever

has been successful in the past. After a while, the choice between work and family doesn't seem like a choice anymore—the structure has determined the outcome.

As in most of the archetypes, managing a "Success to the Successful" situation requires looking at it from a more macro level and asking ourselves "what is the larger goal within which the situation is embedded?" In the case of work vs. family, a larger goal that includes both of them, such as, "I seek a balance between my success at work and time with my family," must guide the daily decisions. In the case of the two proteges, the goal might be to provide an environment in which the full potential of both employees can be developed. Without the guidance of a larger goal, the structure will continue to dictate your actions.


Creating Environments for Success

At the heart of the "Success to the Successful" archetype lies the competitive model of Western economies which is characterized by a win-lose philosophy. An implicit assumption of the competitive model is that whoever wins must, by default, be the best. In reality, however, it may not be the individuals, but the structure they are in that determines the "winner." The assumption here is that

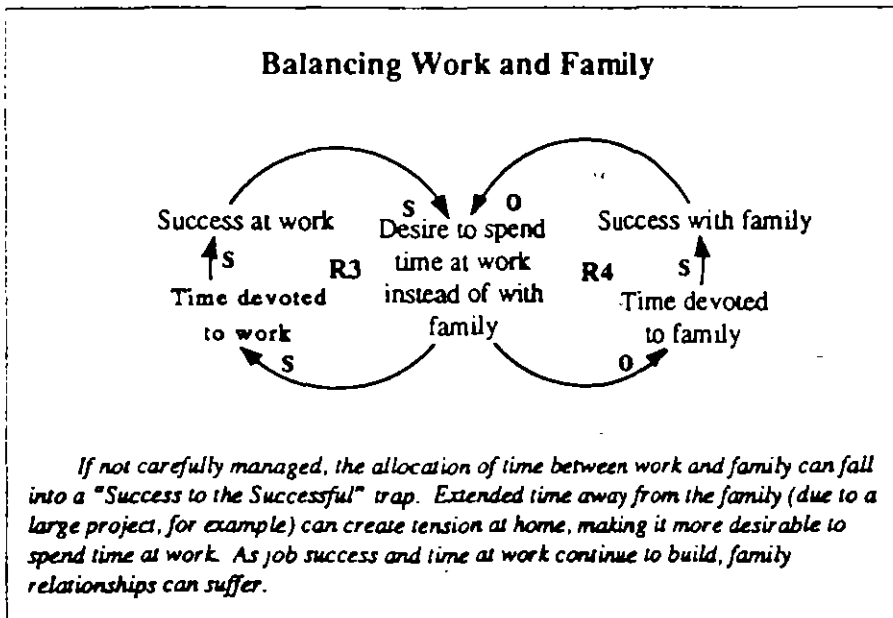
you need a competitive environment out of which the best candidates will somehow surface.

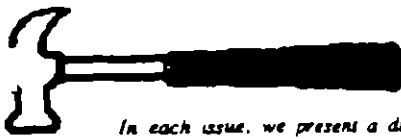
A fundamental question that the archetype begs us to ask is why put two groups or individuals into the structure in the first place? If we want a single "winner," why not put our energies towards understanding what it takes to develop such a winner? We can then focus our energy and resources on that person or project from the beginning rather than waste time, money, and morale by stringing along multiple people and projects. We can, in effect, lop off the other half of the "Success to the Successful" archetype. Instead of diverting resources and systematically letting other groups to fail, we can focus all our efforts and resources on finding ways to build a supportive environment for success.

A way to break out of the "Success to the Successful" archetype is to get rid of its competitive structure and find ways to make teams collaborators rather than competitors. Many Japanese companies, for example, have multiple project teams working the same design. Unlike American companies, however, the goal is *not* to compete against each other and have one team's design win. All of the teams are seen as part of the same larger effort to develop the best design for the company. The teams collaborate with each other, sharing ideas and information, and produce a design that may be a combination of innovations from each of the groups.

The "Success to the Successful" archetype highlights the need for creating a win-win environment where cooperation replaces competition and where creating an environment for success is more important than trying to identify successful individuals. In fact, that's what good academic institutions provide, and ultimately what good corporate environments should provide—an environment in which all their people can thrive and contribute their unique talents. 

Success to the Successful" and oth templates can be found in The Fifth Discipline by Peter Senge (Doubleday).





Too box

In each issue, we present a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. See page 10 for a symbol key for the diagrams.

Organizational Addictions: Breaking the Habit

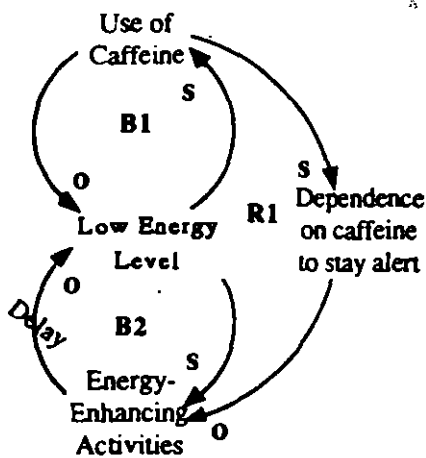
By Daniel H. Kim

It's 6:00 a.m. on a Monday morning. The alarm clock blares, jolting you out of bed. You shuffle down to the kitchen and grab a cup of fresh coffee. A few gulps and...ahh. Your eyes start to open; the fog begins to clear.

10:30 a.m.—time for the weekly staff meeting. "I gotta have something to keep me awake through this one," you think to yourself as you grab a cup of coffee and head into the conference room.

By 3:30 p.m. you start to feel that mid-afternoon energy low, so you head down toward the crowded coffee machine for another cup. "I really gotta cut down on this stuff," you comment to the guy behind you in line. He nods. "I'm a five-cup-a-day guy," he confesses. "I just can't give it up."

Caffeine Addiction



Low energy can be counteracted by more sleep or exercise—but that takes time (B2). A cup of coffee immediately restores energy (B1). But it also leads to a dependence on caffeine to stay alert, which takes attention away from long-term energy-boosters (R1).

Addiction

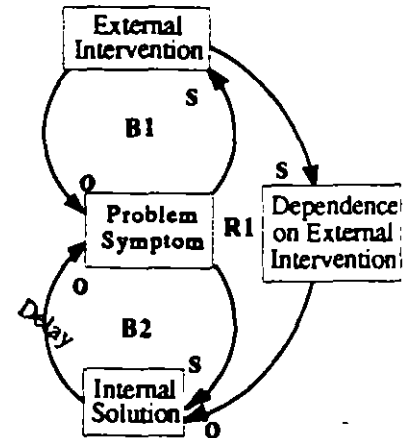
For most of us, the word "addiction" conjures up images of alcoholism and drug abuse or more "acceptable" habits such as coffee drinking—dependencies which are rooted in physical and neurological processes. It is not usually viewed as a social or organizational phenomena. But from a systemic perspective, addiction is a very generic structure which is quite prevalent in both social and organizational settings.

As a systemic structure, the "Addiction" archetype is a special case of "Shifting the Burden" (see "Shifting the Burden: The Helen Keller Loops," September 1990). "Shifting the Burden" usually starts with a problem symptom that cries out for attention. The solution that is most obvious and easy to implement usually relieves the problem symptom very quickly. But the symptomatic solution has a long-term side effect that diverts attention away from the more fundamental solution to the problem (see "Addiction: A Special Case of 'Shifting the Burden'").

What makes the "Addiction" archetype special is the nature of the side-effect. In an "Addiction" structure, a "Shifting the Burden" situation degrades into an addictive pattern in which the side-effect gets so entrenched that it overwhelms the original problem symptom—the addiction becomes "the problem."

With coffee drinking, the problem symptom usually is that you feel tired (see "Coffee Addiction"). When you drink a cup of coffee, the caffeine raises your metabolism, stimulating the

Addiction: A Special Case of "Shifting the Burden"



The "Addiction" archetype is a special case of "Shifting the Burden." In both cases, a problem symptom is "solved" by applying a symptomatic solution (B1), but the solution has a side-effect which diverts attention away from the fundamental solution (R1). This side-effect—the dependence on an external intervention—eventually overwhelms the original problem.

body and making the mind more alert. But in doing this, it forces your body to deplete its reserves of energy faster than usual. When the effects of the caffeine wear off in a few hours, you have even less energy than before. You feel sluggish again and reach for another cup of coffee to get a jump start. Over time, your body begins to rely on the caffeine at regular intervals in order to regulate your energy and metabolism.

Organizational Addictions

In organizational settings, addiction can take the form of a dependence on certain policies, procedures, departments, or individuals. The way we think about problems, or the policies that we pursue, can become addictions when we use them without

Continued on next page



Continued from previous page

consideration or choice, as an automatic knee-jerk response to a particular situation.

Hooked on Heroics

A common yet very subtle example of addiction in companies is "crisis management"—fire-fighting. Most managers say that they abhor fire-fighting because it wreaks havoc on normal work processes and makes it difficult to focus on the long-term. Yet fire-fighting is a way of life in most companies. Its pervasiveness and persistence is a clue that maybe it is part of an addictive structure.

Suppose you have a new product development project that has fallen behind schedule. The timing of its release is critical to its market success. In fact, the delays have reached crisis proportions. You decide to make it a high priority project and assign a "crisis

manager" to do what it takes to get that product out on time. This new manager suddenly has enormous flexibility in what he can do to get the product out. When the product is launched on time, he is touted as the hero of the day.

If we look at crisis management from the "Addiction" archetype, the symptomatic problem is the prevalence of crises that occur in the company (see "Hooked on Heroics"). When a crisis occurs, someone practices great heroism and "saves the day." The problem is solved and the person receives praise for doing a fine job. But what happens to the rest of the organization in the meantime? Oftentimes the solution causes a lot of disruptions which form the seeds of the next problems and perpetuate the crisis cycle.

The insidious side-effect of crisis management is that over time, as crisis management becomes the operating norm, managers begin to become dependent on the use of heroics—the need to have recognition and a feeling of accomplishment in an otherwise paralyzing institution. Usually there are roadblocks to taking action in the company: formalities and rules that say

"No, you can't do this," "You have to do it this way," or "We don't have the resources." Suddenly when there's a crisis, people are given tremendous freedom and leeway and are allowed to do what they couldn't do before. Once it's over, there is tremendous fanfare: the hero is rewarded or promoted. Over time, the company becomes addicted to continually creating crises, pulling the organization through

tremendous turmoil, and creating new heroes.

Breaking the Addiction Cycle

To identify "Addiction" dynamic at work, use the "Shifting the Burden" archetype as a diagnostic to ask questions such as: "What was the addiction responding to?", "Why did we feel a need to engage in this behavior or create this institution in the first place?", and "What are the problem symptoms that we were responding to?"

"Addiction" structures can be much more difficult to reverse than "Shifting the Burden" because they are more deeply ingrained. Just as you can't cure alcoholism by simply removing the alcohol, you can't attempt a frontal assault on an organizational addiction because it is so rooted in what else is going on in the company.

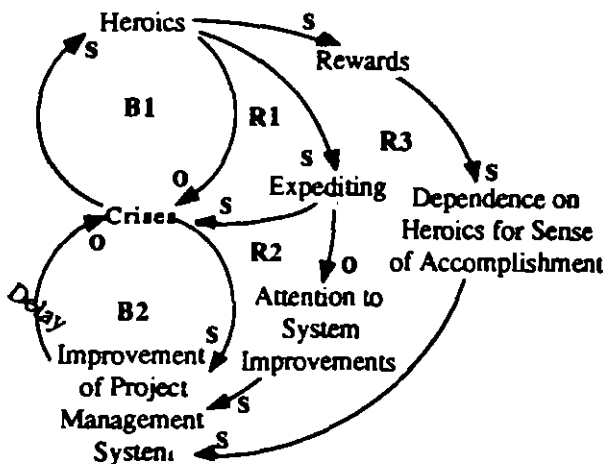
If your company is addicted to fire-fighting, declaring that there will be no more heroics may be the worst thing you can do. If heroics were the only way your organization knew how to release the accumulated pressures produced by ineffective processes, ending that practice may lead to an eventual explosion or systemic breakdown. To break the addictive pattern, you need to explore what it is about the organizational system that created the crisis and left fire-fighting as the only option.

Innovation

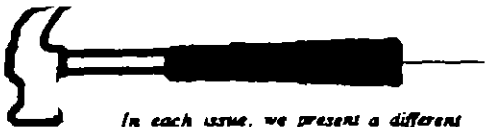
Is there such a thing as a benign or innocuous addiction? One could argue that some addictions are worse than others, and some may not be bad at all. The fundamental problem with any addictive behavior, however, is that it can lead an organization to become very myopic. The addictive solution becomes so ingrained that no other possibility seems necessary. Preventing corporate addictions requires the ability to continually see choices in a fresh way—to shun habitual responses.

The challenge for the learning organization is to get all the members of the organization to continually look at things with fresh eyes. That's the essence of discovery...and the essence of innovation.

Hooked on Heroics



Crisis can be solved either through short-term "heroics" (B1) or long-term improvements in management systems (B2). "Crisis management" tactics such as expediting projects not only propagate more crises, but they also take attention away from fundamental system improvements (loops R1 and R2). Over time, managers can become "hooked" on heroics to give them a sense of accomplishment in an otherwise disempowering situation (R3).

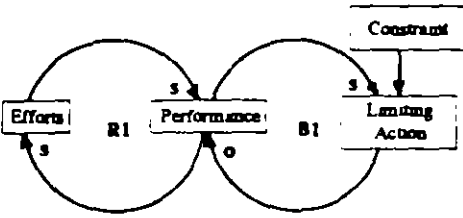
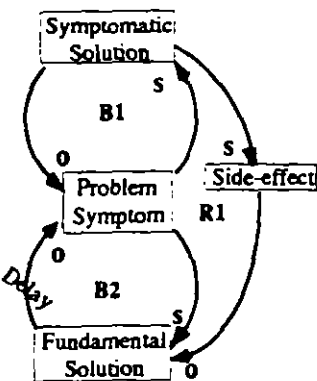
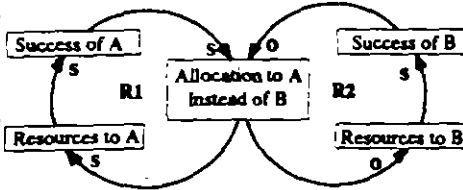
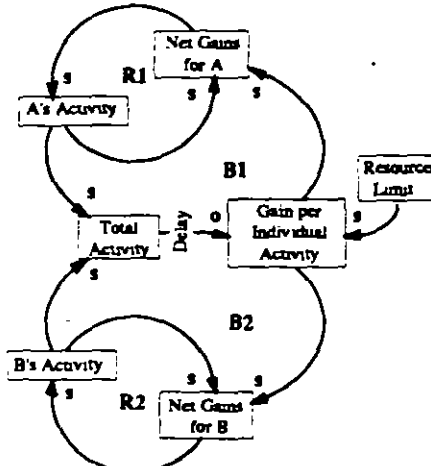


In each issue, we present a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. See page 10 for a symbol key for the diagrams.

Toolbox

Systems Archetypes at a Glance

Archetype	Description	Guidelines
<p style="text-align: center;">Drifting Goals</p>	<p>In a "Drifting Goals" archetype, a gap between the goal and current reality can be resolved by taking corrective action (B1) or lowering the goal (B2). The critical difference is that lowering the goal immediately closes the gap, whereas corrective actions usually take time. (See <i>Toolbox</i>, October 1990).</p>	<ul style="list-style-type: none"> • Drifting performance figures are usually indicators that the "Drifting Goals" archetype is at work and that real corrective actions are not being taken. • A critical aspect of avoiding a potential "Drifting Goals" scenario is to determine what drives the setting of the goals. • Goals located outside the system will be less susceptible to drifting goals pressures.
<p style="text-align: center;">Escalation</p>	<p>In the "Escalation" archetype, one party (A) takes actions that are perceived by the other as a threat. The other party (B) responds in a similar manner, increasing the threat to A and resulting in more threatening actions by A. The reinforcing loop is traced out by following the outline of the figure-8 produced by the two balancing loops. (See <i>Toolbox</i>, November 1991).</p>	<p>To break an escalation structure, ask the following questions:</p> <ul style="list-style-type: none"> • What is the relative measure that pits one party against the other and can you change it? • What are the significant delays in the system that may distort the true nature of the threat? • What are the deep-rooted assumptions that lie beneath the actions taken in response to the threat?
<p style="text-align: center;">Fixes that Fail</p>	<p>In a "Fixes that Fail" situation, a problem symptom cries out for resolution. A solution is quickly implemented that alleviates the symptom (B1), but the unintended consequences of the "fix" exacerbate the problem (R1). Over time (right), the problem symptom returns to its previous level or becomes worse. (See <i>Toolbox</i>, November 1990).</p>	<ul style="list-style-type: none"> • Breaking a "Fixes that Fail" cycle usually requires two actions: acknowledging that the fix is merely alleviating a symptom, and making a commitment to solve the real problem now. • A two-pronged attack of applying the fix and planning out the fundamental solution will help ensure that you don't get caught in a perpetual cycle of solving yesterday's "solutions."
<p style="text-align: center;">Growth and Underinvestment</p>	<p>In a "Growth and Underinvestment" archetype, growth approaches a limit that can be eliminated or pushed into the future if capacity investments are made. Instead, performance standards are lowered to justify underinvestment, leading to lower performance which further justifies underinvestment. (To be covered in an upcoming issue).</p>	<ul style="list-style-type: none"> • Dig into the assumptions which drive capacity investment decisions. If past performance dominates as a consideration, try to balance that perspective with a fresh look at demand and the factors that drive its growth. • If there is a potential for growth, build capacity in anticipation of future demand.

Archetype	Description	Guidelines
<p data-bbox="272 102 516 134">Limits to Success</p> 	<p data-bbox="641 117 1047 404">In a "Limits to Success" scenario, continued efforts initially lead to improved performance. Over time, however, the system encounters a limit which causes the performance to slow down or even decline (B1), even as efforts continue to rise. (See <i>Toolbox</i>, December 1990/January 1991).</p>	<ul data-bbox="1071 117 1518 500" style="list-style-type: none"> • The archetype is most helpful when it is used well in advance of any problems, to see how the cumulative effect of continued success might lead to future problems. • Use the archetype to explore questions such as "What kinds of pressures are building up in the organization as a result of the growth?" • Look for ways to relieve pressures or remove limits <i>before</i> an organizational gasket blows.
<p data-bbox="178 521 609 553">Shifting the Burden/Addiction</p> 	<p data-bbox="641 542 1047 925">In a "Shifting the Burden" archetype, a problem is "solved" by applying a symptomatic solution (B1) which diverts attention away from more fundamental solutions (R1). (See <i>Toolbox</i>, September 1990). In an "Addiction" structure, a "Shifting the Burden" degrades into an addictive pattern in which the side-effect gets so entrenched that it overwhelms the original problem symptom. (See <i>Toolbox</i>, April 1992).</p>	<ul data-bbox="1071 542 1518 925" style="list-style-type: none"> • Problem symptoms are usually easier to recognize than the other elements of the structure. • If the side-effect has become the problem, you may be dealing with an "Addiction" structure. • Whether a solution is "symptomatic" or "fundamental" often depends on one's perspective. Explore the problem from differing perspectives in order to come to a more comprehensive understanding of what the fundamental solution may be.
<p data-bbox="227 989 576 1021">Success to the Successful</p> 	<p data-bbox="641 989 1047 1340">In a "Success to the Successful" archetype, if one person or group (A) is given more resources, it has a higher likelihood of succeeding than B (assuming they are equally capable). The initial success justifies devoting more resources to A than B (R1). As B gets less resources, its success diminishes, further justifying more resource allocations to A. (See <i>Toolbox</i>, March 1992).</p>	<ul data-bbox="1071 989 1518 1244" style="list-style-type: none"> • Look for reasons why the system was set up to create just one "winner." • Chop off one half of the archetype by focusing efforts and resources on one group, rather than creating a "winner-take-all" competition. • Find ways to make teams collaborators rather than competitors.
<p data-bbox="219 1383 584 1415">Tragedy of the Commons</p> 	<p data-bbox="641 1415 1047 1766">In a "Tragedy of the Commons" structure, each person pursues actions which are individually beneficial (R1 and R2). If the amount of activity grows too large for the system to support, however, the "commons" becomes overloaded and everyone experiences diminishing benefits (B1 and B2). (See <i>Toolbox</i>, August 1991).</p>	<ul data-bbox="1071 1415 1518 1766" style="list-style-type: none"> • Effective solutions for a "Tragedy of the Commons" scenario never lie at the individual level. • Ask questions such as: "What are the incentives for individuals to persist in their actions?" "Can the long-term collective loss be made more real and immediate to the individual actors?" • Find ways to reconcile short-term individual rewards with long-term cumulative consequences.



Each Toolbox presents a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. See page 12 for a symbol key for the diagrams.

Mental Models & Systems Thinking: Going Deeper into Systemic Issues

by Richard Karash

In a causal loop diagram of a systemic issue, variables are connected in cause-and-effect relationships. But often the implicit thought processes behind those links are not well understood. How does a change in a teacher's expectations affect a student's performance? How does a change in the amount of money available for new product investment affect the flow of new products? Exploring the mental models behind such links helps us become clearer about the mechanisms that produce the observed behavior and can lead to better solutions.

Adding thought processes explicitly

to causal loop diagrams is one of a series of steps we call Going Deeper™. By mapping mental models onto a diagram, we can begin the process of exploring the more subtle aspects of the system.

The Steps of the Process

Going Deeper begins with a causal loop diagram of a systemic issue. Once the diagram is finished, the first step is to look for the links that represent human choice (as opposed to those that represent hard physical mechanisms). For example, if we have a link that says a change in revenues affects profits, we're dealing with arithmetic laws. But a link

between change in revenues and investments in R&D represents a process that involve quite a bit of human choice.

Once we have selected a link or two that represent human choice, we want to ask ourselves: Why is that choice being made? To explicitly represent the thought process, we add a thought bubble to the link. Like the thought bubbles in cartoons, which represent what the character is thinking but not saying, these thought bubbles represent the intangible thought processes that may or may not be visible to the people involved. When filling in the thought bubble, it is usually helpful to project ourselves into the situation and perhaps even role play it. The thought bubble should capture the line of thinking that makes the actions represented in the loop rational from each individual's point of view.

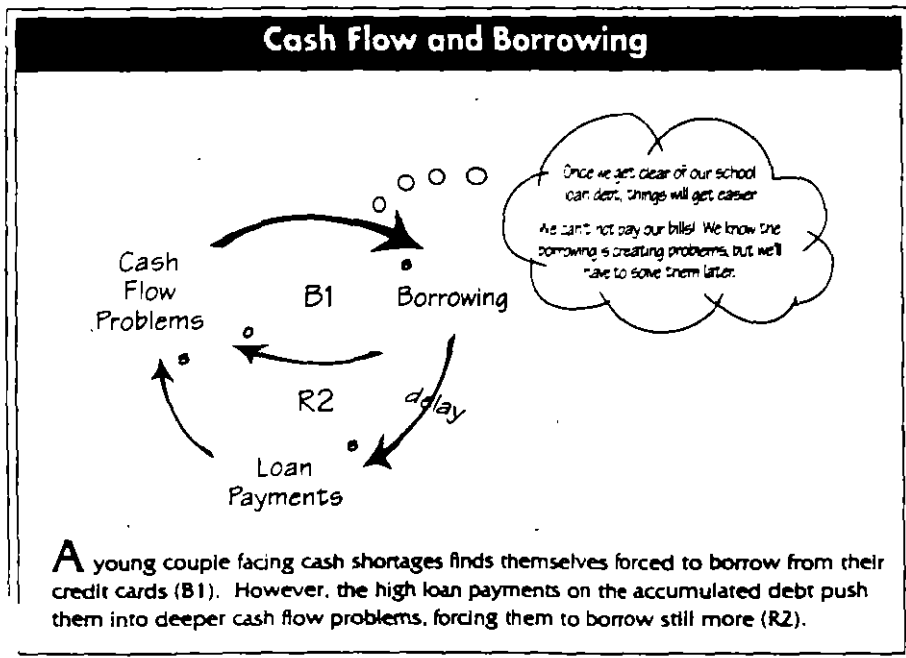
Borrowing Example

To see how the process works, let's look at the story of a young couple, Joan and Bob, who find themselves forced to borrow from their credit cards to get through a sequence of cash shortages. Unfortunately, the high interest and payments on their accumulated debt eventually pushes them into deeper cash flow problems, forcing them to borrow even more to stay afloat.

• Draw the causal loop diagram. In this story, as cash flow problems go up, borrowing goes up. As borrowing increases, the cash flow problems go down (B1 in "Cash Flow and Borrowing"). But over the long term, as borrowing goes up, loan payments go up, and cash flow problems increase (R2). This follows the "Fixes That Fail" storyline.

• Add a thought bubble to the link(s) that represent human choice. In this loop, human choice comes into play in the decision to "solve" the cash flow problem by borrowing, so we want to add the bubble to the arrow between "Cash Flow Problems" and "Borrowing."

• Presume rationality. To fill in the



thought bubble, we want to ask ourselves, "Assuming that these people are acting rationally from their point of view, what is the thinking that leads to the choice to take on extra debt?"

- *Suggest several possibilities.* Perhaps they might think they just need to get through this tough situation and things will get better afterwards ("Once we get clear of our school loan debt, things will get easier."). Or perhaps they feel they have no choice at this point ("We can't not pay our bills! We know the borrowing is creating problems, but we'll have to solve them later.").

- *Project the emotion of the situation into the thought bubble.* We might want to add, "What will the neighbors think if our car is repossessed!"

- *Capture multiple perspectives.* Perhaps Joan expects that their cash problems will get better once she earns her degree and enters the job market, whereas Bob is counting on that promotion the boss promised him "once he proved himself in the company." By projecting viewpoints from multiple perspectives, we can get a fuller sense of the situation.

The whole purpose of the Going Deeper process is to give visibility to non-obvious reasons why the system is functioning the way it is. If we leap to simple conclusions ("they should know better than to build up credit card debt") or blame systemic problems on individuals ("they're just not being responsible"), we may miss the larger learning that could come from a deeper analysis, and the ability to take that learning and apply it to other situations.

Software Development

In another setting, a manager in charge of a new release for a well-established software product envisioned a tidy package with some specific functionality. The plan was to develop it within a short time frame, using a small development team. In September, when they

began the project, the release was scheduled to ship in March, but was delayed to July when a few more features seemed necessary. In November it was delayed again to add even more features. In December the release was again rescheduled—a full 12 months later than originally planned!

To capture this story in a loop, we would say the more features, the longer the projected development time, which means the further out the projected delivery date. And it seems like the later projected delivery date is causing even more features to be added. But *how* is this occurring? (R3 in "Software Development Delay").

If we want to go deeper into the thought processes involved, we would put a thought bubble between "Projected Delivery Date" and "No. of Features," so we can explore *why* features are being added. One possibility may be that marketing thinks that the longer the wait, the higher the customers' expectations ("They won't think it was worth the wait. We'll look like turkeys!"). From the developers' perspective, the delay may be seen as an opportunity to experiment and add new features ("Now I have the time to put in that new XYZ feature I've always wanted to design.").

Sometimes the process of filling in the thought bubbles leads to additional variables that might be included in the diagram. For example, we might hypothesize that the longer the decision remains open, the more bugs are discov-

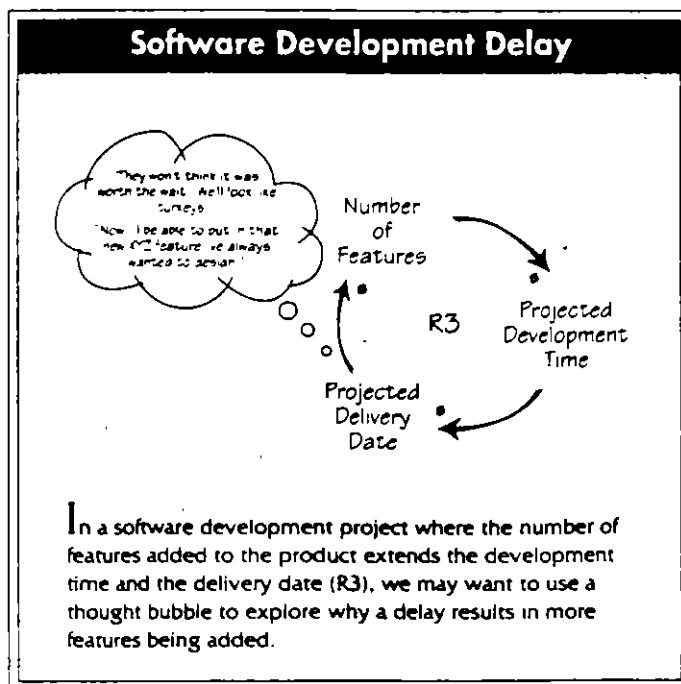
ered in the current release, and the more opportunities are identified for future improvements (another reinforcing loop). By continuing to dig deep into the thinking process, we may unearth systemic interconnections that were not obvious upon initial inspection of the problem.

From Understanding to Action

In dealing with complex situations, we want more than just *understanding*—we want to design effective actions. Examining our mental models and achieving deeper insights can propel a team towards action in a way that doesn't happen if you stop after drawing a causal loop diagram. By using both the systems thinking and mental model framework to explore a problem, we can more effectively move from superficial understanding to deeper understanding, thereby liberating action. ☐

Richard Karash is a member of the Systems Thinking Group at Innovation Associates, Inc. (Framingham, MA). He consults, trains, and coaches trainers in systems thinking. *Going Deeper™* is a trademark of Innovation Associates, Inc.

Editorial support provided by Colleen Lannon-Kim.



thought bubble, we want to ask ourselves, "Assuming that these people are acting rationally from their point of view, what is the thinking that leads to the choice to take on extra debt?"

- *Suggest several possibilities.* Perhaps they might think they just need to get through this tough situation and things will get better afterwards ("Once we get clear of our school loan debt, things will get easier."). Or perhaps they feel they have no choice at this point ("We can't not pay our bills! We know the borrowing is creating problems, but we'll have to solve them later.").

- *Project the emotion of the situation into the thought bubble.* We might want to add, "What will the neighbors think if our car is repossessed!"

- *Capture multiple perspectives.* Perhaps Joan expects that their cash problems will get better once she earns her degree and enters the job market, whereas Bob is counting on that promotion the boss promised him "once he proved himself in the company." By projecting viewpoints from multiple perspectives, we can get a fuller sense of the situation.

The whole purpose of the Going Deeper process is to give visibility to non-obvious reasons why the system is functioning the way it is. If we leap to simple conclusions ("they should know better than to build up credit card debt") or blame systemic problems on individuals ("they're just not being responsible"), we may miss the larger learning that could come from a deeper analysis, and the ability to take that learning and apply it to other situations.

Software Development

In another setting, a manager in charge of a new release for a well-established software product envisioned a tidy package with some specific functionality. The plan was to develop it within a short time frame, using a small development team. In September, when they

began the project, the release was scheduled to ship in March, but was delayed to July when a few more features seemed necessary. In November it was delayed again to add even more features. In December the release was again rescheduled—a full 12 months later than originally planned!

To capture this story in a loop, we would say the more features, the longer the projected development time, which means the further out the projected delivery date. And it seems like the later projected delivery date is causing even more features to be added. But *how* is this occurring? (R3 in "Software Development Delay").

If we want to go deeper into the thought processes involved, we would put a thought bubble between "Projected Delivery Date" and "No. of Features," so we can explore *why* features are being added. One possibility may be that marketing thinks that the longer the wait, the higher the customers' expectations ("They won't think it was worth the wait. We'll look like turkeys!"). From the developers' perspective, the delay may be seen as an opportunity to experiment and add new features ("Now I have the time to put in that new XYZ feature I've always wanted to design.").

Sometimes the process of filling in the thought bubbles leads to additional variables that might be included in the diagram. For example, we might hypothesize that the longer the decision remains open, the more bugs are discovered

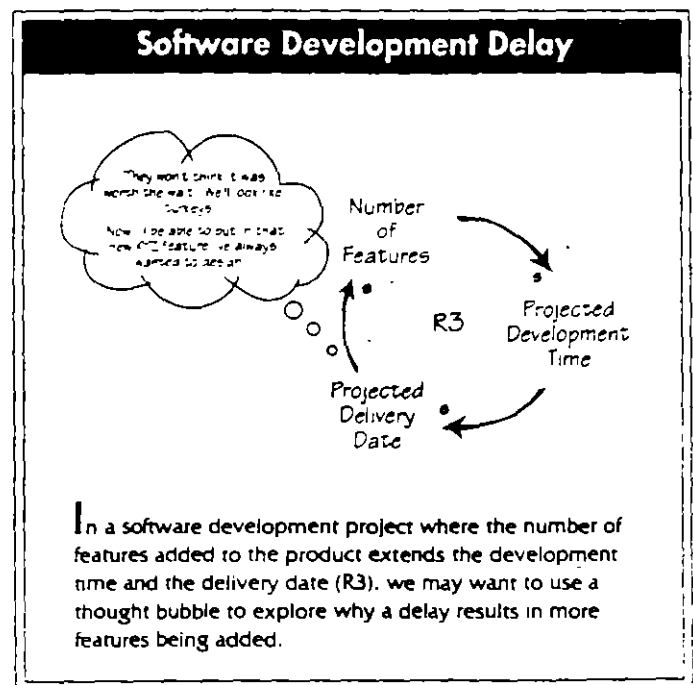
in the current release, and the more opportunities are identified for future improvements (another reinforcing loop). By continuing to dig deeper into the thinking process, we may unearth systemic interconnections that were not obvious upon initial inspection of the problem.

From Understanding to Action

In dealing with complex situations, we want more than just *understanding*—we want to design effective actions. Examining our mental models and achieving deeper insights can propel a team towards action in a way that doesn't happen if you stop after drawing a causal loop diagram. By using both the systems thinking and mental model framework to explore a problem, we can more effectively move from superficial understanding to deeper understanding, thereby liberating action. ■

Richard Karash is a member of the Systems Thinking Group at Innovation Associates, Inc. (Framingham, MA). He consults, trains, and coaches trainers in systems thinking. *Going Deeper™* is a trademark of Innovation Associates, Inc.

Editorial support provided by Colleen Lannon-Kim.





Each Toolbox presents a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. See page 12 for a symbol key for the diagrams.

Six Steps to Thinking Systemically

by Michael Goodman and Richard Karash

Bijou Bottling Company is a fictitious beverage bottler with an all too real problem: chronic late shipments. Its customers—major chain retailers—are looking for orders shipped complete and on time. About five years ago, in a U.S. region covering about six states, this problem reached crisis proportions...

In the face of day-to-day pressures, groups often leap to solutions after only a modest amount of brainstorming. A systemic approach, however, provides a structured problem-solving process for digging deeper into our most vexing problems.

To get a sense for how systems thinking can be used for problem identification, problem solving, and solution testing, we have outlined a six-step process. To use this process on a problem in your workplace, try the worksheet on page 8.

1. Tell the Story

The starting point for a systems thinking analysis is to get your head above water enough to start thinking about the problem instead of just acting on it. An effective way to do this is to gather together all of the important players in the situation and have each one describe the problem from his or her point of view.

At Bijou Bottling Company, the problem was usually a customer complaint: "Where were the 40 cases of 2-litre Baseball tie-in product that were ordered last week?!" Somehow Bijou would get the goods there on time, whatever it took—including airshipping heavy soda in glass bottles at enormous costs. But this crisis management led to a culture where people built their careers on coming in at the 11th hour and turning around a customer complaint.

2. Draw "Behavior Over Time" Graphs

In the storytelling stage, most of the energy is focused on the pressures of the current moment. When we move to "Behavior Over Time" (BOT) graphs, however, we begin to connect the present to the past and move from seeing events to recognizing patterns over time.

Draw only one variable per graph on a Post-it™ note so it can be easily moved around in the steps that follow. The time frame should span from past up to the present—but it can also include future projections (see "Bijou Over Time").

3. Create a Focusing Statement

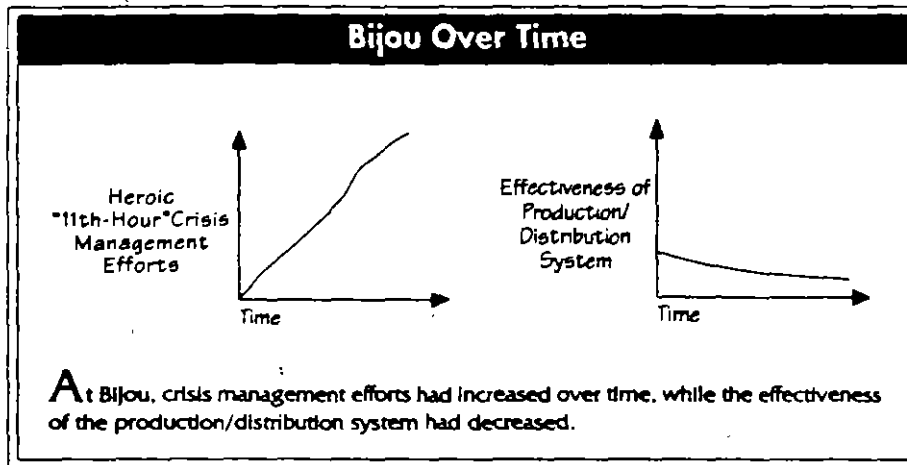
At this point, you want to create a statement that will help channel energy during the rest of the process. This statement may involve a picture of what people want, or a question about why certain problems are occurring. At Bijou, for example, the focusing statement was: "We're pretty good at solving each problem as it arises. But why are these problems recurring with greater frequency and intensity? What is causing them?"

4. Identify the Structure

You now want to describe the systemic structures that are creating the behavior patterns you identified. The systems archetypes are an easy way to begin building a theory of why and how things are happening (see "Systems Archetypes at a Glance," May 1992).

Begin by reviewing the story, graphs, and focusing statement to see if they follow the storyline of an archetype. If so, draw the loop diagram for that archetype, place the Post-its of the variables in the diagram, and move them around on a flip chart until you have a diagram that seems to capture what is going on.

The group at Bijou decided that



their problem matched the "Shifting the Burden" storyline, in which a problem is "solved" by applying a short-term solution that takes attention away from more fundamental improvements. They identified a balancing loop that described how customer problems were solved with heroic "11th-Hour" efforts (the symptomatic solution) at the expense of improvement and redesign of the production/distribution system (the fundamental solution). As people "learned" over time that heroism is rewarded, their willingness and ability to address system-wide problems decreased (see "Shifting the Burden to Heroism").

5. Going Deeper™ Into the Issues

Once you have a reasonably good theory of what is happening, it is time to take a deeper look at the underlying issues in order to move from understanding to action. There are four areas you should clarify:

Purpose of the System. Ask yourself, "In the larger context, what do we really want here?"

Mental Models. Begin the exploration of mental models by adding "thought bubbles" to those links in the diagram that represent choices being made (see "Mental Models and Systems Thinking: Going Deeper into Systemic Issues," February 1995).

The Larger System. Add links and loops to enrich the story and connect the relationships to the larger system.

Personal Role. Acknowledge and clarify your own role in the situation.

For example, when the people at Bijou looked at the larger system, they wondered what role their customers played in the system. They theorized that customers were taking problem situations and escalating them into crises in order to get the company's attention (B4).

6. Plan an Intervention

When planning an intervention, use your knowledge of the system to design a solution that will structurally change it to produce the results you want. This might take the form of adding a new link or loop that will produce desirable behavior, breaking a link or loop that produces undesirable behavior, or a combination of the two. The most powerful interventions often involve changing the thinking of the people involved in the system.

At Bijou, the key to change was realizing that the problems were largely self-inflicted. They realized that they had to make progress on production/distribution system improvements while still doing enough fire-fighting to keep things afloat. In the longer term, they would need to change the reward systems that promoted heroic behavior. They also recognized the need to sustain the improvement efforts even when the pressure came off—otherwise the problems would be back again soon.

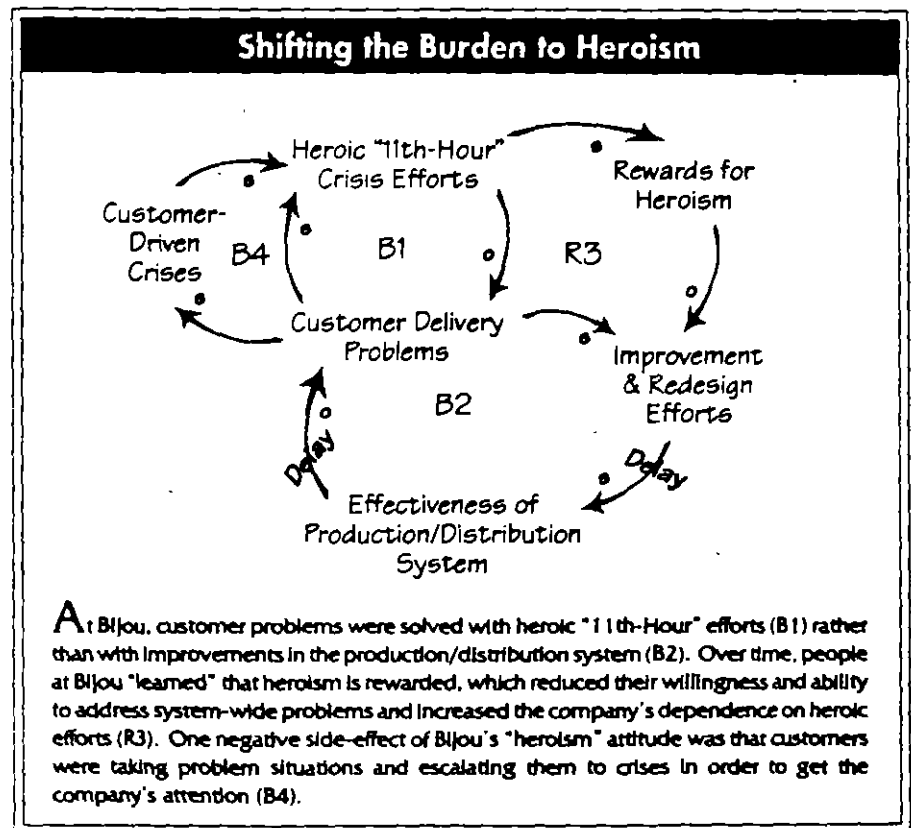
Part of a Cycle

Even as systems thinkers, it is easy to fall back into a linear process. But learning is a cycle—not a once-through process with a beginning and an end. Once you have designed and tested an intervention, it is time to shift into the active side of the learning cycle. This process includes taking action, seeing the results, and then coming back to examine the outcomes from a systemic perspective.

Michael Goodman is vice president of Innovation Associates, Inc. (Framingham, MA), and heads IA's Systems Thinking Group. He is a frequent contributor to *The Systems Thinker*. Richard Karash is a member of the Systems Thinking Group at Innovation Associates, Inc. where he consults, trains, and coaches trainers in systems thinking.

Post-It™ Notes is a trademark of the 3M Corporation. Going Deeper™ is a trademark of Innovation Associates, Inc.

Editorial support for this article was provided by Colleen Lannon-Kim.



"Six Steps" Worksheet

1. Tell the Story:

2. Draw Graphs:

3. Create a Focusing Statement:

4. Identify the Structure:

How does the structure explain the observed behavior?

5. Going Deeper™ into the Issues:

- Purpose:
- Mental Models:
- Larger System:
- Personal Role:

6. Plan an Intervention:

How is the planned intervention consistent with the structure as described above?

Some generic structures occur so frequently that they have been identified as a generic set of tools called "systems archetypes." This column is designed to help readers recognize archetypes at work in newspaper and magazine articles.

Car Leasing: Are Automakers Gambling Away Their Future?

by Kellie T. Wardman

"Ford and some other companies that have pushed [car] leasing hard 'are making money hand over fist right now,' says Randall McCahren, whose company, Bank Lease Consultants Inc., tracks industry trends. Yet some analysts and auto executives worry that the leasing business could make a dramatic U-turn and become a money-loser in two or three years. That's an immediate concern, because the aggressive lease deals being written today could undermine profits several years down the road. By 1997, most analysts agree, the ever cyclical auto market will likely head south again—maybe sooner, if the Federal Reserve keeps hiking interest rates.... Moreover, these good-quality used cars may undermine new-car sales, thereby magnifying the downturn" ("A High-Stakes Spin of the Wheel," *Business Week*, December 19, 1995).

...

In the highly competitive auto industry, car manufacturers are under continual pressure to create more attractive deals. The most recent trend is to offer cut-rate leases on new cars, which helps in the short term by pumping up unit sales and total revenues. The long-term side-effects of cut-rate leases, however, are slowly beginning to emerge.

One problem facing many car manufacturers concerns the "residual values" of the leased cars. Residual values—the car's anticipated wholesale value at the end of the lease—are set

when a company writes a new lease. The process itself presents no inherent problems to the carmaker's bottom line, as long as the actual market value of the car equals its anticipated residual value at the end of the lease. Unfortunately, the actual market price is falling short of expectations—which means losses for carmakers when it comes time to sell the previously leased cars. One ex-

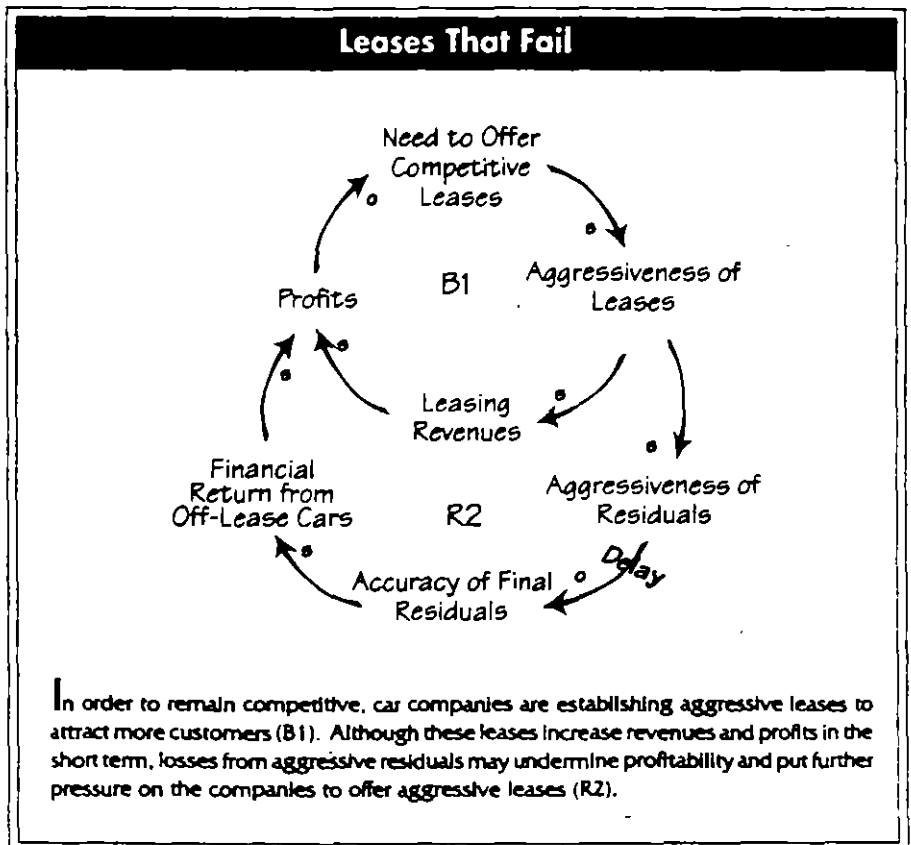
ample cited by *Business Week* is Nissan Motor Co.'s Infinity luxury brand. It "set an unrealistically high three-year residual on its Q45 model in the early '90s. Last year, when the cars began coming off lease, Infinity had to sell them at losses ranging from \$5,000 to \$7,000 apiece, according to some analysts."

In addition, according to *Business Week*, residual values are currently at an all-time high and may be poised to fall. This means that car manufacturers and independent lease companies may be facing tremendous losses as their cars come off lease.

Leases That Fail?

The struggles around residual values suggest that the car companies are caught in a "Fixes That Fail" structure. In "Fixes That Fail," a problem symptom cries out for resolution. To alleviate it, a solution is quickly implemented that alleviates the symptom in the short

Continued on next page



C *Continued from previous page*
 term. Over the long term, however, the unintended consequences of the "fix" exacerbate the problem symptom and actually make the situation worse.

In order to stay competitive, car companies are establishing aggressive leases to attract more customers (B1 in "Leases That Fail"), which increases revenues and profits in the short term. Losses from these aggressive residuals may undermine future profitability.

According to the *Automotive Lease Guide* (ALG), which helps the industry set residual values, the average 1995 model car or truck will retain 57% of its sticker price two years from now. This figure is the highest since 1986—and according to *Business Week*, ALG publisher Doug Aiken has been warning the car firms not to "get too far out on a limb with residual values." This warn-

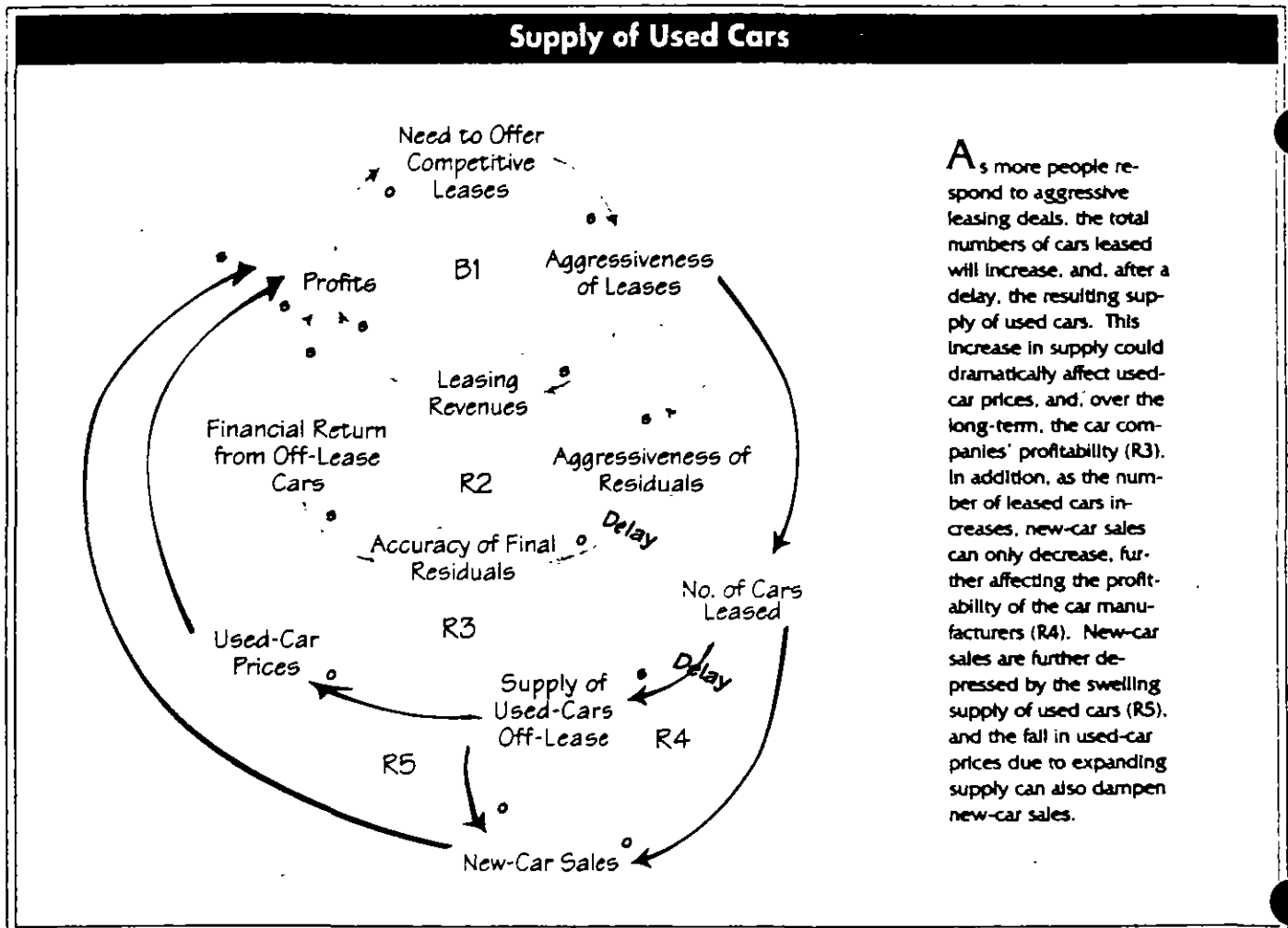
ing has not kept automakers from setting their residuals higher than those recommended by ALG in order to attract customers. For example, *Business Week* cited one independent auto lease company that estimated that the Honda Civic would be valued at 75% of its sticker price two years from now, compared with ALG's estimate of 61%. As the aggressiveness of residual values increases, the likelihood of their accuracy decreases, which means lower profits in the long run (R2).

Second-Order Consequences

In addition to the residual values issue, the success of the leasing agreements may create problems for car companies further down the road. The key to profitability in car leasing lies in the market for used cars. As the leases become more aggressive and the offers more ap-

pealing, however, more people will respond to the leasing deals, increasing the total numbers of cars leased, and, after a delay, the resulting supply of used cars (see "Supply of Used Cars"). According to CNW Marketing/Research, when the expected downturn in the auto market comes, "The automakers will face an unprecedented situation: A record 2.7 million cars and light trucks will be coming off lease, up from 490,000 in 1993." This increase in supply could dramatically affect used-car prices, and, over the long-term, the car companies' profitability (R3).

In addition, as the numbers of cars leased increases, new-car sales can only decrease, further affecting car manufacturers' profitability (R4). New-car sales are further depressed by the swelling supply of used cars (R5), and the fall in used-car prices due to expanding supply



As more people respond to aggressive leasing deals, the total numbers of cars leased will increase, and, after a delay, the resulting supply of used cars. This increase in supply could dramatically affect used-car prices, and, over the long-term, the car companies' profitability (R3). In addition, as the number of leased cars increases, new-car sales can only decrease, further affecting the profitability of the car manufacturers (R4). New-car sales are further depressed by the swelling supply of used cars (R5), and the fall in used-car prices due to expanding supply can also dampen new-car sales.

can also dampen new-car sales.

Toward More Fundamental Solutions

To successfully manage a "Fixes That Fail" dynamic, we must first acknowledge that the fix is merely alleviating a symptom. Then we must commit to solving the real problem *now*. That often means continuing to apply the fix while simultaneously working out a plan for a more fundamental solution.

For the car companies, this may mean continuing to offer aggressive lease deals to remain competitive in the short term while addressing the deeper underlying issues around product attractiveness and profitability. The car companies must also address the current situation of falling residual values and the impending glut of used cars. Some companies, for example, are working on exporting a portion of their used-car supply to create a new demand for them overseas. Others are developing programs to lease the used cars—or at the very least, reconditioning and adding extended warranties to enhance the sales of off-lease cars.

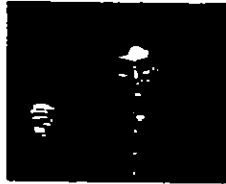
Despite those strategies, *Business Week* maintains that "the used-car market is probably too vast for individual companies to influence." Given the additional complexity of this market, the decision to go so heavily into the leasing business may have far too many negatives in the long run to make it a worthwhile strategy. Car companies may want to reconsider their leasing programs in the context of the larger systemic consequences and see if there may be more effective ways to increase sales and profitability. ●

Kellie T. Wardman is publications director at Pegasus Communications and an editor of *The Systems Thinker*.



THE RESOURCE LIBRARY

The following resources are available as part of *The Learning Organization Resource Library™* catalog, offered by Pegasus Communications, Inc. For more information, call (617) 576-1231.



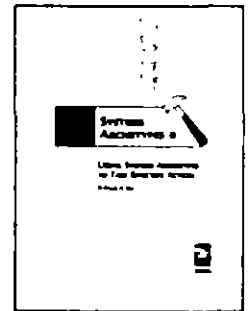
Sustainable Communities: A Management Challenge by Fritjof Capra

In this 60-minute video, Fritjof Capra describes how lessons from nature can assist in our efforts to create learning communities. Understanding the principles of ecology and how systems organize themselves will help us build sustainable human communities—whether they are educational, corporate, spiritual, or other. Capra also explains how a shift from looking at objects to seeing relationships can enable our organizations to become self-correcting systems. Order #V9403 \$125

Systems Archetypes II by Daniel H. Kim

More than just a "how-to" guide, *Systems Archetypes II* provides a grounded approach to problem diagnosis and intervention that can enable more effective action.

The articles included in this booklet show how to use the archetypes to tackle complex issues in at least three different ways. First, the archetypes can be used as diagnostic tools for developing an understanding of a current situation. Second, as planning tools, they can help us anticipate and plan for future consequences. Third, the archetypes can be used as theory building tools to help build a growing body of knowledge about our understanding of the world. Learn to apply the archetypes to specific business situations by following the 7-step guides in this booklet. Order #TR-SA02 \$15



Systems Thinking in Action™: The Video Collection Vol. II: A Framework For Change

Included in this 30-minute preview video are short segments from the following tapes:

PETER M. SENGE: *Building Learning Infrastructures*

JOHN D. STERMAN: *System Dynamics: The Foundation of the Learning Organization*

FRITJOF CAPRA: *Sustainable Communities: A Management Challenge*

FRED KOFMAN: *Revealing the Heart of the Learning Organization*

DANIEL QUINN, PETER SENGE, MARGARET J. WHEATLEY: *The Purpose of Business in the 21st Century: A Dialogue*

Order # PRV002 \$14.95



TOOLBOX

Each Toolbox presents a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. A symbol key for the diagrams is on page 12.

Going Deeper™: Moving from Understanding to Action

by Richard Karash and Michael Goodman

The March 1995 Toolbox column outlined a six-step process for applying systems thinking to a specific issue ("Six Steps to Thinking Systemically," V6N2). One of the steps was listed as "Going Deeper™ into the Issues." In this follow-up article, Richard Karash and Michael Goodman explore the Going Deeper approach in more depth, explaining each of the steps of this process and illustrating how it can provide a more focused insight into analyzing a systemic problem.

Once a group has created a systems diagram that reasonably explains the problematic behavior they have been experiencing, they often ask, "Now what do I do?" To move things forward at this point, it is

usually necessary to bring in the other learning disciplines—mental models, team learning, personal mastery, and shared vision—to augment the systems thinking approach. The generative questions provided by applying the

other disciplines to a problem can help a team move from *understanding* to *action*. After all, the aim of any systems thinking analysis is not just to gain better understanding, but to improve the situation as well.

The Going Deeper™ process is a 4-step model for taking a deeper look at underlying issues in order to create an intervention strategy (see "Going Deeper™ Questions"). Below, we explain each step, revisiting the Bijou Bottling case that was introduced in the March article.

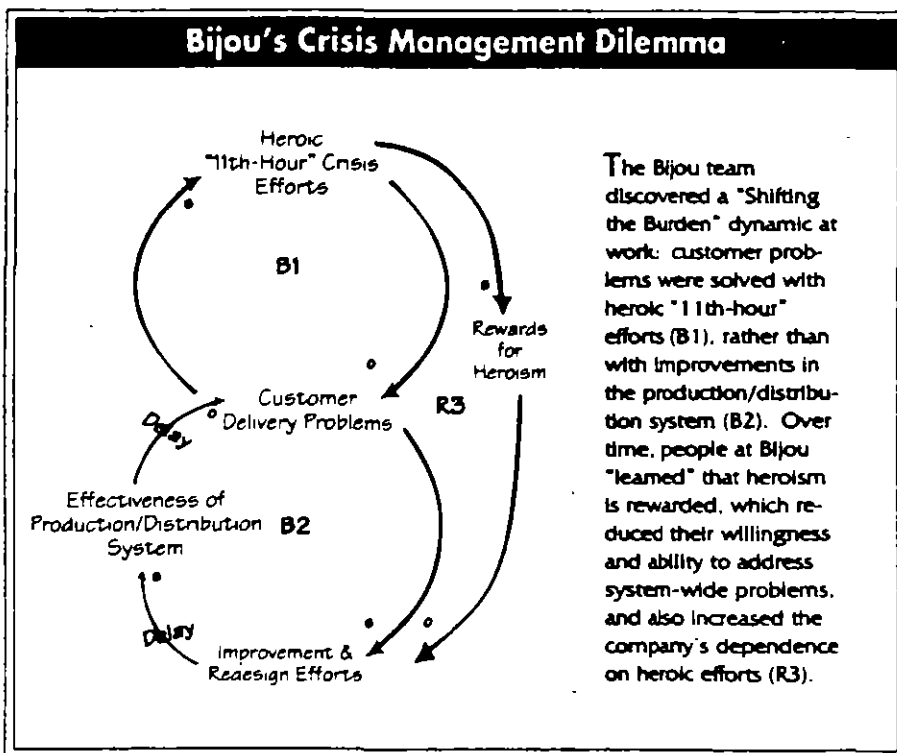
Preliminary Work

Before beginning the Going Deeper process, you should have developed a causal loop diagram that reasonably explains a problem or issue you are trying to understand. For example, at Bijou Bottling, the team was struggling with chronic late shipments. When they mapped out the problem, they discovered a "Shifting the Burden" archetype at work: the company's commitment to superior customer service resulted in heroic, 11th-hour interventions to move shipments out. Over time, however, the company became addicted to crisis management. Not only did this take attention away from more fundamental system-wide improvements, but it also contributed to more crises and reinforced the perceived need for crisis management (see "Bijou's Crisis Management Dilemma").

Once you have captured the relevant causal feedback loop structures, as Bijou did in identifying an archetype, you can then begin the Going Deeper process listed below. It is likely, however, that you will cycle back to do more diagramming, since this is an iterative process.

1. Explore the Purpose

Having a clear purpose in mind is essential for any systems thinking effort to be successful. Many of us have experienced the power of someone asking the



question, "What are we *really* trying to do here?" The Going Deeper process helps to remind us of the larger vision by asking two questions: (1) in the broadest context, what results do we really want from this system? and (2) compared to that goal, what results are we getting now? Answering these questions clarifies the tension between vision and current reality.

In the Bijou case, the team said, "What we really want is for all this to run smoothly—for us to be a real team together, and to be delighting our customers. We also want our work to be more fun and not so hard on the people here. Right now, we're consumed in fire-fighting, crisis management, and internal conflict."

2. Examine Mental Models

Mental models are important elements

of the structure that may be producing the undesirable results. One way to surface underlying mental models is to ask, "Assuming that everyone in the system is acting reasonably and responsibly from their point of view, what is the thinking (beliefs, assumptions, rationale) that leads to the choices being made?" You can then map these mental models on your causal loop diagram by adding "thought bubbles" to the links that represent those choices (for more on this process, see "Mental Models & Systems Thinking: Going Deeper into Systemic Issues," February 1995).

At Bijou, there were widely held mental models around issues of commitment, "hustle," and doing "whatever it takes" to make things right for the customer. People would stop at nothing when goods were needed for a special promotion. Along with this focus came

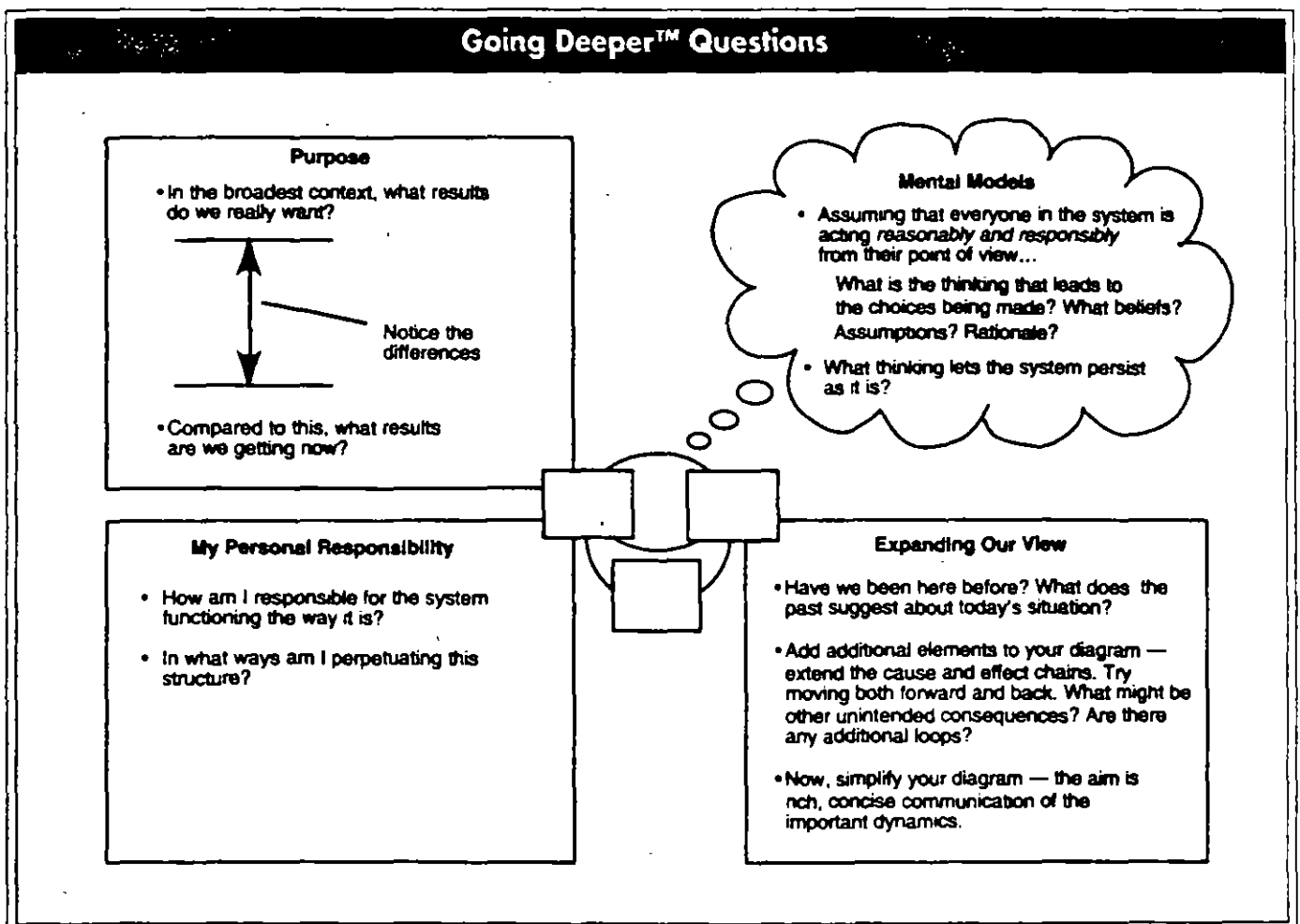
a more subtle message, however, which might be expressed as, "Don't bother me unless it's important—unless it's a crisis!" This unspoken assumption made it hard to get people to focus on long-term improvements in operating processes and systems.

3. Acknowledge Personal Responsibility

It's so much easier to see how other people are creating problems than it is to see our own role in them. Therefore, we always need to explore our personal responsibility for the problems we are seeing. What are we doing that might be perpetuating the current problem?

At Bijou, people began to see that they were contributing to the "Shifting the Burden" structure by devoting all

Continued on next page 



Continued from previous page

their energy to problem-solving, rather than dealing with the underlying operational issues. They all assumed that "someone else" would take care of the deeper problems. By seeing the problem as coming from other departments, they had been blocking the very cross-functional teamwork that was necessary to improve the situation. They had come to believe in their powerlessness to affect the "system," when in fact their own involvement was the only route to improving it. Moving away from powerlessness toward a creative orientation is a fundamental requirement for making change.

4. Expand the View

Before designing and implementing an action plan, it is important to examine the situation in terms of the larger system. First, it is helpful to see the problem in a historical context, by looking forward and backward in time. Too often, a problem situation is viewed as a snapshot—a unique moment in time,

disconnected from the past or future. To get a sense for the historical context, try drawing graphs of the important behaviors. Ask yourself, "Have we been here before?" Then project into the future the type of behavior you expect to see if your intervention is successful. What will be some clear signals that your efforts are working?

Another way to see the larger system is to reexamine the causal connections in the original diagram and see if additional elements can be added to the loops. Look especially for potential unintended consequences that might result from the proposed solutions. At Bijou, looking at the larger system helped them realize that customers had learned how to work the system—they quickly learned that in order to get attention, they needed a crisis, so they became adept at building "fires" (loop B4 in "Customer-Driven Crises"). Managers at Bijou also realized that all of the expensive fire-fighting was creating cost problems, which diverted resources away from system improvement

and created even more internal friction.

Redesigning the System for Improvement

The purpose of the Going Deeper questions is to move toward action, seeking a change in the system that will produce better results. People also generally want an improvement that will be self-sustaining and will change the pattern of behavior over time—not just a one-time blip that fades away.

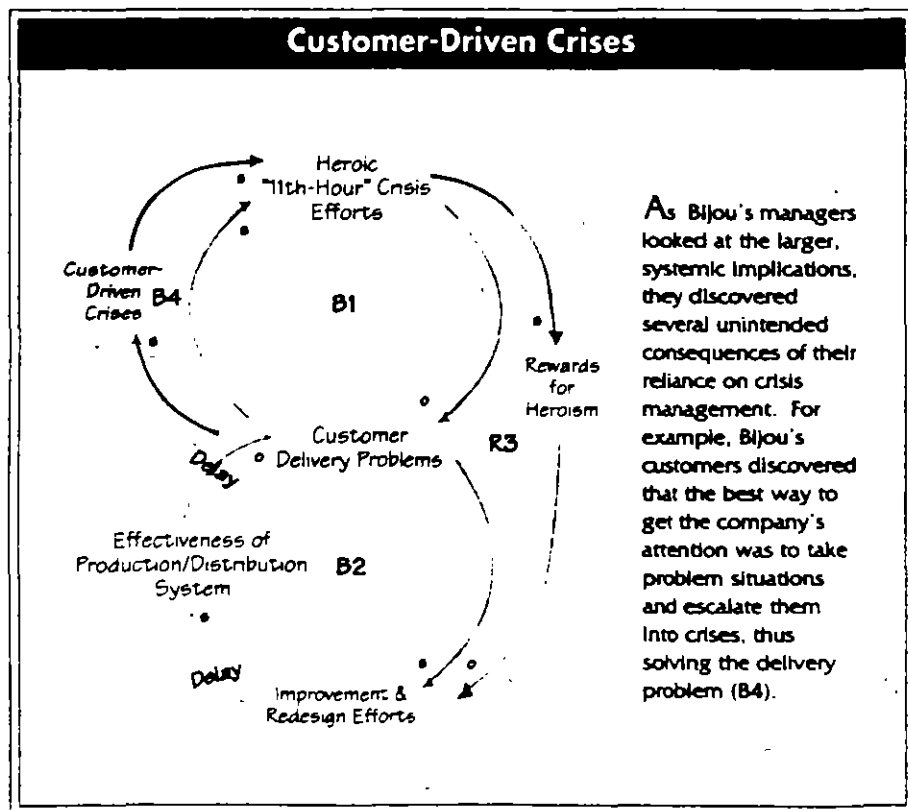
At Bijou it was clear that in order to solve their delivery problems they needed to weaken the crisis loop (B1) and strengthen the improvement loop (B2). Accomplishing this would require a change in attitude—a shift away from fire-fighting toward really working on the system. It would also require a sense of personal responsibility and involvement on everyone's part—no more hiding behind the belief that change is "someone else's job." And it would take real understanding of how the system works, which could be gained by mapping the relevant processes through additional loop diagrams.

Systems thinking is not just a technical discipline or a "silver bullet" that can make a problem simply disappear. Used in conjunction with the other learning disciplines, it is a powerful tool for individual, team, and organizational change. The Going Deeper questions seek to bring together aspects of all the learning disciplines in order to move a team from understanding to action.

Richard Karash is a consultant, facilitator, and trainer in the disciplines of the learning organization and an adjunct staff member at Innovation Associates, Inc. (Framingham, MA).

Michael Goodman is vice president of Innovation Associates, Inc., and heads IA's Systems Thinking Group. Louis van der Merwe of IA South Africa also contributed to the methods described in this article.

Going Deeper™ is a trademark of Innovation Associates, Inc.





Each Toolbox presents a different systems tool using relevant business examples. Readers are encouraged to practice using these tools by applying them to issues of personal interest. See page 12 for a symbol key for the diagrams.

ENTRY POINTS TO MODELING: Listening for "Dilemmas"

by Don Seville

This is the first of several articles in the "Entry Points to Modeling" series that explore when simulation modeling is most appropriate.

One of the biggest contributions that systems thinking can make is to help managers build theories about why things happen the way they do. Actually testing those theories requires tools such as computer simulation models, which enable you to see how different assumptions play out over time. But because creating a model can be expensive and time consuming, it realistically cannot be applied to every issue. So how can you determine when simulation is appropriate?

When to Simulate

In general, simulation modeling is useful for understanding complex relationships, for developing and testing specific policies, and for understanding the implications of long time delays on a problem or issue (see "When to Simulate," April 1995). While these general guidelines give you a sense of why modeling can be useful, they don't tell you when a particular problem you're working on can benefit from simulation modeling.

That is why experienced practitioners watch for certain signals that indicate when it is time to move into simulation modeling. One such entry point is to listen for organizational "dilemmas."

A "dilemma" occurs when a group is aware of multiple consequences of a policy or strategy, but there is no clear agreement around which consequence is strongest at what point in time. In such a situation—where there are strong passions around a specific issue and the organization or team is "stuck" at its current level of understanding—simulation modeling has the potential to be very effective. The following six-step process describes how modeling can be used to resolve such dilemmas.

1. Identify the Dilemma

The first step is to listen for situations where there are two different theories about the consequences of a decision (for more on this process, see "Using Systems Thinking 'On-Line': Listening for Multiple Hypotheses," August 1995). Dilemmas are often characterized by one party strongly advocating for a decision or strategy, followed by public disagreement or private mumbling about how that action will cause just the opposite of the intended outcome.

For example, the plant management team of a major component supplier wanted to "load the plant"—that is, continually push sales in order to maximize the plant's capacity usage. But others expressed concern that load-

ing the plant might create other problems that would affect both quality and service.

2. Map the Theories

Once you have identified the dilemma, it is important to clarify the issues involved by explicitly mapping out each of the viewpoints using causal loop diagrams or systems archetypes. For example, in the plant capacity issue, the strategy behind loading the plant was to increase sales in order to maximize the capacity utilization of the plant, until the available capacity falls to zero and no more sales can be filled (B1 in "Theory 1: Loading the Plant"). The benefit of this strategy is twofold: by increasing sales, the plant will boost profits (R3); and by increasing the capacity utilization, the cost per unit will fall, which increases profits and allows more investment in plant capacity (B2).

However, other people felt strongly that if plant utilization remained too high, the ability to respond to customer changes (flexibility) would go down, eventually hurting the company's reputation as a supplier and leading to a decline in sales (B4 in "Theory 2: Unintended Side-Effects"). They also believed that loading the plant would cause the stress level in the organization to rise, eventually eroding quality and further hurting their reputation (B5).

3. Assess Dynamic Complexity

Some situations of "multiple hypotheses" resolve themselves when the different parties work together to map out each story and find that one is clearly more accurate. But in cases where both sets of interconnections are plausible and the uncertainty or disagreement still exists, further work is needed to resolve the dilemma. If the uncertainty is simply around a number (such as accurate cost data) or the probability of one outcome versus another, it is a static problem and techniques such as decision analysis may be appropriate.

Simulation modeling, on the other hand, is most effective where there is some degree of *dynamic complexity*—where the link between cause and effect is subtle and the implications over time are not obvious.

To check for dynamic complexity, ask if the uncertainty or disagreement is around what feedback loop is dominant at what time—in other words, the long-term impact might be different than the short-term effects. If this is the case, then some degree of dynamic complexity exists and it is appropriate to move into simulation modeling. In the manufacturing example, a key area of uncertainty was the effect of various capacity utilization strategies on the company's reputation (and therefore sales) over time. The importance of this time delay indicated a degree of dynamic complexity that could benefit from a simulation approach.

4. Developing the Simulation Model

At this point, you are now ready to move into the development of the model. First, you want to focus the model-building effort by asking, "What do we need to learn in order to be able to resolve the dilemma?" Stating the objective up front will guide the rest of the process.

Once you have established your objective, you can define the boundaries of the simulation model by identifying the key decisions (critical policies that the organization makes), the important indicators (what you need to see from the system to assess the decision), and the uncertainties (most fragile assumptions about the relationships or outside world) associated with the dilemma.

In the capacity utilization example, the parties recognized that the dilemma would be resolved when they knew both the short- and long-term impact of different capacity utilizations on sales and profits. Their primary decision was to select a particular utilization goal (de-

sired production relative to capacity), which could be assessed by looking at the long-term behavior of sales and reputation. The company's key uncertainties included demand and customer sensitivity, because they didn't know exactly how the market would evolve, and the impact of different utilizations depended heavily on their assumptions about the customer's sensitivity to price, quality, and flexibility.

Once you have established the focus and boundaries of the model, you are ready to build the simulation model by defining the relationships between important variables, such as the relationship between manufacturing flexibility and reputation in our plant capacity example. (For more on the actual mechanics of model building, see "From Causal Loop Diagrams to Computer Models—Part II," August 1994).

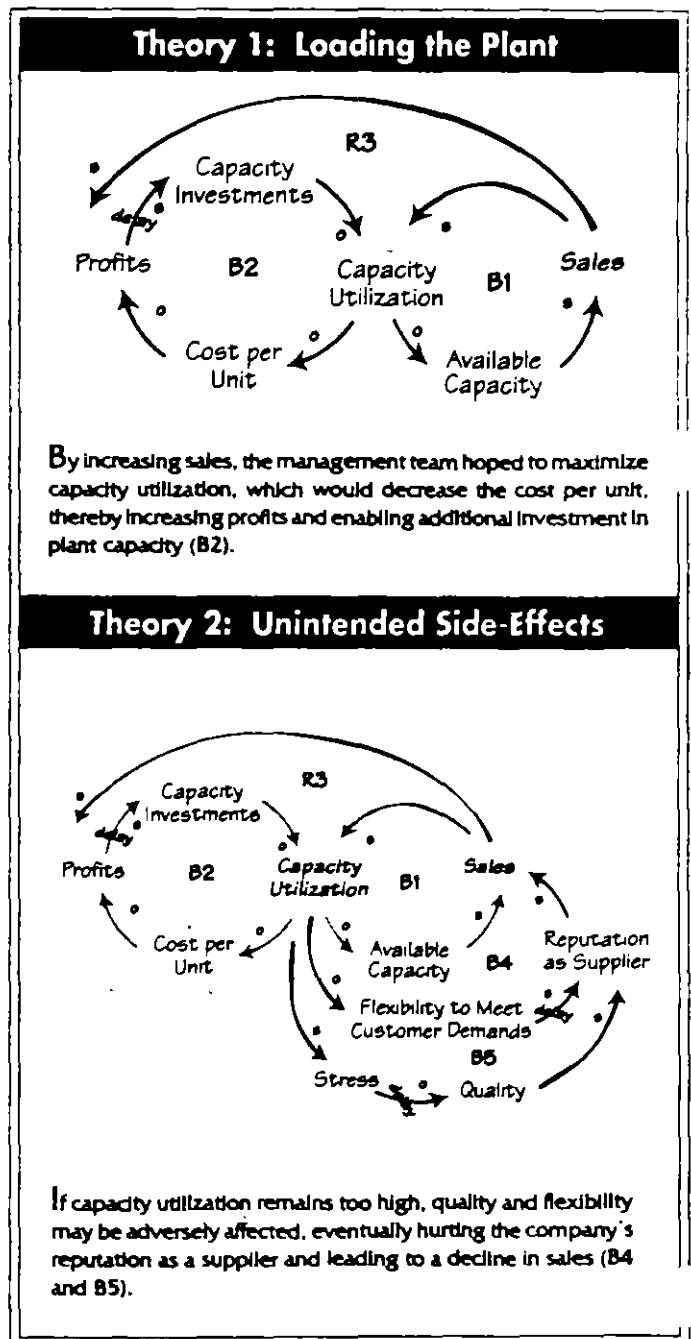
5. Divergence: Testing the Assumptions

Once you have built the model, you can begin testing the different assumptions behind your causal theories to see the effect of those interrelationships over

time. At this stage, you are attempting to be *divergent*—trying out many possible scenarios, any of which can lead to new questions and experiments.

Using the simulator they had developed, the management team was able to test different capacity utilization levels, using various assumptions about the market and customer "sensitivity." In the first scenario, the group made two simplifying assumptions: that there

Continued on next page 



By increasing sales, the management team hoped to maximize capacity utilization, which would decrease the cost per unit, thereby increasing profits and enabling additional investment in plant capacity (B2).

If capacity utilization remains too high, quality and flexibility may be adversely affected, eventually hurting the company's reputation as a supplier and leading to a decline in sales (B4 and B5).

Continued from previous page would be no change in overall market size, and that the organization could not build new capacity. They then ran the simulator, testing four different capacity utilization "goals": 70%, 75%, 78%, and 82% of maximum possible capacity.

When they ran the simulation, they discovered that when the plant is running at 70% or 75% of its maximum capacity, it is able to maintain a set level of production. But at 78% or 82%, an unexpected oscillation occurred—after a short period of time, the number of orders began to drop dramatically, and then gradually rebounded (see "Scenario 1" in "Capacity Utilization Scenarios").

From these results, the group hypothesized that at higher capacity utilizations the plant is less flexible in meeting customer demands, which affects the company's reputation and leads to a decline in sales. Once the number of orders falls below the utilization goal, the plant then has time to improve flexibility and quality. Over time, its reputation and sales gradually rebound until

once again it is in a situation with high capacity utilization but low flexibility—and the cycle begins again.

In the second scenario, the group tested the same four capacity utilization goals, but with new assumptions: that the market would grow steadily, and that the company would expand capacity when the forecasted demand exceeded the current capacity (see "Scenario 2"). To their surprise, when they ran this simulation, they found that a lower utilization goal actually leads to far more sales in the long run. From this they hypothesized that at a lower utilization, there is a greater unfilled demand, which leads to more optimistic forecasting and investment, more plant capacity, and a better reputation as a growing reliable supplier.

6. Convergence: Resolving the Dilemma

Testing various scenarios allows you to explore assumptions and gather data. But understanding more about behavior over time is only useful if it helps move toward resolution of the dilemma.

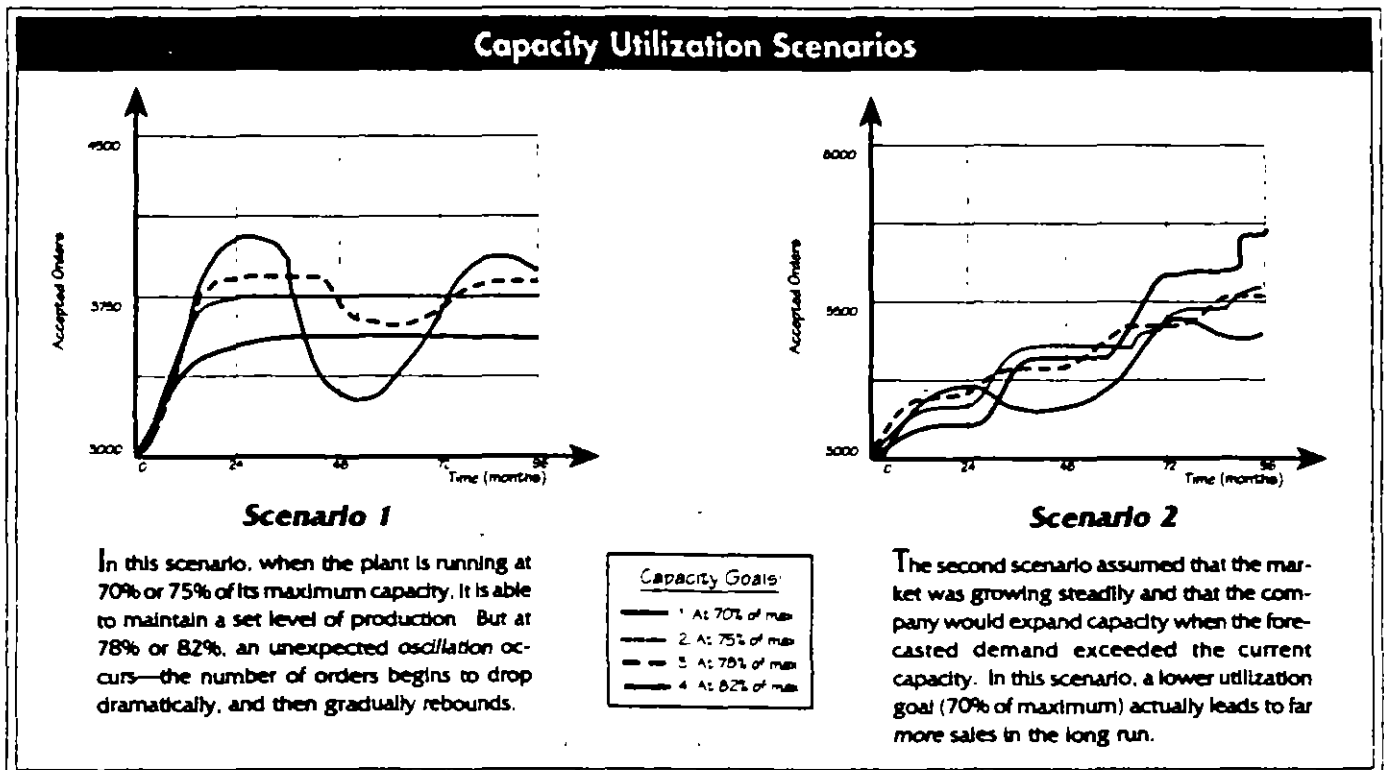
Therefore, the divergent phase should be followed by a convergent phase, in which the group closes in on the policies that produce the most desirable short-term and long-term behavior for the most likely future scenarios.

In the plant capacity example, the team discovered that there was an optimal capacity utilization level, above which the organization created undesired oscillation. The resolution to their dilemma was to set capacity utilization at a level that balanced the need to load the plant with the need to maintain flexibility and a high company reputation. In this case, the use of a simulation model enabled the team to productively address an issue that had been a long-standing dispute in the plant, and to develop a policy that was acceptable to all of the involved parties.

Don Seville is a research affiliate at the MIT Center for Organizational Learning and an associate with GKA Incorporated.

Many of these ideas emerged from conversations with Jack Homer of Homer Consulting, Inc., who also collaborated on the design of the model.

Editorial support for this article was provided by Colleen P. Lannon.



4.2

DIAGRAMAS DE

FORRESTER

EN

POWERSIM

Dinámica de Sistemas

**Manual de Referencia y Guía de Referencia
del
Programa Powersim**

Cápítulos 1 y 2

Capítulo 2 Modelando

- El propósito del lenguaje de POWERSIM es hacer una descripción, ó un modelo de un sistema real ó imaginario.
- Un modelo consiste en un conjunto de componentes interrelacionados llamados variables.
- La construcción de un modelo se hace definiendo variables y relacionarlas entre ellas.
- Powersim provee un *diagrama editor* para definir la simulación de modelos.
- Powersim permite observar la estructura del modelo en el mismo diagrama.

Modelando el Diagrama de Flujo

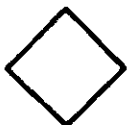
- Los bloques de un modelo más importantes del modelo son las variables (niveles, auxiliares y constantes) y los objetos para conectar variables (flujos y conectores).



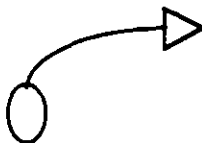
NIVEL



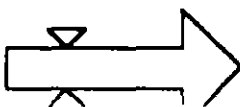
AUXILIAR



CONSTANTE



CONECTOR



FLUJO

- Una variable con un signo de interrogación quiere decir que la variable aún no ha sido definida o que tiene un error en su definición.
- Un flujo con un signo de interrogación denota que el flujo no está debidamente definido.
- Un flujo con signo de admiración denota que el flujo donde el flujo anterior había sido editado manualmente por el usuario.

Variables y Flujos

Constantes y Auxiliares

- Un modelo en Powersim conteniendo constantes y auxiliares puede ser comparado con una hoja de cálculo de Excel o de Lotus.
- Celdas con números corresponde a una constante.
- Celdas con formulas corresponde a auxiliares.

Niveles

- Un nivel es un acumulador en el cual se le extraen y se le pone flujos, de un valor previo, cada vez que se hace la simulación.

Flujos

- Representa la idea de transportación de cantidades a los niveles.
- Los flujos pueden tener una fuente o un destino que este fuera del sistema.
- La magnitud de un flujo es controlada por una variable la cual esta unida al valor del flujo.

- Cuando el cursor halla cambiado hacia abajo suelta el botón del mouse y corre la nueva simulación

Para expandir horizontalmente

- Seleccionar gráfica de tiempo del objeto
- Mover el punto del mouse hasta tocar en medio del filo del objeto
- Cuando el cursor cambie horizontal presione botón izquierdo y traslada el filo a la derecha, suelta el botón cuando doubles el ancho del objeto

Tabla de tiempo

Para crear tabla de tiempos:

- Seleccionar material de la tabla
- Mover el cursor de tabla de tiempo y dar click
- Mover objeto de tabla de tiempo debajo de la gráfica de tiempo del objeto, se hace poniendose dentro del objeto, presionando botón izquierdo del mouse, cuando aparezca el cursor moviendo el objeto hacia abajo se pone debajo de la gráfica de objeto, luego soltar botón
- Dar al objeto el mismo ancho que en la gráfica de tiempo del objeto, seleccionando la tabla de tiempo. Con shift presionado seleccione el tiempo de la gráfica y luego se escoge el mismo tamaño del formato del menú, escoge ajuste horizontal
- Abrir tabla de tiempo definida de la caja de diálogo, con doble click, en la tabla del tiempo del objeto
- Seleccionar variables y dar click en botón add, esto añade variables a los parámetros de la tabla de tiempo de la lista
- Dar OK y correr simulación.

- Presionar OK y la caja se cerrará, al cerrarse se desaparecerá el signo de interrogación de la variable deseada.

Ya que tienes todas las variables definidas estas listo para la simulación, ahora salva el documento en el archivo del disco.

Como salvar un documento

- Escoger save as en el menú principal
- En la caja de propiedades poner nombre del directorio y directorio
- Escribir autor y descripción de la simulación
- Dar OK para cerrar cajas y propiedades
- En la caja en save as escribir nombre de archivo y dar OK

Simulación del Modelo

Exhibiendo resultados:

- Escoger run para escoger simulación
Los resultados serán mostrados como un autoreporte en el diagrama
- El resultado es presentado como gráfica

Exhibir resultados mostrando objetos dinámicos Gráfica de Tiempo

Para crear gráfica de tiempo

- Maximizar diagrama
- Escoger herramientas de gráfica de tiempo
- Mover el cursor de gráfica de tiempo debajo del modelo de objetos en el diagrama
- Seleccionar variables, apretando shift y dando click
- Mover el mouse dentro de una de las de las variables y presionar botón izquierdo
- Mover el punto del mouse dentro de la figura deseada y presionar boton izquierdo, moverlo dentro de la gráfica de tiempo

Cambia el nombre de nivel 1 al deseado, presiona enter.

Para crear un Flujo.

- Dar click en el ícono de flujo.
- Mover el punto del mouse dentro del diagrama y notar que el cursor cambia a flujo.
- Mueve la flecha del flujo dentro del nivel, presiona botón izquierdo del mouse, saca la flecha del flujo fuera del nivel y suelta el botón. Una nueva variable es creada con el nombre de Rate 1, este indica que la variable auxiliar es usada como valor de flujo.

Cambia nombre de variable Rate 1 por la deseada.

Para crear Variable Adicional.

- Dar click al ícono auxiliar de la barra de herramientas.
- Mover el cursor cerca del nivel y también del valor de flujo y dar click, dar nombre a la nueva variable de auxiliar diferente a la variable anterior.
- Dar click al ícono de constante en barra de herramientas.

Ubicar constante debajo del nivel y dar nombre deseado.

Crear enlace entre variables.

- Selecciona el enlace de la barra de herramientas dando un click.

Mover el enlace dentro de la figura deseada y dar click, mueve el cursor a la otra figura deseada y suelta el botón y se crea el enlace.

Definir Variables.

Para dar valor a las variables.

- Seleccionar figura deseada, dar doble click y se abre la caja de definición de variable. En la parte de definición definirla y oprimir el botón set, esto indica el valor inicial de la variable, presionar Flog y definir incremento o decremento de la variable

Capítulo 1 Introducción

Este capítulo nos menciona como utilizar paso a paso el Powersim, sabiendo utilizar el Windows.

Instalación y correr del Powersim

Instalación. Primero se debe de instalar en la computadora, ya instalado aparecerá un encabezado del Powersim en el Windows.

Para Correrlo. Se debe de abrir el título del Powersim en el Windows dando un doble click o desde el menú principal.

Si se requiere abrir otra ventana del diagrama se puede abrir desde el menú principal.

Para crear variables y conecciones se deben de seleccionar desde la barra de herramientas, las herramientas apropiadas.

Cuando se selecciona una herramienta el cursor cambia.

Construyendo un modelo estructural creando variables y conectores

Para crear un Nivel.

- Mover el mouse al punto del ícono del nivel en la barra e herramientas y dar click con el botón izquierdo.
- Mover el cursor, dentro del diagrama de la ventana y nota que el cursor cambia a un símbolo de nivel.
- Con el cursor de nivel puesto en el centro de diagrama de nivel, presiona el botón izquierdo del mouse otra vez y una nueva variable de nivel es creada con nombre de nivel 1.

Definiendo Variables

- Cada variable de un modelo es definida por una ecuación involucrando a las variables de las que depende.
- Para definir las variables se hace con el menu de Edición o cuando se selecciona la variable con 2 clicks.

Definiendo Variables

- Esta definición de variables se hace en la caja de texto (text box) de la variable a definir.
- La definición es expresada como una expresión matemática, la cual debe involucrar a todas las variables con las que esta conectada.

Definiendo Constantes

- Las constantes son definidas al meter una literal como una expresión definiendo una variable.
- La definición puede contener una o más funciones constantes, las cuales son funciones que sus valores no cambiarán durante una simulación del programa.
- Una constante puede depender de otra variable en el estado inicial.
- Si hay conectores a una constante la cual no este definida o su definición no es una expresión constante, la constante es convertida en un auxiliar.
- Si se hacen más conectores o se mueven conectores a un auxiliar implica que la variable se transforma en una definición bien dada de una expresión constante, así la auxiliar se convertirá en constante.

Definiendo Niveles

- Un nivel difiere de un auxiliar en que su valor es incrementado cambiandolo durante la simulación.
- Un nivel siempre depende del valor que tenía previamente a la simulación.

- Se debe especificar el valor inicial . Este valor inicial será definido de acuerdo a algunas reglas como las usadas para definir auxiliares y constantes.
- El valor inicial puede depender de otras variables en el modelo. Esto puede ser logrado conectando información de otras variables al símbolo de nivel.

Trabajando con Flujos

Creando un Flujo

- Puede hacerse de 2 formas
- Escogiendo un valor de flujo o un flujo por separado.

Conectando Niveles

- El diagrama de editor permite que se conecten los niveles y flujos existentes.
- Una mano pequeña aparecerá cuando un símbolo de flujo es seleccionado.
- Para unir un flujo con un nivel se pone la mano pequeña junto al símbolo de nivel, se presiona el botón del mouse y se traza un flujo, el nivel donde se desea, una vez que se trace el flujo aparecerá en el extremo contrario una pequeña nube que indica el fin del flujo.

Documentación

- Las variables deben de ser documentadas poniendo una explicación en el "Documentation text box" del "Define Variable dialog box".

Comentarios

- Son usados para poner explicaciones en la definición de una variable.
- No afectan el comportamiento del modelo cuando este es simulado, pero sirven como referencia al lector del modelo.

Nombres

- Los nombres de las variables, índices, rangos, unidades y funciones están compuestas de una letra seguida de un cero o más letras.

Unidades de Medidas

- Un error común modelando es cuando se utilizan unidades incorrectas para definir una ecuación específica de un modelo.
- Powersim permite que el modelador defina las unidades y asigne una unidad de medida a cada variable del modelo.
- Las unidades también pueden ser asignadas a expresiones en una definición de una variable.

Asignando Unidades a una Variable

- Cada variable debe tener una unidad de medida.
- Esto se puede lograr definiendo una expresión de unidad en la "Unit of Measure Combo box" en la parte superior derecha del "Define Variable Dialog box".

Usando Unidades en Expresiones

- Para definir una unidad de una constante o de un resultado de una función, se pone la expresión de la unidad dentro de llaves.

Relación con otros Lenguajes de Modelación

Matemáticas

- Las ecuaciones de Powersim corresponden en gran manera a muchas ecuaciones matemáticas.
- Muchas expresiones en Powersim están escritas exactamente en lenguaje matemático.
- Existen algunas diferencias ya que Powersim utiliza caracteres standar en los cuales no incluye como sumatorias, integrales, etc.

DYNAMO

- Las ecuaciones vistas en Powersim son muy parecidas a las del modelo DYNAMO Profesional Plus. Aunque hay una diferencias, una de ellas es que en DYNAMO hay que escribir más estatutos.

STELLA y ithink

- En los programas STELLA/ithink y en Powersim los diagramas tienen mucho en común ya que utilizan diagramas para editar los modelos.
- También para la construcción de dinámica de sistemas y en flujo de diagramas definidos por Jay. W Forrester alrededor de 1960.
- Los modelos STELLA/ithink se pueden convertir fácilmente en Powersim.

CASO 1

B.F. Calhoun en sus experimentos con ratas Noruegas observó el efecto de la aglomeración sobre la mortalidad de sus infantes.

El confinó una población de ratas Noruegas en un terreno pequeño, con abundante comida y espacio para vivir, con la eliminación de enfermedades y depredadores, solo el comportamiento con respecto a las otras permaneció como un factor que pudiera afectar el incremento en su número. Ellas no podían escapar de las consecuencias de su comportamiento social al elevarse la densidad de población. Calhoun encontró que el amontonamiento de las madres, falló para construir nidos apropiados y abrigo adecuado para los infantes, conduciéndolos a la muerte.

La descripción verbal anterior incorpora las bases de un modelo bajo las siguientes suposiciones:

- a) El espacio confinado no permite inmigración, emigración y la presencia de depredadores.
- b) Existe amplia y suficiente comida para sobrevivir
- c) El espacio de confinamiento tiene condiciones ambientales constantes.
- d) Se descartan los efectos de la edad sobre la capacidad reproductiva.
- e) La proporción de hembras a machos es de 1.

Los resultados de los experimentos han dado información sobre lo siguiente:

- 1) Una rata adulta, normalmente (sin efectos de la aglomeración) es capaz de producir 0.4 infantes por mes (este valor es conocido como fertilidad normal)
- 2) La forma en que afecta la densidad de población PD (medido en ratas por metro cuadrado) a la fertilidad normal esta dada en la figura de abajo por ejemplo esta gráfica significa que si la densidad PD es de 0.02, la fertilidad normal (0.4) se ve afectada por un factor de 0.2, lo que significa que cada rata hembra solo generaría $0.4(0.2)$ infantes por mes.
- 3) La tasa de nacimientos depende directamente del número de ratas hembras, la fertilidad normal, y el multiplicador de sobrevivencia infantil (ISM).
- 4) Las muertes de ratas solo dependen de su vida promedio.

CASO 2

Como todos sabemos, los productos básicos presentan comportamiento cíclico muy característico. Un ejemplo clásico de esto es la carne de puerco y sus derivados. La producción, el consumo y el precio de este producto presentan oscilaciones recurrentes que pueden ser simuladas y estudiadas con un modelo de dinámica de sistemas.

El principal elemento en el modelo es el inventario de carne de puerco que mantienen los carniceros. Cuando los cerdos son sacrificados en las granjas, los productos del puerco que se envían a las ciudades incrementan el inventario de las carnicerías. Cuando la carne es vendida, el inventario decrece.

En términos generales, la cantidad de carne de puerco que consume la gente depende del precio que a su vez depende del nivel de inventario. Cuando el inventario es alto los precios decrecen y viceversa, cuando el inventario es bajo los precios suben.

Para los efectos de este modelo, suponga que el número de cerdos sacrificados mensualmente es una constante: 4 millones de cerdos por mes. Los cerdos pesan en promedio 250 lb. cada uno y se aprovecha para carne aproximadamente el 60% del peso total.

En promedio, cada persona en EUA consume 3 lb. de carne de puerco al mes. Suponga una población constante de 200 millones. Sin embargo cuando el precio es alto, la gente consume menos de 3 lb al mes y viceversa. Es decir que el consumo per-capita variará alrededor de 3 lb. / persona / mes, en función a cambios en el precio real de la carne de puerco con respecto al precio normal de ésta. Considere el precio normal de la carne de puerco como \$1/lb. El precio real depende del precio de la carne de cerdo en pie mas un " mark-up" de \$0.50 (Recuerde que el rendimiento de carne comercializable de un cerdo en pie es de 60%).

A su vez, el precio de la carne de cerdo en pie, que es de \$0.50/lb; varía en función a las existencias de carne de puerco en el inventario de los carniceros, cuya meta es mantenerlo en 2 semanas de consumo promedio. Si el inventario cae por debajo de esa cobertura, el precio del cerdo en pie tiende a subir ya que los carniceros están dispuestos a pagar mas por ellos para reponer inventarios. Por el contrario cuando las existencias rebasan ese nivel el precio tiende a bajar.

En base a esta información haga los siguiente:

- a) Desarrolle el digrama causal.
- b) Desarrolle el diagrama de flujo
- c) Formule el modelo en POWERSIM

CASO 3

El problema del transporte en Monterrey

En la ciudad de Monterrey se desea hacer un estudio sobre el cumplimiento de la demanda en el transporte urbano.

La población de la ciudad, que actualmente es de 1,300,000 habitantes presenta un tasa de natalidad equivalente al 3% anual de la población, así como una tasa de mortalidad correspondiente al 1% de la población. Cabe señalar que en el estado de Nuevo Leon y particularmente en la ciudad de Monterrey se presenta un gran movimiento de inmigrantes que, según lo revelan estudios muy confiables, representa un 2.8% de la población, en tanto que la tasa de emigrantes solo es de 0.37% de ésta.

Por otro lado se sabe que la población que requiere de servicios de transporte (tales como automóvil, taxi, camión urbano y otros) corresponde al 53% de la población total en tanto que el 42% de ésta última tiene la oportunidad de transportarse en automóvil propio. Con respecto a la población que no posee automóvil, se sabe que el 81% tiene la necesidad de un camión urbano en tanto que el 15.5% tiene preferencia por el servicio de taxi, el resto de la población que no posee autimóvil, pero que tiene necesidades de servicio de transporte utiliza algunos otros medios (Peseras, bicicletas, motocicletas, etc.)

Cabe señalar que de acuerdo a una nueva política ecológica del estado se deben plantar en la ciudad una cierta cantidad de árboles anualmente correspondiente al 0.36% de la población de la ciudad. La muerte de éstos corresponde al 2.8% anual de la población total arborea. La población actual de ár-

boles es de 356,875.

Con respecto al transporte urbano, se sabe que la incursión anual de unidades de camiones urbanos, que actualmente es de 1300, está íntimamente ligada a la siguiente relación:

No. de camiones a comprar = (Número de arboles / Factor ecológico)

donde factor ecológico = .185

La destitución de camion al mes corresponde al 15% de las unidades en circulación en tanto. Por el lado de los taxis (cuyo número es de 580), el número anual de unidades a comprar se calcula de igual manera que los camiones, sin embargo en este caso el Factor Ecologico es de 995, el porcentaje de destitución de taxis es del 10% del total de los que están en circulación. El cupo promedio de personas por cada camión es de 55 personas en tanto que la capacidad promedio del taxi es de 2 personas.

Conteste:

- a) Determine el grado de cumplimiento promedio de la demanda para los próximos 25 años de Camiones urbanos y taxis. Obtenga conclusiones.
- b) Determine el número de unidades de camiones urbanos y taxis que habrán en un plazo de 50 años suponiendo que continuaran los mismos patrones de comportamiento.

4.3

SISTEMAS DINAMICOS

EN LA EXPLORACION

Y

EXPLOTACION DE

PETROLEO

A System Dynamics Model for the Indian Oil and Gas Exploration/Exploitation Industry

SANJIB CHOWDHURY and K. C. SAHU

ABSTRACT

A system dynamics model has been developed to study the long-term dynamic behavior of the Indian oil and gas exploration/exploitation industry. The dynamics of finite and nonrenewable fossil fuel resources are portrayed in the model and are linked with the financial sector. The credibility of the model has been enhanced through model validation. Model-generated projections were compared with the projections made by the industry and the Indian government and found satisfactory in both trend and values. Sensitive parameters are identified through "what-if" tests. The effect of important variables on the model has been studied in order to choose alternative policies. The expected behavior of the model has been analyzed under a standard run and under changed conditions. Perhaps the most striking revelation of the study is that oil production will start declining after attaining a peak of 39.5 million tons during 1994-96. Similarly, gas production will start falling after reaching a peak of 25.1 billion cubic meters during 1997-99. India will face a difficult situation. Alternative policies, if adopted, will mitigate the oil crisis to some extent in the short term, but will be unable to contain growing demand and the consequent imports in the long run.

Introduction

Oil has shaped contemporary civilization to such an extent that no country, developed or developing, can ignore it in its short- and long-range planning. Hardly a day passes without oil being in the news. No other single commodity has such an influence on society. The factors that influence the oil and gas exploration/exploitation industry have been considered in the present model, along with their effects on the system. These factors encompass the following: the finiteness of hydrocarbon reserves; the dynamics of the discovery and production of hydrocarbons; the economic limits to production and exploitation; the difficulties and uncertainties involved in oil exploration; the growing demand for oil; international oil price fluctuations; exploration strategies; the possibility of oil substitution and its effects; economic obligations; unilateral decisions made by the government in fixing domestic oil prices; the sales tax, royalties, and oil cess¹ (which influence the industry's financial health); limitations faced by the government in investment; the financial implications of exploration and exploitation activities.

SANJIB CHOWDHURY is Deputy Director (Industrial Engineering) of the Oil and Natural Gas Commission, India. He received B. Tech., M. Tech., and Ph.D. degrees from the Indian Institute of Technology, Kharagpur. K. C. SAHU is a professor in the Department of Industrial Engineering and Management at the Indian Institute of Technology, Kharagpur.

Address reprint request to Dr. K. C. Sahu, Department of Industrial Engineering and Management, Indian Institute of Technology, Kharagpur 721 302, India.

The views expressed herein are those of the authors, and do not necessarily reflect those of a specific organization.

¹Oil cess is a special tax imposed by the Indian government and levied on the quantity of oil produced. It is a source of government revenue and is the product of quantity of oil produced, oil cess rate, and oil price.

Such a large, complex, nonlinear, and dynamic problem cannot be easily represented or solved by known or extended mathematical optimization models. A socioeconomic or corporate planning model with numerous interconnected variables forming loops is particularly suitable and can be used with relative ease and convenience by following system dynamics (SD) methodology without sacrificing the basic character of the problem. As a result, a computer simulation model based on SD methodology has been chosen for the present study. SD has the additional advantage of being more transparent to user-managers. It can be further refined by their interactive participation.

The Uniqueness and Uncertainties of the Oil and Gas Exploration/Exploitation Industry

In spite of scientific and technological advances, the uncertainties involved in oil exploration are great. Oil exploration is considered to be a high-technology and high-risk activity. With the present state of knowledge, oil explorers/hunters all over the world can search only for the necessary conditions for the generation, migration, and accumulation of oil and gas. There is no proven method for the direct detection of hydrocarbons.

Unlike conventional industry, wherein inputs and outputs are deterministic (that is, with a given input a quantified output is assured), the oil and gas exploration/exploitation industry has a deterministic input and a probabilistic output. The establishment of a linear relationship between input and output is not realistic. It is not possible to know with any certainty that a given input in the oil industry will produce an output (the discovery of oil and gas) within a specified period of time, even a long one. Such uncertainties inherent in the industry make it difficult to project future discoveries.

Objectives of the Model

The objectives of the model are as follows:

- To study the dynamics of the discovery and production of oil and gas reserves
- To project future exploration and exploitation activities.
- To draw a future oil-import profile and to project the probable invasion of oil substitutes into the market.
- To evaluate the impact of important variables on the model and choose alternative policies
- To develop a holistic modeling framework for the oil and gas exploration/exploitation industry covering geological, technical, economic, and environmental factors. This will facilitate the identification of exploration strategies for establishing additional recoverable hydrocarbon reserves. It can also help in planning the rates of exploitation of proven hydrocarbon reserves by taking into account the long-term prospects of the oil supply
- To study the effects of system variables, identify sensitive parameters, and understand the interactions and dynamics between different variables.

As derivatives of the objectives stated above, the likely financial implications for different activity levels can also be deduced. To restate, the objective of the model is not to test financial policies but to facilitate management in the following important ways:

- To aid decision making by allocating limited financial resources in a satisfactory manner
- To project the probable future trend of key financial variables.
- To generate the likely future financial statements, namely, the income statement, balance sheet, and fund-flow statement for different activity levels

Structure and Overview of the Model

The model consists of two sectors, namely, technical and financial. The dynamic behavior of finite and nonrenewable fossil fuel resources is portrayed in the technical sector. This sector comprises exploration and exploitation of oil and gas reserves. The interaction and dynamics of reserves accretion, production of oil and gas, exploration activities, oil imports, oil substitutes, and many other variables have been studied. The technical sector is the heart of the model, and its activities and variables are linked with the financial sector, which consists of the income statement, balance sheet, and fund-flow statement. The future trend of key financial variables has been studied, and likely financial statements for different activity levels have been generated.

THE TECHNICAL SECTOR

Figure 1 depicts the influence diagram of the exploration and exploitation of oil and gas reserves. A fraction of the revenue is invested in exploration and is disbursed for surveys and exploratory drilling activities. The higher the revenue, the greater the investment in exploration. A rise in the exploration investment means an increase in exploratory drilling. Increased exploratory drilling results in larger discoveries of oil and gas. With increasing discoveries, the level of the remaining discoverable resources is reduced. The depletion of remaining discoverable resources leads to a drop in the discovery of reserves. The discovery of reserves increases the level of remaining recoverable oil reserves. The discovery of oil and gas, with a delay, leads to production. As the discovery rate goes up, the rate of oil and gas production also shoots up. The remaining oil reserves and the rate of production together determine the life of reserves or the reserves production ratio (RPR). The greater the remaining reserves, the higher the RPR. As production grows, the RPR declines. The rate of oil production is also influenced by demand. If demand is higher than production capacity, the oil industry will produce at full capacity. If demand is less than production capacity, production will be limited to the demand level. Oil imports rise with increased demand. An increase in production reduces the import level.

As oil prices go up, more and more oil substitutes penetrate the market. A high RPR, on the other hand, discourages substitution.

THE FINANCIAL SECTOR

Figure 2 portrays the influence diagram of the income statement. Revenue increases with the rise in the quantity of oil and gas produced and their prices. Cost also soars with the increase in royalties, oil cess, sales tax, amortization, depletion, depreciation, and operational cost. The capacity to amortize rises with the increase in revenue.

A rise in depletion and depreciation reduces the level of net producing property and net fixed assets, respectively. The higher the net producing property and net fixed assets, the higher the capacity to deplete and depreciate, respectively. Pretax income and interest swells with the growth of revenue, but decreases with the increase in total cost. Interest payable increases with the rise in outside loans (interest due), but decreases with the growth in accumulated investment in public savings (interest earned). As the interest payable rises, income after interest declines. The corporate tax increases with the rise in income. Net income swells with the growth in income after taxes, but is reduced with the rise in the corporate tax. The higher the government capital and profitability ratio, the higher the dividend paid by the company. The more the dividend paid, the less the retained earnings.

Figure 3 shows the influence diagram of the balance sheet. The larger the accumulated reserves, government capital, long-term liability for gratuities and outside funds, the

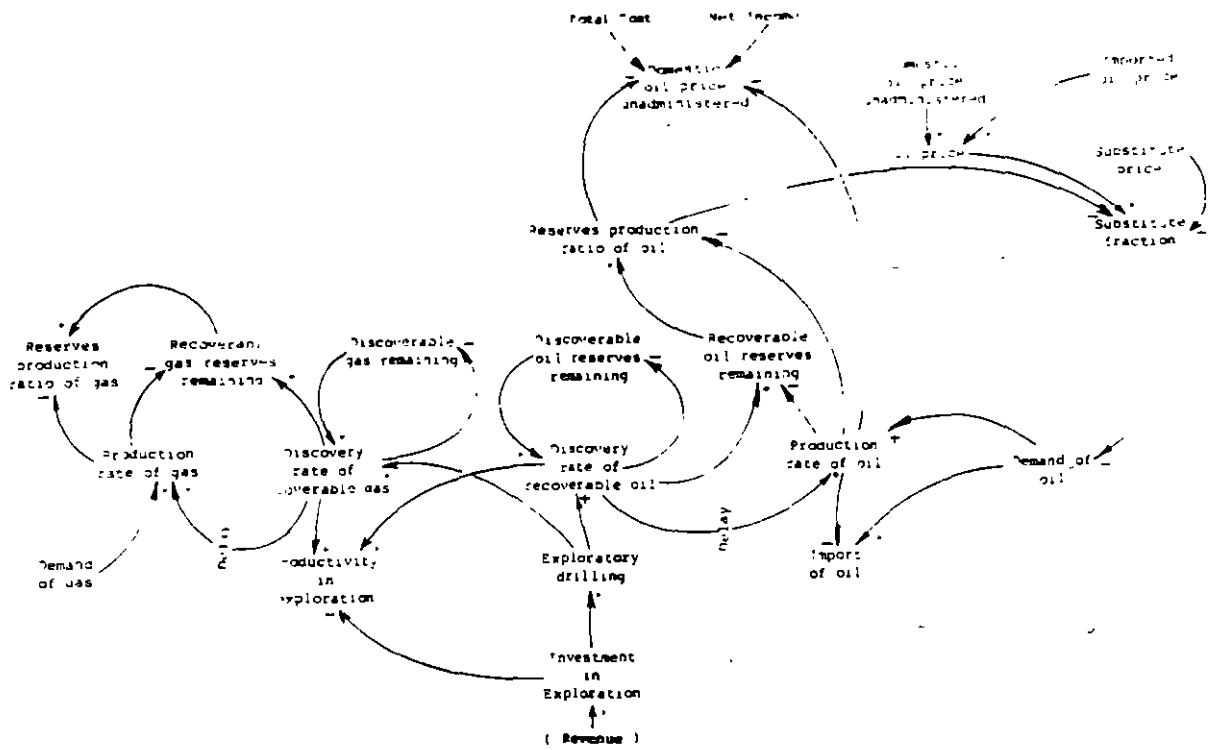


Fig. 1. Influence diagram of the exploration and exploitation of oil and gas reserves.

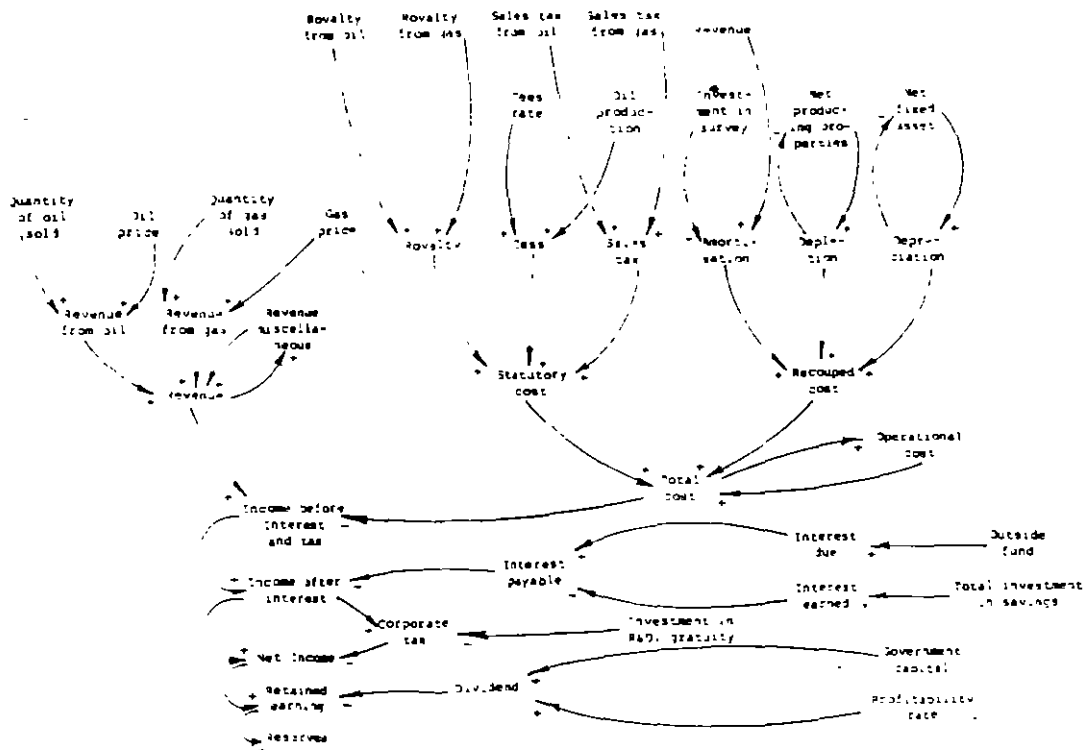


Fig. 2. Influence diagram of the income statement.

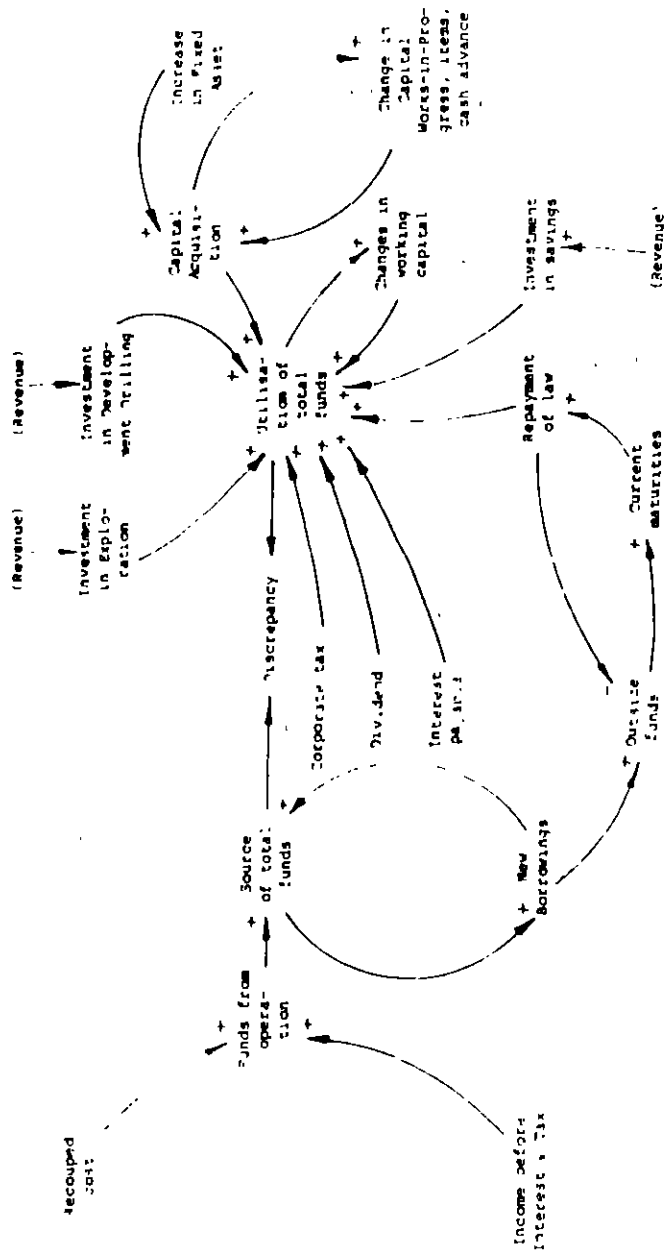


Fig. 3. Influence diagram of the balance sheet.

larger the total resources. Outside funds swell with an increase in new borrowings, but contract with an increase in repayment of loans. Current maturities and loan repayments rise with the growth of outside funds. The higher the net producing property, net fixed assets, and change in cash advance and work in progress, the higher the block capital. The greater the block capital and working capital, the greater the capital employed. The larger the capital employed and accumulated net investment in public savings, the larger the total resources.

The influence diagram of the fund-flow statement is shown in Figure 4. The total funds increase with the increase in income before interest and taxes, new borrowings, and recouped costs (amortization, depletion, and depreciation). The utilization of total funds increases with the rise in investment in exploration, investment in developmental drilling, capital acquisition, changes in working capital, investment in public savings, loan repayment, interest payable, dividends, and corporate taxes.

Model Calibration and Execution

The detailed model formulation is described in Chowdhury [1]. The model has been calibrated with input data wherein the historical trend has been allowed to continue for certain variables. The data for some variables are taken from established facts, well-known studies, and likely values as envisaged by experts. The model is simulated over a period of 40 years from 1965 to 2005 (using DYMO-SIM, a Fortran-based software package [2]). The output of the model is deduced from the input data of variables which are interconnected and influence one another.

Model Validation

Divergent views persist regarding the techniques of model validation. Classical system dynamics practitioners assert that model validation should preferably be subjective, comparing the intuitions, opinions, and judgments of experts with the model-generated policies. This would make the model realistic, pragmatic, and goal oriented in terms of its utility [3].

Quantitative tests in model validation may satisfy all the criteria, but it is not enough to rely on these as they may not represent reality. Forrester [4, 5] rejects traditional validation techniques such as statistical tests and sensitivity analyses because they overemphasize quantitative validation rather than the usefulness of the model. He proposes 13 subjective criteria for model validation, with the central theme being that a model cannot have absolute validity and that tests are necessary but not sufficient. Passing a test does not mean a model's acceptance, but failing a test advocates its rejection. Nevertheless, rejection of a model for its failure to represent a system may not be appropriate, because the model may turn out to be successful in the long term, even though it fails the test for a short period. According to Forrester [6], quantitative validation of a model should be carried out when possible and anticipated results are expected to justify the cost and efforts.

The model-generated values for certain important variables have been compared with the actual values for a period of ten years and were found to be similar in trend. No significant variation is observed between the simulated and actual values for key variables.

In order to enhance the fidelity and acceptability of a model, some authors prefer statistical tests as the model validation criteria because of their elegance, dynamic behavior, and mathematical approach. The *t* test and percentage root mean square error have been computed for quantitative validation of the model. The estimated *t* statistics were found to be acceptable against the values obtained from the statistical table at the

95% confidence level. The percentage root mean square error for some key variables is below 10%, and the rest of the key variables are within the range of 10%-20%.

Considering the uncertainties involved and the probabilistic nature of the oil and gas exploration/exploitation industry, where input is deterministic but output (discovery and production) is probabilistic, it is observed that variations less than 20% in planning projections for the industry are generally accepted

The Sensitivity Test

Historical data for some parameters of the SD model may not be available or adequate. In such cases, the initial values of these parameters are judged intuitively on the basis of certain assumptions and in consultation with knowledgeable persons in the industry. The initial input data thus obtained may produce erroneous results. The sensitivity test tries to obtain a set of reasonably appropriate initial values (through trial and error), so that the error in the result is minimized. A good SD model should not be too sensitive to environmental factors.

Most of the parameters in the present model were found to be insensitive. The parameters that significantly affect the model have been identified and are discussed below.

CHANGES IN THE INITIAL VALUE OF THE DISCOVERY RATE OF OIL.

If the initial value of the discovery rate of oil is increased by 10% and 20%, no significant change in the peak production value is observed. The peak production value of gas is found to decline slightly. However, the peak production values of oil and gas are advanced by two and three years for a change of 10% and 20%, respectively, in each case. Total oil production during the simulation period rises slightly, but that of gas remains nearly the same. The occurrence of peaks for related activities are also advanced. The production capacity and other associated activities decline at a faster rate in the postpeak period. As a result, oil imports rise during this period (Table 1).

THE EFFECT OF CHANGES IN ULTIMATE DISCOVERABLE OIL AND GAS RESERVES

If the ultimate discoverable oil and gas reserves increase to the projected optimistic values of 2265 million tons and 1485 billion cubic meters, respectively [1], then oil production peaks at a higher value of 46.7 million tons during 1996-98. This is an increase of 18% over the peak of the standard run, and the peak is postponed by two years. Similarly, gas production peaks at a slightly higher value of 26.8 billion cubic meters, and its occurrence is delayed by one year (Figure 5). The cumulative discovery of oil and gas up to 2005 increases by 19% and 7.6%, respectively, over the standard run. For the same period, the cumulative production of oil and gas increases by 14% and 5.2%, respectively (Figure 5).

All of the related activities and variables, such as exploratory drilling and developmental drilling, increase during the simulation period (Figure 6).

The increased production rate of oil during the simulation period takes care of the growing demand and eases oil imports to some extent. But because demand is growing at a faster rate than production, increased production is unable to contain both the demand and the necessity of oil imports. It is unlikely that any dramatic change would take place to mitigate such a crisis.

If the ultimate discoverable oil and gas reserves are limited to 1750 million tons and 1385 billion cubic meters, respectively [1], the effect of the model on the key variables is not significantly different over the standard run. Oil production, gas production, exploratory drilling, and developmental drilling peak at marginally lower values and the occurrence of peaks advances slightly.

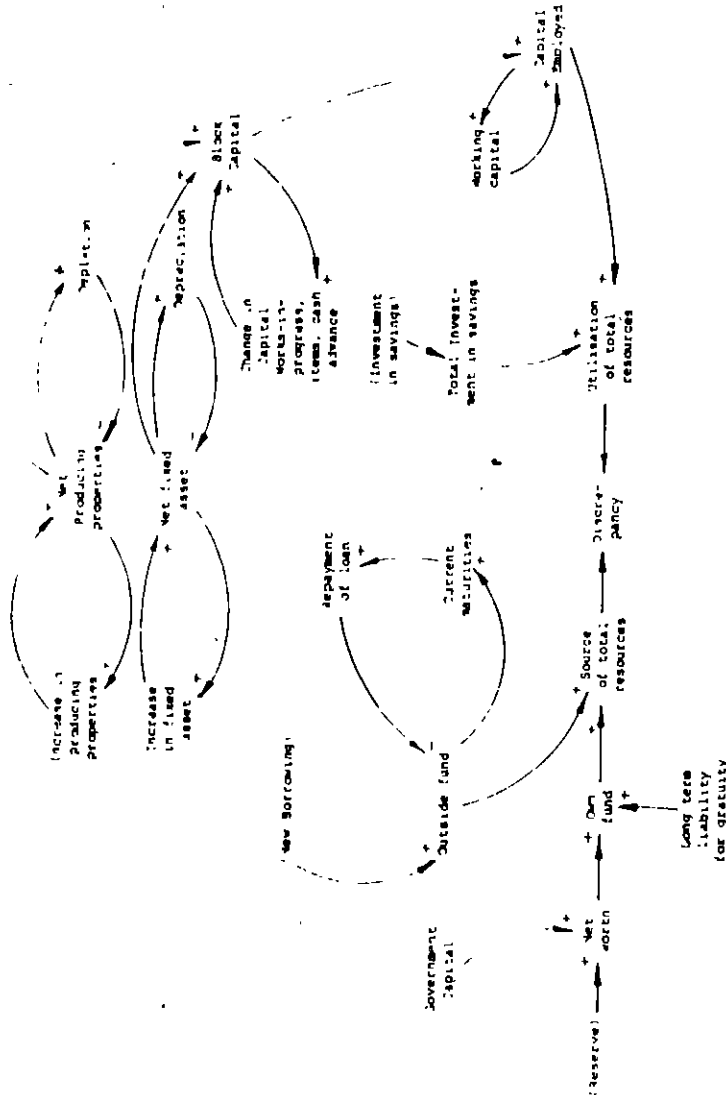


Fig. 4. Influence diagram of the fund-flow statement.

TABLE I
Effect of Changes in the Initial Value of the Discovery Rate of Oil (DOI) (millions of tons)

Variable	(DOI) = 1.18 (standard run)	(DOI) = 1.3 (10% increase)	(DOI) = 1.4 (20% increase)
Total discovery up to 2005			
Oil reserves (million tons)	1200	1200	1200
Gas reserves (billion cubic meters)	893	899	904
Oil production (million tons)			
Total production up to 2005	809	830	846
Peak production	39.5	39.0	38.8
(year of peaking)	(1994-96)	(1992-94)	(1991-92)
Terminal year production	25.56	22.37	20.26
Gas production (billion cubic meters)			
Total production up to 2005	419	422	422
Peak production	25.1	23.8	22.8
(year of peaking)	(1997-99)	(1995-98)	(1994-97)
Terminal year production	20.74	18.53	16.9
Exploratory drilling (million meters)			
Peak value	0.61	0.59	0.566
(year of peaking)	(1995-96)	(1994-95)	(1993-95)
Terminal year value	0.41	0.36	0.33
Developmental drilling (million meters)			
Peak value	0.656	0.625	0.60
(year of peaking)	(1995-97)	(1995-96)	(1993-95)
Terminal year value	0.493	0.434	0.395
Oil imports in terminal year (million tons)	69.4	73.7	76.6

Model Behavior on the Standard Run

The validated standard run is obtained after several trial runs, during which the initial values, constants, and delay and table function values were adjusted and modified until the model generated the desirable behavior.

The discovery rates of oil (Figure 7) and gas (Figure 8) showed a fluctuating upward trend to 1982 and 1977, respectively. Thereafter, they showed a fluctuating declining trend for the rest of the simulation period. The oil and gas discovery rates peak at 97.3 million tons in 1982 and 66.9 billion cubic meters in 1977, respectively.

The production rates of both oil and gas rise slowly but steadily and touch a peak plateau of 39.5 million tons during 1994-96 for oil and 25.1 billion cubic meters during 1997-99 for gas. These rates for oil and gas start declining after 1996 and 1999, respectively, and are at the level of 25.5 million tons and 20.73 billion cubic meters at the end of the simulation period. The discovery and production rate curves both have the familiar bell shape and are analogous to the biological growth and decay curve. The cumulative discovery and production of oil (Figure 7) and gas (Figure 8) follow the logistic curve pattern.

Activities such as exploratory drilling and developmental drilling follow the growth and decay trend much like the production curve.

The demand for oil increases throughout the simulation period, and oil production rises until 1995. Therefore, oil imports (Figure 9) remain at nearly the same level until 1987 and increase marginally until 1994. Because oil production rates start declining after reaching a peak plateau, oil imports grow monotonically for the rest of the simulation period.

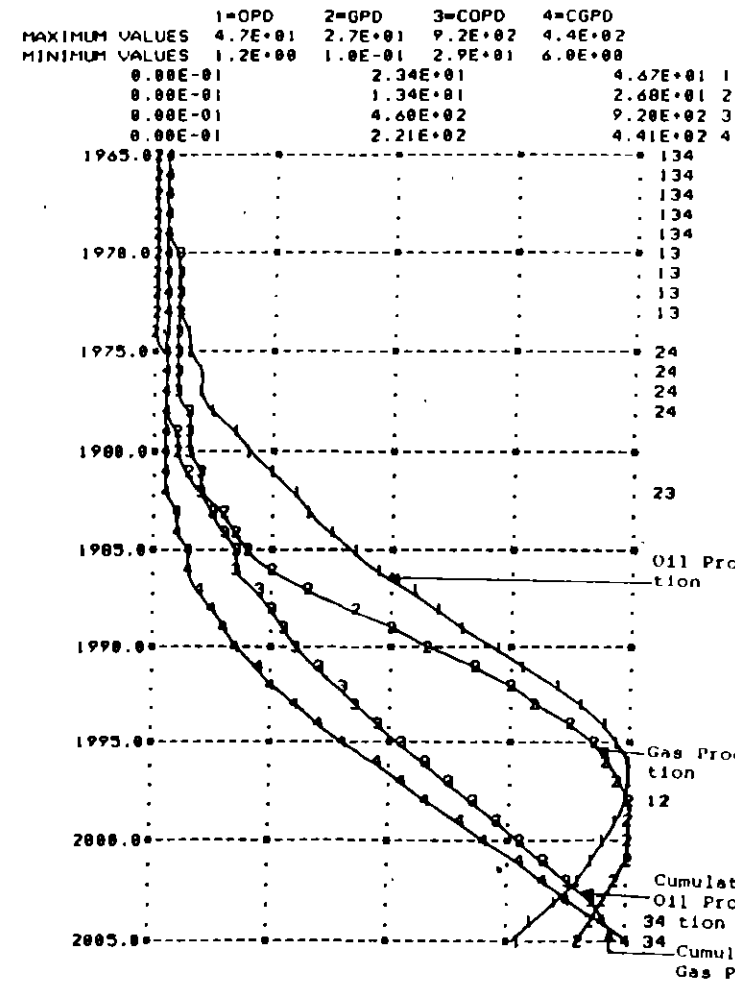


Fig. 5. Effect of changes in ultimate discoverable oil and gas (UDG = 2265, UDG = 1485) on the rate of oil and gas production and cumulative production of oil and gas.

The share of oil substitutes (Figure 9) is found to be negligible up to 1987 (nil until 1981, less than 1% up to 1984, less than 2% up to 1987) and increases from 1988 onward. It captures 21% of the oil market in 2005.

The financial variables and ratios—revenue, total cost, income before interest and taxes, net income, operational funds, new borrowings, total funds, corporate taxes, capital acquisition, investment in public savings, investment in exploration, investment in de-

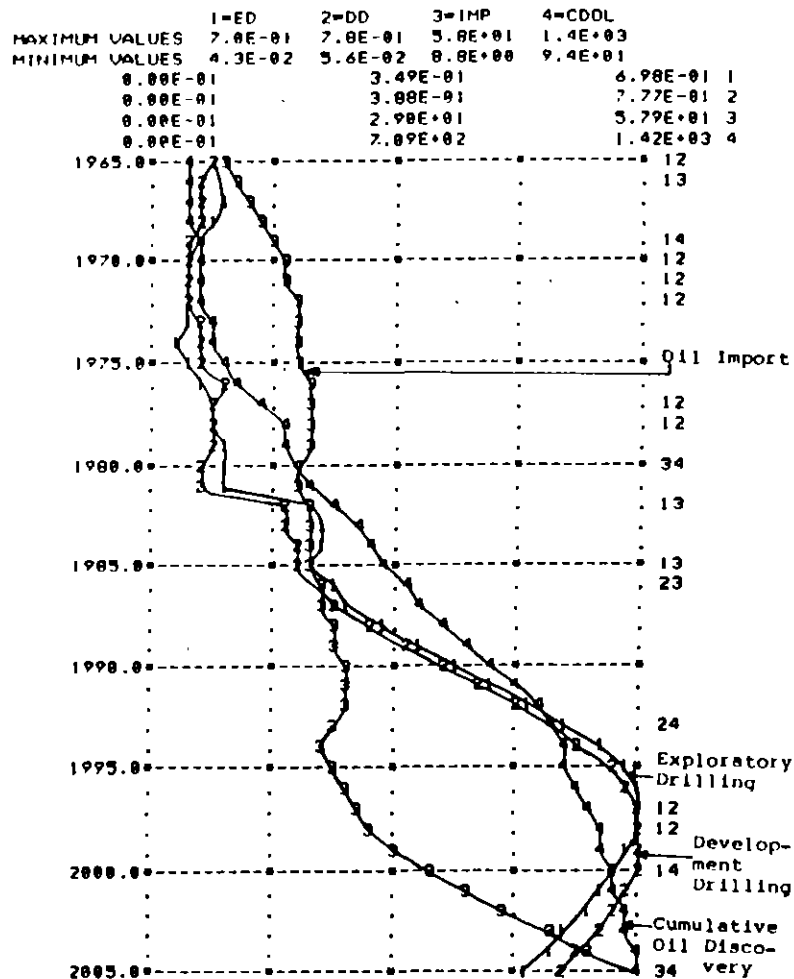


Fig. 6. Effect of changes in ultimate discoverable oil and gas (UDO = 2265, UDG = 1485) on exploratory drilling, developmental drilling, oil imports, and cumulative discovery of oil.

developmental drilling, revenue from oil, revenue from gas, miscellaneous revenue, re-couped cost, operational cost, amortization, net income to net worth ratio, interest coverage, debt equity ratio, debt capital ratio, net fixed assets turnover ratio, working capital turnover ratio, return on investment, income before interest but after taxes to capital employed ratio, income before interest and taxes to capital employed ratio—follow the bell pattern (similar to the production curve) with different gradients and peaks. Other financial variables—namely, net worth, outside funds, total resources, block capital, working capital, capital employed, total investment in public savings, interest, loan repayment,

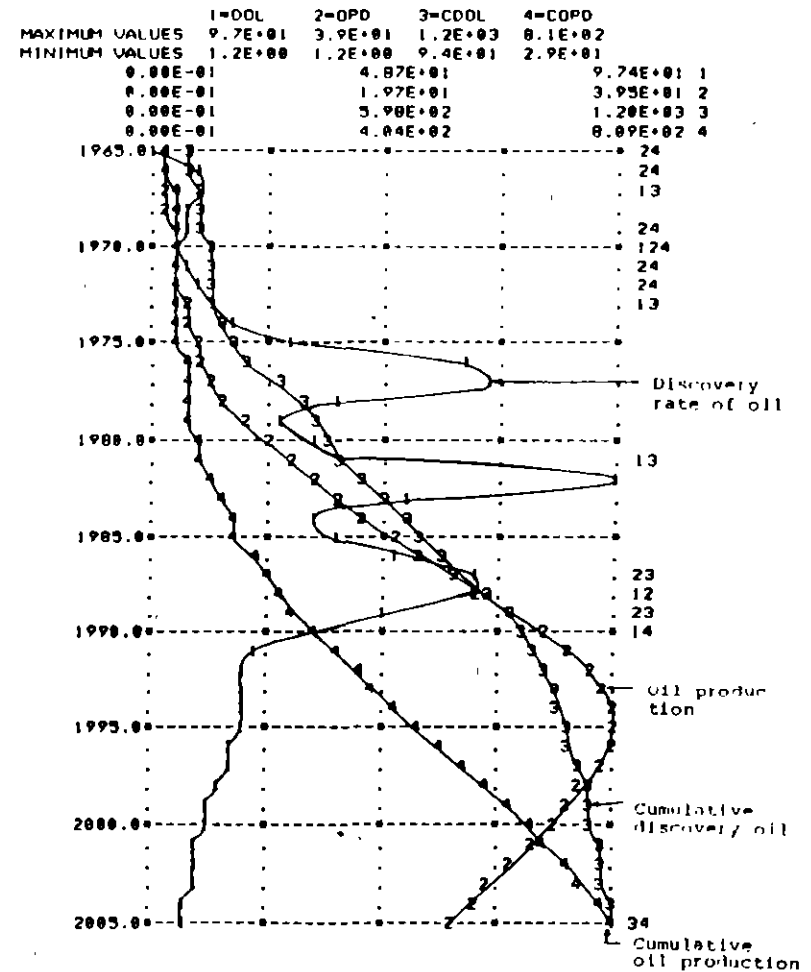


Fig. 7. Rate of discovery and production of oil and cumulative discovery and production of oil per year.

net fixed assets, net producing property, capital works in progress, cash advance and materials in transit, depletion, depreciation, reserves government capital ratio—do not change appreciably for some years in the initial stage. Thereafter, they grow monotonically till the end of the simulation period.

Comparisons of Future Projections

To further enhance the acceptability of the present model and the reliability of future projections, the model-generated values were compared with the projections made by the oil and gas exploration/exploitation industry for the period 1988–2005.

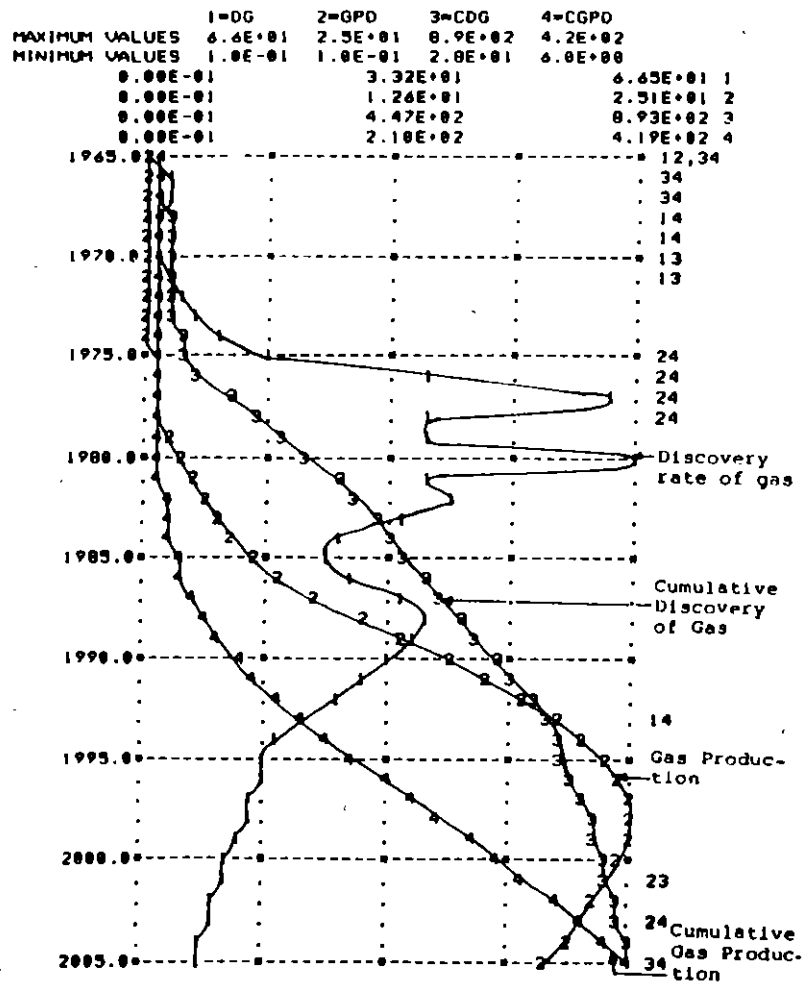


Fig. 8. Rate of discovery and production of gas and cumulative discovery and production of gas per year.

The corporate plan of the oil and gas exploration/exploitation industry, submitted to the Indian government, mainly contains the projections of oil production, gas production, exploratory drilling, developmental drilling, and reserves accretion up to 2005. These have been taken as reference-level projections for the purpose of comparison.

In Figures 10-13, the model-generated projections and those made by the oil and gas exploration/exploitation industry are plotted for oil production, gas production, exploratory drilling, and developmental drilling, respectively, for the period 1988-95 and are found to be compatible.

The variations between the simulated projections and those made by the industry

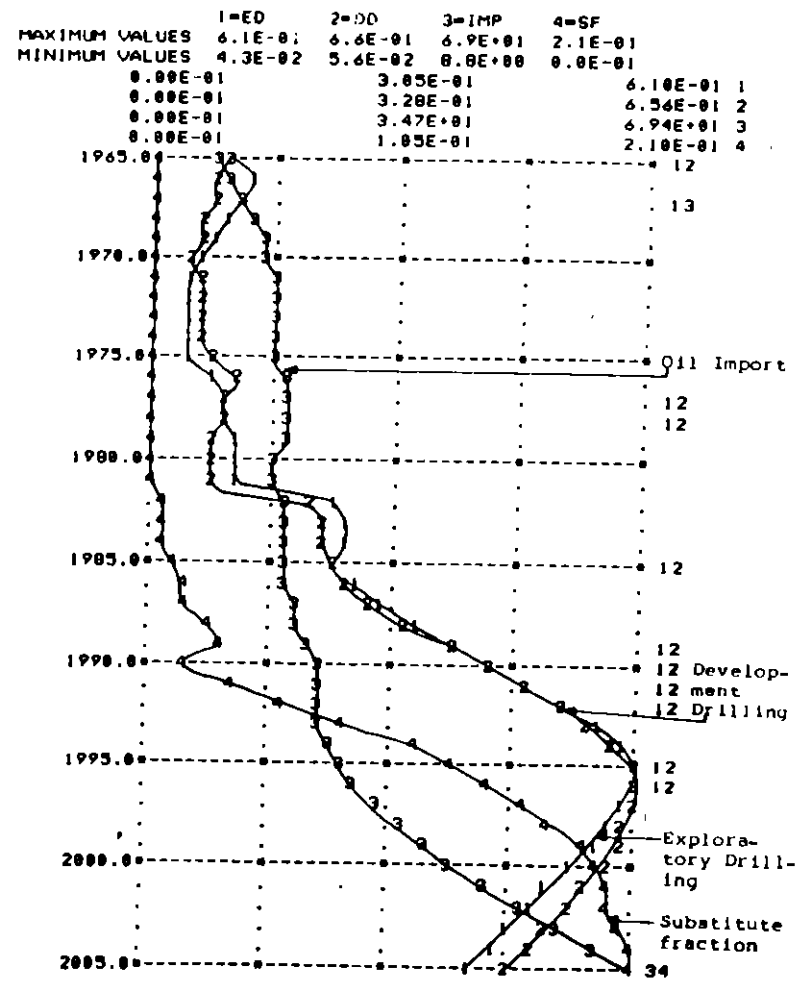


Fig. 9. Exploratory drilling, developmental drilling, oil imports, and substitute fraction per year.

regarding oil production, gas production, exploratory drilling, developmental drilling, and reserves accretion range between 1% and 7% for the period 1990-95 and between 1% and 13% for the period 1995-2000 (except exploratory drilling, whose variation is found to be 27%). But substantial variations are observed for these variables during the period 2000-2005 (Table 2).

It may be concluded from the above that the variations up to the year 2000 are not significant and that model-generated projections conform with the projections made by the oil and gas exploration/exploitation industry.

However, the variations beyond 2000 arise from a basic difference in assumptions. The projections made by the oil and gas exploration/exploitation industry anticipate a

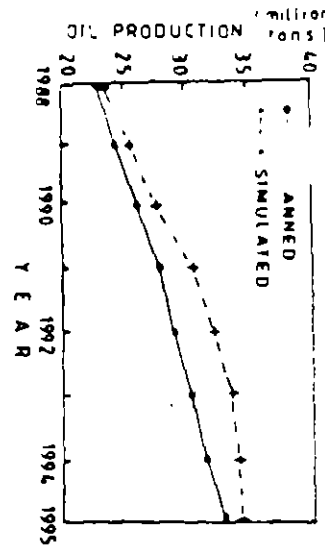


Fig. 10. Oil production per year.

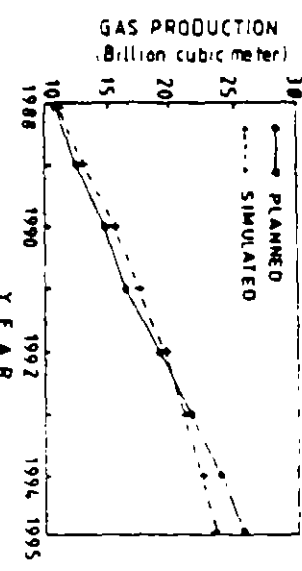


Fig. 11. Gas production per year.

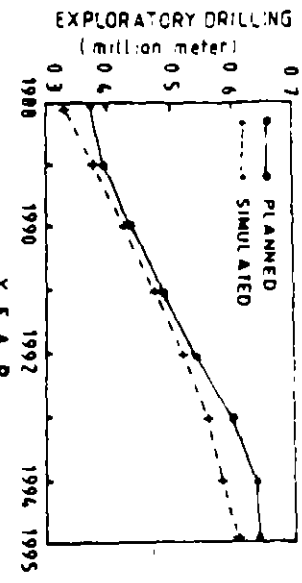


Fig. 12. Exploratory drilling per year.

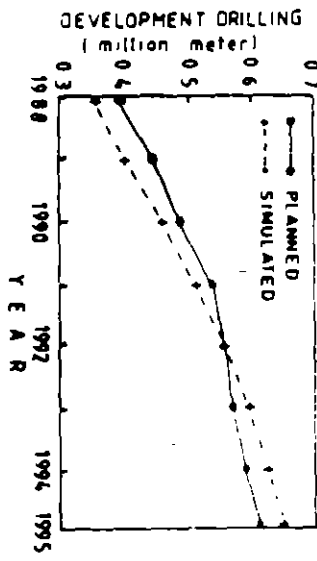


Fig. 13. Developmental drilling per year.

TABLE 2
Comparisons of Future Projections: Model Generated and by Industry

Variable	1990-95			1995-2000			2000-2005		
	Model-generated projection	Industry projection	Variation (%)	Model-generated projection	Industry projection	Variation (%)	Model-generated projection	Industry projection	Variation (%)
Oil production (million tons)	191.4	177.2	7.4	183.9	186.2	1.2	145.0	212.6	46.6
Gas production (billion cubic meters)	105.8	107.2	1.3	124.4	129.4	4.0	112.4	162.1	44.2
Exploratory drilling (million meters)	2.76	2.859	3.5	2.865	2.087	27.1	2.304	1.951	15.3
Developmental drilling (million meters)	2.94	2.859	2.7	3.206	3.497	9.0	2.728	4.24	56.5
Oil and gas reserves accretion (million tons)	200.2	189.5	5.3	137.7	156.5	13.6	87.1	140.2	60.9

TABLE 3
Effect of Changes in Government Subsidies for Exploration (GSE)

Variable	GSE = 0				
	(standard run)	GSE = 1	GSE = .15	GSE = .2	GSE = 25
Total discovery up to 2005					
Oil reserves (million tons)	1200	1210	1250	1270	1290
Gas reserves (billion cubic meters)	893	928	947	969	992
Oil production (million tons)					
Total production up to 2005	809	853	883	916	951
Peak production	19.5	39.8	40.6	41.7	42.9
(year of peaking)	(1994-96)	(1991-92)	(1990-91)	(1989-90)	(1988-89)
Terminal year production	25.56	21.1	19.6	18.47	17.47
Gas production (billion cubic meters)					
Total production up to 2005	419	430	436	448	466
Peak production	25.1	21.5	23.0	22.9	21.4
(year of peaking)	(1997-99)	(1995-97)	(1994-96)	(1993-95)	(1992-95)
Terminal year production	20.74	17.78	16.67	15.87	15.25
Exploratory drilling (million meters)					
Peak value	0.61	0.647	0.671	0.713	0.762
(year of peaking)	(1995-96)	(1994-95)	(1993-95)	(1992-94)	(1991-93)
Terminal year value	0.41	0.381	0.376	0.378	0.384
Developmental drilling (million meters)					
Peak value	0.656	0.617	0.606	0.603	0.60
(year of peaking)	(1995-97)	(1994-96)	(1993-95)	(1992-94)	(1991-93)
Terminal year value	0.493	0.412	0.384	0.363	0.345
Oil imports in terminal year (million tons)	69.4	75.8	77.93	80.0	82.0

monotonic growth until the end of the simulation period, which is unlikely. The finiteness of hydrocarbon reserves will compel the exploration and exploitation activities to decrease after they reach their respective peaks, as observed in the case of the model-generated variables.

Policy Testing

According to Mohapatra and Sharma [7], the current policy design in SD is unstructured. An acceptable policy is obtained through trial and error and after spending a lot of time and effort. Talvage [8] observes that SD model formulation and simulation are easier than obtaining a reasonably stable and acceptable behavior of the model.

In order to choose alternative policies, the effect/impact of two important model variables has been studied/evaluated: (a) the effect of government subsidies for exploration and (b) the effect of stepping up production by reducing lag time.

THE EFFECT OF GOVERNMENT SUBSIDIES FOR EXPLORATION

The influence of government subsidies for exploration is tested with different values. There is no appreciable change in the peak production of oil and gas. But the occurrence of the peak is advanced by three, four, five, and six years in the case of oil and two, three, four, and five years in case of gas for subsidy levels of 10%, 15%, 20%, and 25%, respectively.

The rise in the peak for exploratory drilling and developmental drilling is marginal. For a subsidy of 20%, it is observed that discovery and production values up to 2005 rise by 6% and 13%, respectively, for oil and 8.5% and 7%, respectively, for gas (Table 3).

TABLE 4
Effect of Stepping Up Production by Reducing Lag Time

Variable	P.L.O = 12.8 years P.L.G = 12.8 years (standard run)		P.L.O = 11.2 years (10% reduction) P.L.G = 11.2 years (6% reduction)	
	Total discovery up to 2005			
Oil reserves (million tons)	1200		1200	
Gas reserves (billion cubic meters)	893		895	
Oil production (million tons)				
Total production up to 2005	809		810	
Peak production	19.5		41.0	
(year of peaking)	(1994-96)		(1991-95)	
Terminal year production	25.56		22.98	
Gas production (billion cubic meters)				
Total production up to 2005	419		435	
Peak production	25.1		26.4	
(year of peaking)	(1997-99)		(1996-98)	
Terminal year production	20.74		19.57	
Exploratory drilling (million meters)				
Peak value	0.61		634	
(year of peaking)	(1995-96)		(1995)	
Terminal year value	0.41		0.375	
Developmental drilling (million meters)				
Peak value	0.656		0.674	
(year of peaking)	(1995-97)		(1991-96)	
Terminal year value	0.493		0.450	
Oil imports in terminal year (million tons)	69.4		73.2	

P.L.O, lag time for production level to match the discovery level for oil; P.L.G, lag time for production level to match the discovery level for gas.

Production capacities of oil and gas and the related activities continue to rise at a faster rate than those of the standard run until they cross the peak-level values of the standard run and attain their respective peaks.

As a consequence, oil demand and imports would get temporary relief until that period. Thereafter, production capacity and other related activities fall below the level of corresponding value in the standard run, thus creating a difficult situation in the long term. Government subsidies for exploration result in a slightly earlier discovery of reserves and an earlier attainment of peak production. Though subsidies offer marginal relief in the short term, they are unable to match the growing demand and hold down oil imports for the long term because of the finiteness of reserves.

THE EFFECT OF STEPPING UP PRODUCTION BY REDUCING LAG TIME

In order to ease the growing demand and import of oil, it is often found necessary to step up production capacity by introducing an early production system, which means the reduction of lag time for production levels to match the discovery level.

In such cases, the production of oil and gas peak at a slightly higher value and the peak occurrence is advanced by one year in each case.

The discovery and production of oil and gas up to 2005 remain nearly at the same level as that in the standard run. Exploratory drilling and developmental drilling peak at slightly higher values. The growing demand and need for imports are eased to a small extent until peak production is attained (Table 4).

Production capacity and related activities in the alternative policy run fall at a faster

rate in comparison to the standard run after attaining their own peaks and crossing the peak levels of the standard run. As a result, oil demand and import position, are aggravated in the postpeak period.

Conclusions

An SD model covering geological, economic, and environmental factors has been developed to study the long-term dynamic behavior of the Indian oil and gas exploration/exploitation industry.

The model consists of technical and financial sectors. The technical sector comprises the exploration and exploitation of oil and gas reserves, in which the interaction and dynamics of reserves accretion, production of oil and gas, exploration activities, oil imports, oil substitutes, and many other variables have been studied.

The financial sector comprises the income statement, balance sheet, and fund-flow statement. It studies the future trend of key financial variables and generates likely financial statements for different activity levels.

The fidelity and acceptability of the model have been enhanced through model validation. The *t* test and percentage root mean square error with respect to important variables were found to be satisfactory. The comparisons between model-generated values and projections made by the industry and the Indian government were also found to be satisfactory both in trend and values. The "what-if" test was used to identify sensitive parameters. Most of the variables were found to be insensitive to the environment. The impact of important model variables has been evaluated in order to choose alternative policies. The likely behavior of the model under a standard run and under changed conditions was analyzed.

The following points may be concluded from the results of the standard run and those of alternative runs under changed conditions:

1. The discovery rates of oil and gas reserves will show a fluctuating upward trend until the reserves reach their respective peaks; thereafter they will experience an undulating declining trend.
2. The production rates of oil and gas will have a slow and smooth ascent, a plateau at 39.5 million tons during 1994-96 for oil and 25.1 billion cubic meters during 1997-99 for gas, and a smooth descent.
3. Activities like exploratory drilling, developmental drilling, and different financial parameters will follow a trend similar to that of nonrenewable finite natural resources, with an initial upward movement (growth), a plateau or stability at the peak, and decline (decay).
4. As long as oil production continues to grow, oil imports will remain within a manageable range. As production capacity declines after crossing the peak (in 1996), oil imports will continue to rise unabated until the end of the simulation period. The share of oil substitutes will rise, and they will capture nearly one-fifth of the oil market in 2005.
5. An increase in government subsidies for exploration and a stepping up of production by reducing lag time will provide marginal relief for the short term, but will be unable to contain growing demand and imports in the long run.

However, the SD model cannot fully represent the complex reality. It must be supplemented by the intuition, judgment, and experience of managers and planners. The model should be viewed in the proper perspective.

References

1. Chowdhury, S., *Corporate Planning for the Oil and Gas Exploration/Exploitation Industry: Some Quantitative Approaches*. Ph.D. dissertation, Indian Institute of Technology, Kharagpur, 1990.
2. Bora, M. C., and Mohapatra, P. K. J., *DYNAMOSIM Users Manual*, Indian Institute of Technology, Kharagpur, 1985.
3. Sommer, M., The Econometric Challenge to SD and Vice-Versa: Some Future Perspectives, *Technological Forecasting and Social Change* 25, 263-380 (1984).
4. Forrester, J. W., and Senge, P. M., Tests for Building Confidence in System Dynamics Models, *TIMS Studies in Management Science* 34, 61-73 (1980).
5. Forrester, J. W., Confidence in Models of Social Behavior with Emphasis on System Dynamics Models, Working paper No. D-1969, MIT, Cambridge, MA, 1973.
6. Forrester, J. W., *Industrial Dynamics*, MIT Press, Cambridge, MA, 1961.
7. Mohapatra, P. K. J., and Sharma, S. K., Synthetic Design of Policy Decisions in System Dynamics Models—A Model Control Theoretical Approach, *System Dynamics Review* 1, (1985).
8. Talavage, J. J., Model Analysis to Air System Dynamics Simulation, in *System Dynamics*, A. A. Legasto, et al., eds., North-Holland, New York, 1980.

Received 7 September 1990, revised 31 October 1991

4.4

ELEMENTOS

DEL

LENGUAJE

GPSS/H

WHAT IS GPSS/H?

GPSS/H is a statement-oriented computer language and execution environment.

GPSS/H is used for modeling systems in which “units of traffic”...

- People
- Vehicles
- Parts
- Packages
- Messages
- Tasks
- Machines

...compete for scarce resources.

- People
- Physical Space
- Machines
- Vehicles
- Channels
- Work Orders
- Parts

WHY GPSS/H?

GPSS/H provides a natural framework.

- **Language elements closely parallel their real world equivalents**
- **A well-written GPSS/H model should be readable by a non-simulationist**

GPSS/H is a detailed modeling tool.

- **Models aren't (usually) built in a day**
- **Complex logic of some systems can make simulation models difficult to build or modify by non-simulationists**
- **The above results from the choice of a detailed modeling tool—not from the choice of GPSS/H itself**

STRUCTURE OF GPSS/H: COMPUTER ENVIRONMENT

GPSS/H runs on the IBM PC.

For fastest speed and largest models, GPSS/H Professional software (HPro) is used.

HPro runs on 386 or higher machines with math coprocessor and any operating environment that can run DOS applications. (MS-DOS, DR-DOS, Windows 3.x, OS/2 2.x)

**Other environments are also supported:
VAX/VMS, Unix workstations, and IBM mainframes.**

STRUCTURE OF GPSS/H: RUNNING A MODEL

“Batch” mode — output appears on the screen and/or in one or more output files.

Interactive model execution mode — you control the execution of the model.

Interactive mode (“the debugger”) is useful for model validation and debugging.

GPSS/H is a “compile-and-go” system — when you type

```
GPSSH filename
```

the model, *filename.gps*, is processed and execution - or the interactive session, begins all at once.

GPSS/H compiles so quickly that compile time is not an issue; however, Run-time GPSS/H gives you the option of sending pre-compiled models to customers to execute!

IDENTIFIERS

SETTYPE ASSIGN TYPE,2,PL Part type 2

Most statements will **not** have identifiers.

When used, an identifier should begin in Column 1 or 2.

Identifiers must be typed in upper case.

Identifiers are alphanumeric, should be three to eight characters, and **MUST** begin with a letter.

Identifiers should not conflict with reserved GPSS/H names.

OPERANDS

SETTYPE ASSIGN **TYPE, 2, PL** Part type 2

Depending on the Operation, there can be 0 to 9 Operands (most use no more than 3).

Some operands are required, others optional, depending on the Operation Code.

Operands must begin at or before Column 25 (this number can be explicitly overridden).

The first Operand is separated from the Operation Code by at least one space.

Operands must be typed in upper case.

Operands are generally separated from each other by commas—NO SPACES.

FOLLOWER LINES

Two statement types accept one or more “Follower Lines” immediately following the line containing the Operation Code.

The statement types are:

- **FUNCTION**
- **PUTPIC and BPUTPIC**

Follower Lines have their own syntax rules.

SAMPLE STATEMENTS

Can you identify the parts?

* What is this line?

* The next line is a FUNCTION statement with 1 Follower
* line. A value is returned based on the value of the
* simulation clock (AC1).

```
HOLD      FUNCTION  AC1,D5  
120,2/240,4/300,5/420,7/480,8
```

```
        ASSIGN      TYPE,2,PL      Part type 2
```

* The next line contains the "auxiliary operator" **SNF**
* Here, the GATE is 'open' if the Storage named "SYSTEM"
* is **Not Full**.
*

```
BEGIN    GATE SNF    SYSTEM  
  
        ADVANCE  
        TERMINATE  1
```


PUTPIC

Our First Meaningful Control Statement is PUTPIC.

PUTPIC is used to produce custom output to the screen or to a file.

The text of the output normally appears on a Follower Line immediately following the PUTPIC statement.

A Follower Line is read in its entirety and processed according to the rules for PUTPIC Follower Lines — not by the syntax rules for "normal" Control Statements.

USING PUTPIC

What about this example?

```
SIMULATE
PUTPIC      (90+6,99.44,'water')
* Tears are still *** percent *
END
```

Does the this model contain a comment?

No, “*” in column 1 of a PUTPIC follower line is the exception to the rule!

The result will look like this:

```
96 Tears are still 99.44 percent water
```

To run this one, type

GPSSH 2-4

STRUCTURE OF A SIMPLE PUTPIC: THE PUTPIC FOLLOWER LINE

The Follower Line is taken as literal text, except for asterisks, and decimal points adjacent to asterisks.

One or more contiguous asterisks - possibly with a decimal point are used as the "placeholder" for each value in a Follower Line.

The example below is incorrect, because the Output List contains two items but there is only one group of asterisks in the Follower Line.

```
PUTPIC      (3, 'days')      not enough placeholders
This class takes **
```

```
PUTPIC      (3.14)           placeholders with a decimal
The first three digits of PI = *.**
```

A NICE FEATURE OF THE ASTERISK PLACEHOLDERS

It is impossible for the printed representation of a value to “not fit” into the field defined by the asterisks!

If you try to plug a four-digit number into a two-asterisk field, GPSS/H will do it, pushing the rest of the picture to the right.

You are not forced (as with FORTRAN) to “pre-compute” the number of digits of numerical output.

```
      PUTPIC      (3*8*60, 'minutes')  
This class takes ** * all told.
```

...produces...

```
This class takes 1440 minutes all told.
```

If you try to plug a two-digit number into a four-asterisk field, GPSS/H will leave two blanks to the left of the number...this is useful for tabular numeric output data

SOME OTHER CONTROL STATEMENTS

By themselves, GPSS/H Control Statements form a simple programming language, summarized briefly below:

Control Constructs:

- **DO-ENDDO**
- **IF-ELSEIF-ELSE-ENDIF**
- **GOTO**

Data Declaration and Assignment:

- **Integer, Real, or Character data**
- **One- and Two-dimensional arrays**

Input/Output:

- **GETLIST for Input, PUTPIC for Output**

ELEMENTS OF GPSS\H MODELS

- Control Statements
 - ✓ PUTPIC
 - ✓ DO
 - ✓ END
- Blocks
- System Variables
 - ✓ AC1
 - ✓ Built - in
 - ✓ User- defined
- Entity Classes
- Transactions

GPSS/H TERMINOLOGY: “TRANSACTIONS” VS. “ENTITIES”

The phrase “Entity Class” refers to any of several GPSS/H language constructs:

- Computational Entities
- Statistical Entities
- Resource Entities

You will see this terminology in *Getting Started With GPSS/H* and elsewhere.

Terminology is usually language specific! What means one thing in GPSS/H may have a different meaning in other simulation languages.

TRANSACTIONS AND BLOCKS (OVERVIEW)

Transactions:

- Are created using Blocks
- Flow through Blocks
 - ✓ Being delayed for specified amounts of time
 - ✓ Being delayed due to conditions
 - ✓ Being delayed because of competition for scarce resources
- Are destroyed using Blocks

The 'actions' that Transactions take flowing through the Blocks of your model are what represents your system.

TRANSACTIONS AND BLOCKS (continued)

Transactions that are waiting to try to execute Blocks take turns being the *current* Transaction, according to rules we will study later.

Each Transaction that is waiting to try to execute a Block will get at least one chance to move into that Block during each *instant of simulated time* - or - before *time* moves forward.

For now, we do not need to know the mechanism that decides in what order a Transaction gets its try.

BLOCK STATEMENTS VS. CONTROL STATEMENTS

GPSS/H sorts out statement types before executing the model, as follows:

- Compiler Directives are processed, then discarded
- Comment lines are discarded
- A list of Control Statements is built and prepared for execution
- A list of Block Statements is built and prepared for execution

Regardless of how the Blocks and Control Statements are mixed together in your model file, they are *unshuffled* into two separate ordered lists - prior to execution.

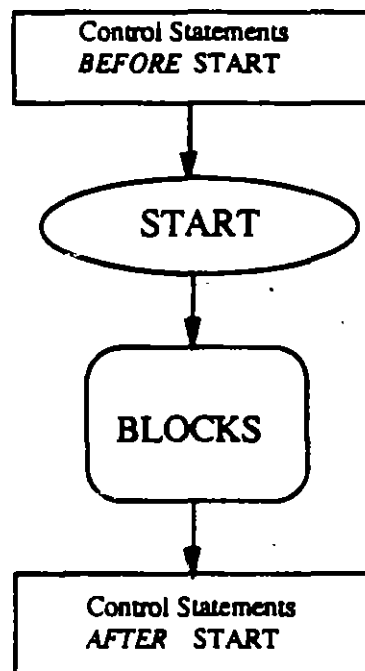
GPSS/H begins execution by executing each Control Statement, in sequence.

THE START CONTROL STATEMENT

START is a Control Statement that causes Blocks to begin execution.

Blocks will continue to execute until *Block execution is finished*.

After Block execution is finished, Control Statement execution resumes.



There may be more than one START Statement in a model.

CREATING AND DESTROYING TRANSACTIONS

Transactions are created using the GENERATE Block.

Transactions are destroyed using the TERMINATE Block.

Each time a TERMINATE Block is executed, the Termination Counter is decremented by the value of the first operand (A Operand) of the TERMINATE Block.

TERMINATE A

Consider the order of execution in the following model: (You can run it by typing: GPSSH 3-2)

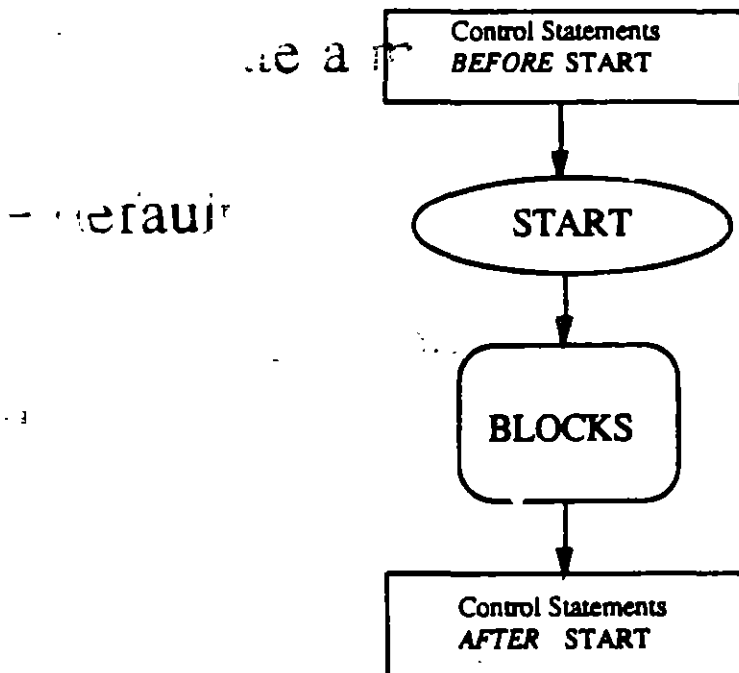
<u>MODEL</u>	<u>EXECUTION ORDER</u>
* SIMULATE	1 (CONTROL STATEMENT)
* GENERATE	3 (BLOCK)
TERMINATE 1	3 (BLOCK)
* START 1	2 (CONTROL STATEMENT)
* PUTPIC	4 (CONTROL STATEMENT)
BLOCK EXECUTION COMPLETE!	
* END	5 (CONTROL STATEMENT)

USING BPUTPIC

The model would run, but the BPUTPIC Block would never be executed by a Transaction!

```
SIMULATE
*
GENERATE
TERMINATE 1
*
START 1
*
BPUTPIC This statement changed!
BLOCK EXECUTION COMPLETE!
*
END
```

Why? Because the Xact is always **TERMINATED** before it reaches the BPUTPIC Block.



Try running this model by typing: GPSSH 3-3

RECOMMENDED ORGANIZATION

```
SIMULATE  
Control statement 2  
Block 1  
Block 2  
Block 3  
Control statement 3  
Control statement 4  
END
```

We recommend that your models should start with definition and declaration Control Statements (examples of which we'll study later), followed by all of the Blocks, followed by the Control Statements used to control the run (e.g. START).

TIME

Everything that you do has some time associated with it.

For example:

Driving to work
Talking on the phone
Assembling a widget

This is measured in REAL wall-clock time: seconds, minutes, etc.

In simulation, along with real time, there is also SIMULATED time and COMPUTER time.

SIMULATED time is the time duration represented in your model. For example, if you simulate your system for 1 year, it may take only seconds of real time to complete the run.

COMPUTER time is like the link between REAL and SIMULATED time. It's the real time that your computer took to complete your specified simulated time.

Problema No. 48

Objetivo. Aplicar la función GENERATE para diferentes situaciones

I) Construya un bloque GENERATE en el cual arriben transacciones cada 7 ± 2 unidades de tiempo, durante el tiempo de la simulación. Considerando que el tercer arribo ocurre al tiempo 21, enliste los posibles tiempos en los que se llevará a cabo el cuarto arribo. ¿Cuál es la probabilidad de que el cuarto arribo tome lugar en el tiempo 30? ¿Qué nivel de prioridad será asignado a las transacciones arribando en el bloque GENERATE?

II) Construya un bloque GENERATE en el que las trasacciones arriben:

a) Cada 6 unidades de tiempo.

b) Cada 6 unidades de tiempo, excepto la primera que debe generarse en el tiempo 15.

c) Cada 6 unidades de tiempo, pero solamente deben generarse 10 arribos en total.

III) Construya un bloque GENERATE en el cual las transacciones arriben cada 15 ± 5 unidades de tiempo:

a) Con un nivel de prioridad de cero.

b) Con un nivel de prioridad de 9.

IV) Los tiempos de interarribo de transacciones en un bloque GENERATE particular están distribuidos uniformemente sobre los enteros:

a) 4,5,6,7 y 8. Muestre un bloque GENERATE con el cual se obtenga el efecto deseado.

b) 4,5,6,7,8 y 9. ¿Puede construir un bloque GENERATE para resolver este inciso?

V) El bloque GENERATE mostrado en la figura E5 resulta en condición de error cuando el modelo es corrido en la computadora. ¿Puede explicar porqué?

f) ¿Porqué el bloque GENERATE de la figura E6 es inválido?

g) ¿Porqué el bloque GENERATE de la figura E7 es inválido?

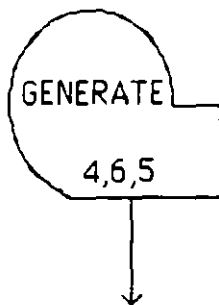


Figura E5

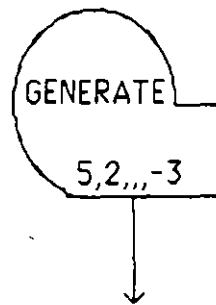


Figura E6

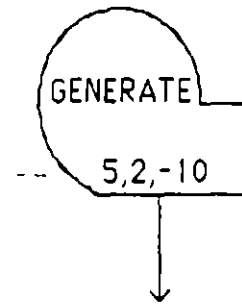


Figura E7

VI) La figura E8(a) es válida, mientras que la figura E8(b) es inválida. Explique porque.

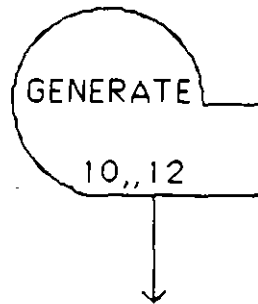


Figura E 8(a)

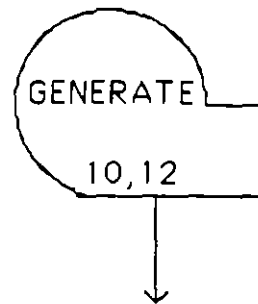


Figura E 8(b)

VII) Considere que solamente los bloques GENERATE mostrados en la figura E9 son usados en un modelo GPSS en particular, y que al tiempo 20 no se han removido aún las transacciones del modelo. ¿Cuántas transacciones con nivel de prioridad cero están en el modelo al tiempo 20? ¿Con nivel de prioridad 7? ¿Con nivel de prioridad 13?

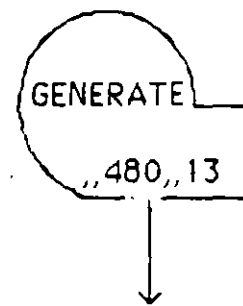
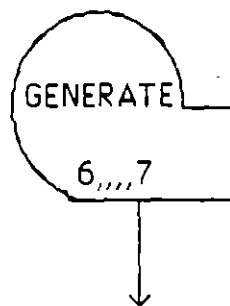
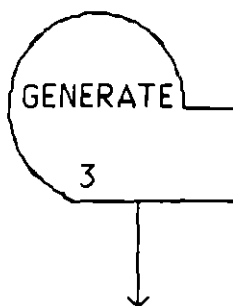


Figura E 9

VIII)

a) Suponga que es intención de un analista que la unidad de tiempo implícita en su modelo sea de 1 minuto. En un punto en particular del modelo, las transacciones se generan con tiempos de interarribo distribuidos uniformemente entre 3 y 6 minutos. El analista provee el bloque GENERATE mostrado en la figura E10(a) para producir este efecto. ¿Porqué es erróneo su trabajo?

b) Reconociendo su error, el analista decide que la unidad de tiempo implícita en su modelo sea 0.1 minuto. El modifica los operandos del bloque GENERATE, resultando la figura E10(b). ¿Cómo se comportan los diferentes valores de la variable aleatoria de tiempo de interarribo aquí?

c) Más tarde, el analista decide que el debe trabajar con una unidad de tiempo implícita más pequeña. El escoge 1 segundo como unidad. Muestre como aparecerá el bloque GENERATE de la figura E10(b) modificandolo para que corresponda a unidades de tiempo más pequeñas. Ahora, ¿Cuántos tiempos de interarribo diferentes pueden generarse en el bloque GENERATE?

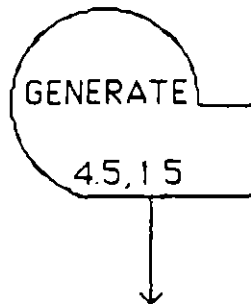


Figura E10(a)

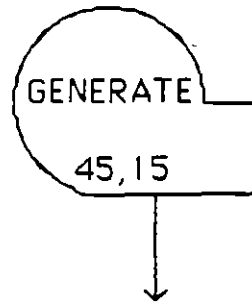


Figura E10(b)

IX) En un bloque GENERATE arriban transacciones cada 0.6 ± 0.2 días. Establezca cuales deben ser los valores de los operandos A y B para el bloque GENERATE si la unidad de tiempo implícita es:

- a) 0.1 días.
- b) 1/5 días.
- c) 0.3 horas.
- d) 0.1 horas

Problema No. 49

Objetivo. Aplicar la función GENERATE y TERMINATE unidas, en diferentes situaciones

Para cada uno de estos ejercicios asume que todos los bloques TERMINATE, a excepción donde se señala, tienen un valor por default de cero en el operando A.

- a) El segmento de dos bloques de la figura E1 se utiliza como "reloj" en un modelo GPSS. La instrucción START tiene un operando A con un valor de 8. ¿En qué tiempo de la simulación se parará la corrida?

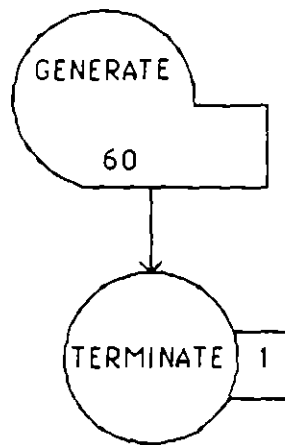


Figura E1

- b) Suponga que el operando A en la figura E1 del bloque GENERATE es cambiado por 70, y que las demás condiciones permanecen constantes, ¿En qué tiempo de simulación se parará la corrida?
- c) Suponga que el operando A en la instrucción START en el ejercicio 1 es cambiado por 3, y que las demás condiciones permanecen constantes ¿A qué tiempo de simulación el procesador GPSS detendrá el modelo? ¿Cuál será el valor final del Contador de Terminación para este caso?
- d) Un modelador decide utilizar dos segmentos de dos bloques para controlar la corrida de su simulación. Los dos segmentos se muestran en la figura E4. Si él usa una instrucción START con operando A de 25, ¿Cuándo se detiene la simulación? ¿ Es el valor final del contador de terminación cero o -1? Explique su respuesta.

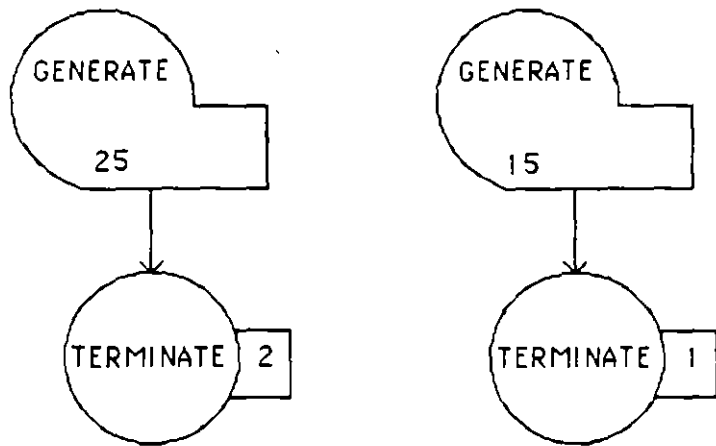


Figura E 4

PROBLEMA No. 50

Objetivo. Ser capaz de aplicar eficientemente la instrucción START y el bloque TERMINATE.

En el modelo de la figura 1 (ver página siguiente), el número de clientes servidos es una variable aleatoria.

- a) Muestra el diagrama de bloques que haría que la simulación se detenga después de que 25 clientes hayan sido atendidos.

- b) Muestra el diagrama de bloques que haría que la simulación se detenga cuando alguno de los dos eventos siguientes ocurra primero:
 1. Cuando el reloj alcance el valor 480
 - o
 2. Cuando exactamente 25 clientes hayan sido servidos.

PROBLEMA No. 51

Objetivo. Realizar una investigación sobre la siguiente duda o pensar que se pide y en base a un modelo, explicar lo que está pasando.

En las estadísticas de una línea de espera en GPSS ¿podría ser posible que el valor para el PORCENTAJE DE ENTIDADES CERO sea del 100 % (es decir que todas las transacciones fueran entidades cero) y el valor del CONTENIDO MAXIMO sea 1. Da la referencia (autor y número de la página) del libro o manual de donde se obtuvo la respuesta, o propón un ejemplo con resultados impresos de un modelo en GPSS, para contestar a la pregunta.

PROBLEMA No. 52

Objetivo. Realiza una corrida en minutos y posteriormente en segundos, al revisar el reporte ¿Qué encuentras? y explica que pasa.

Modifique la figura 1 sobre la suposición de que la unidad de tiempo es 1 segundo, no 1 minuto. Corra el modelo resultante para un tiempo de ocho horas de simulación. Compare las estadísticas del servidor, estadísticas de la línea de espera y el número de transacciones que entran al sistema.

PROBLEMA No. 53

Objetivo. En este problema nos enfocamos a los costos, así que deberás realizar la simulación y realizar varias corridas para saber cual es la mejor opción.

Los mecánicos de un taller de reparación llegan a una estación de suministro de herramientas cada 300 ± 250 segundos. El encargado del puesto de servicio tiene un tiempo de atención de: 280 ± 150 segundos para cada mecánico que hace una solicitud.

a) Construir un modelo de GPSS para esta situación, posteriormente corra el modelo para un tiempo de simulación de 8 horas. Suponga que, debido a la pérdida de producción de las máquinas descompuestas, existe un costo de 5 USC, por segundo que un mecánico espera en el puesto de suministro. En este sentido cual es el costo asociado con las 8 horas del tiempo de simulación.

b) Suponga que el encargado del puesto de suministro en (a) gana \$4 US dlls por hora. Este empleado puede ser reemplazado por otro que gana \$4.5 US dlls por hora y requiere 280 ± 50 segundos para atender una solicitud de los mecánicos. Simule esta situación para un día de 8 horas con el empleado alternativo y calcule el costo asociado con los mecánicos en espera.

c) ¿ Es mejor usar el primero empleado o el segundo?

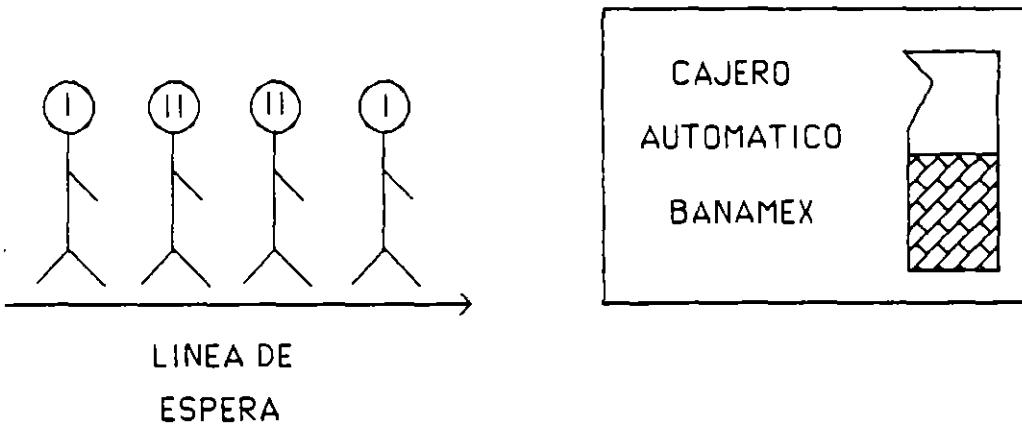
d) Simule el mismo modelo pero para 2, 3 y 4 días de 8 horas respectivamente, ¿cuál sería ahora su respuesta al inciso c ? Explique su respuesta.

PROBLEMA No. 54

Objetivo. Realizar la simulación en GPSS y revisar la corrida (en GPSSREPT), de la cual extraerás los resultados para poder obtener conclusiones.

Dos tipos de clientes arriban a una estación de un cajero automático, el primer tipo sólo desea conocer el saldo en su cuenta, la distribución de su tiempo de interarribo es de 5 ± 3 minutos. Los clientes del segundo tipo desean también retirar en efectivo, su distribución de tiempo de interarribo es de 6 ± 2 minutos. Al cajero le toma 2 ± 1 minutos dar una consulta de saldo y 3 ± 1 minutos, un proceso de retiro de efectivo.

Se pretende simular este sistema un tiempo de un día de 24 horas de trabajo y conocer las estadísticas del cajero y la línea de espera.



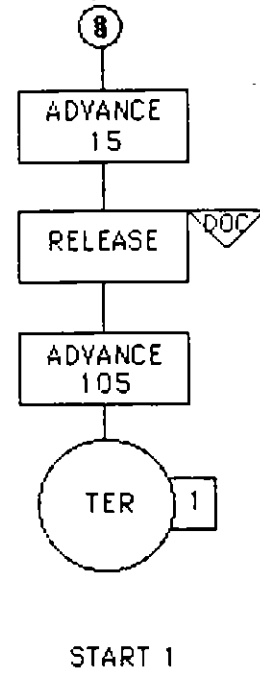
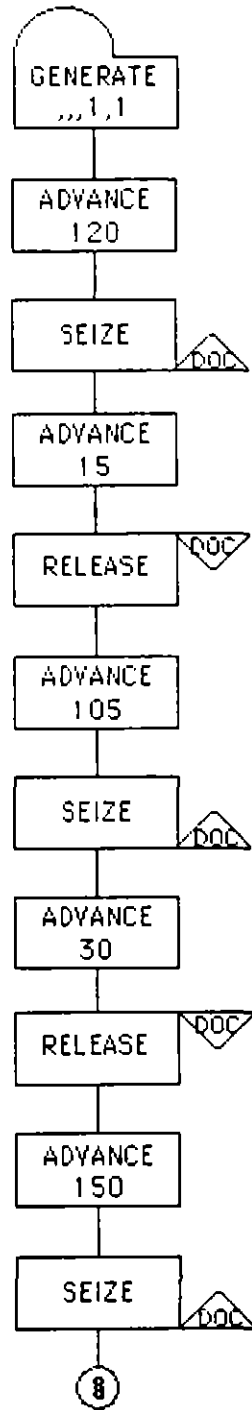
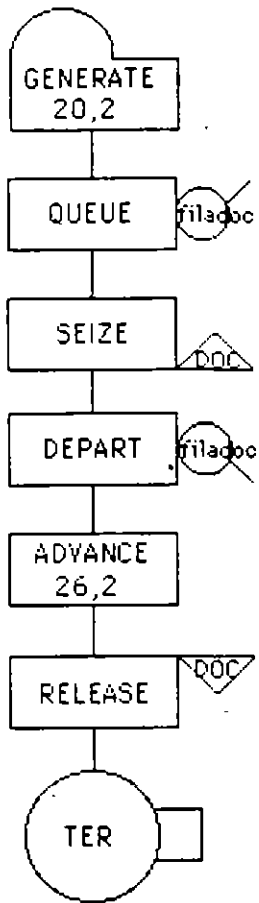
PROBLEMA No. 57

Objetivo. Utilizar los bloques vistos para resolver el problema y verificar un diagrama ya dado.

Con relación al problema no. 56, contesta a las siguientes preguntas:

- 1) Usando los bloques vistos hasta el momento, que harías con ellos para conocer, el tiempo promedio que le toma al doctor atender a un paciente. (Esto se refiere al inciso (a))

- 2) Un analista propuso el diagrama de la página siguiente para contestar el inciso (b) del problema 56. Menciona si esta bien o mal y porque.



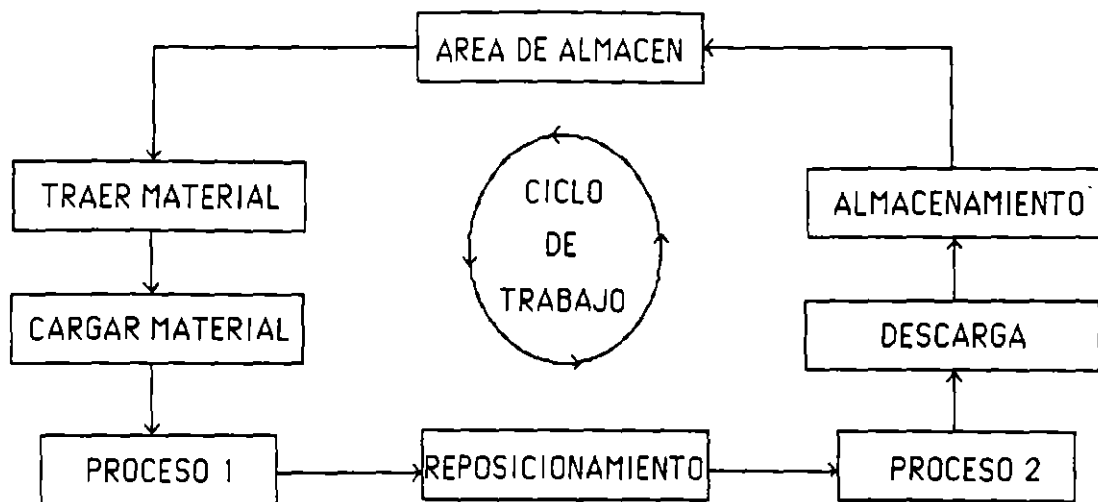
PROBLEMA No. 58

Objetivo. Realizar un problema donde las condiciones iniciales son las de arranque del proceso.

En una fábrica determinada, una máquina es usada para llevar a cabo el proceso de terminado de la materia prima. Los pasos requeridos en el proceso con sus respectivos tiempos están dados en minutos a continuación.

1. Traer la materia prima de la bodega (12 ± 3)
2. Cargar la materia prima en la máquina de terminado (10 ± 4)
3. Llevar a cabo el proceso 1 de terminado (80 ± 20)
4. Reposicionamiento del material en la máquina de terminado (15 ± 7)
5. Llevar a cabo el proceso 2 de terminado (110 ± 30)
6. Descargar la materia terminada de la máquina (10 ± 4)
7. Almacenar la materia terminada (12 ± 3)
8. Regresar al paso 1.

El ciclo a través del cual se lleva a cabo este proceso se muestra gráficamente en la siguiente figura:



Los pasos 1, 2, 4, 6 y 7 son pesados de hacer manualmente y es necesario un montacargas para realizarlos.

Existe un solo montacargas en la fábrica. El operador de la máquina de terminado no es el único que la usa. Existen otros obreros que la solicitan cada 30 ± 19 minutos y la mantiene en uso 25 ± 10 minutos.

Supóngase que el estado inicial del sistema se encuentra de tal forma que el operador está a punto de traer materia prima del almacén (cuando el operador ha hecho el proceso de descarga y almacenamiento, él retiene el montacargas para traer y cargar nuevo material).

Diseña el modelo para recolectar estadísticas del tiempo que tiene que esperar el operador de la máquina para obtener el montacargas y llevar a cabo el proceso 4 y otro conjunto de

estadísticas para obtener el montacargas y llevar a cabo la secuencia de pasos 6,7,1 y 2. También coleccionar estadísticas de la espera de los otros obreros para usar el montacargas. Realiza la simulación para un tiempo de 400 hrs.

PROBLEMA No. 59

Objetivo. El objetivo en este problema es conocer cuántos ensambladores debe tener esta fábrica para maximar ganancias para un tiempo de una semana de trabajo (40hrs).

La manufactura de cierta línea de piezas involucra un cierto número de procesos de ensamble, seguidos por un cierto tiempo de calentado en un horno. Ya que los hornos son costosos de mantener, varios ensambladores comparten el mismo horno, en el cual solo una pieza a la vez puede ser calentada. Un ensamblador no puede empezar un nuevo ensamble hasta que ha terminado de usar el horno.

Los costos y tiempos para este proceso están dados en las siguientes tablas:

Operación	tiempo requerido (minutos)
Ensamble	30 ± 5
Calentado	8 ± 2

a) Salario de cada ensamblador	\$ 3.75 /hora
b) Costo del horno	\$ 80 /día de 8 hrs.
c) Material	\$ 2 por pieza
d) Valor terminado de la pieza	\$ 7 por pieza

PROBLEMA No. 60

Objetivo. Realizar una modificación en las condiciones iniciales a un problema, no debe de causar mayores problemas, así que lo debes realizar rápidamente.

En el caso del problema del "horno" se supuso que inicialmente cada uno de los ensambladores está justo en el proceso de empezar un ensamble. Muestre como modificar el modelo (diagrama de bloques y listado del programa) para el caso de cuatro ensambladores para reflejar las siguientes condiciones iniciales:

- a) Tres de los ensambladores están justo en el proceso de empezar un ensamble; el cuarto está justo empezando a usar el horno para calentar una pieza.

- b) Un ensamblador está justo por empezar un ensamble; a uno le faltan 10 minutos para terminar un ensamble y dos están esperando usar el horno.

PROBLEMA No. 61

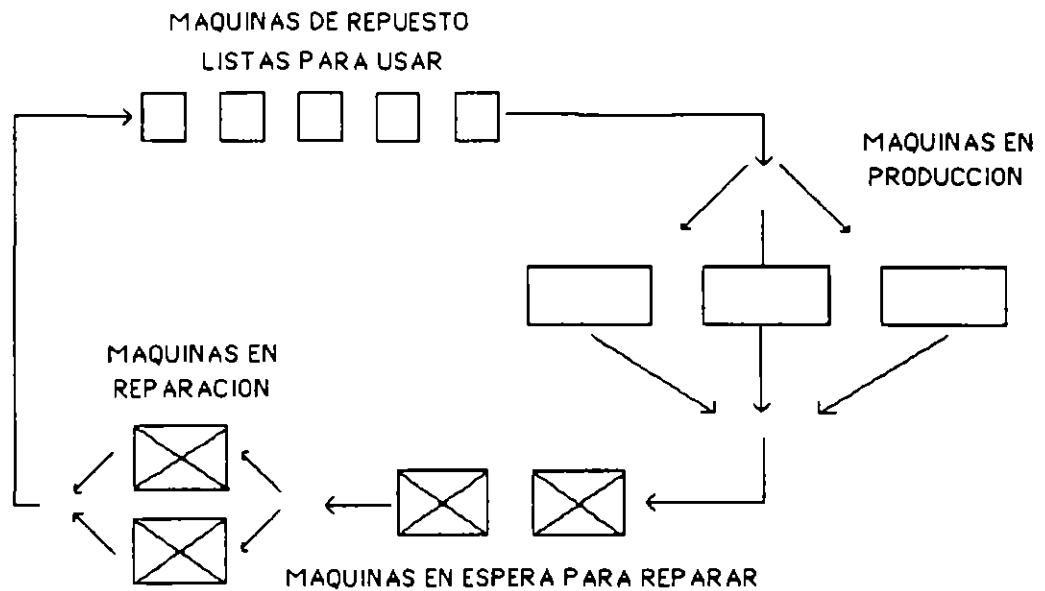
Objetivo. Buscar el menor número óptimo de mecánicos y de máquinas rentadas con el fin de minimizar costos.

Una fábrica de ropa tiene 50 máquinas propias y trabajan 8 horas diarias, 5 días de la semana. Cada máquina está sujeta a una falla aleatoria, cuando una máquina falla ésta es reemplazada por una máquina de repuesto inmediatamente, o cuando esté disponible, mientras tanto la máquina descompuesta es mandada a un taller de reparación, cuando ésta es reparada se convierte en una máquina de repuesto (ver la siguiente figura).

El administrador de la fábrica desea conocer cuántos mecánicos colocar en el taller y cuántas máquinas rentar para complementar las 50 propias. El objetivo es minimizar los costos bajo las siguientes suposiciones: los mecánicos son pagados \$ 3.73 dólares por hora y las máquinas para rentar están disponibles a \$30 dólares/día. El costo por hora de penalización por tener menos de 50 máquinas en producción es de \$20 dólares/máquina.

Basados en experiencias pasadas, el tiempo de reparación de una máquina es de 7 ± 3 hrs, y el tiempo hasta la falla es de 157 ± 25 hrs.

Simula para un tiempo de 3 años (40 horas semanales.)



PROBLEMA No. 62

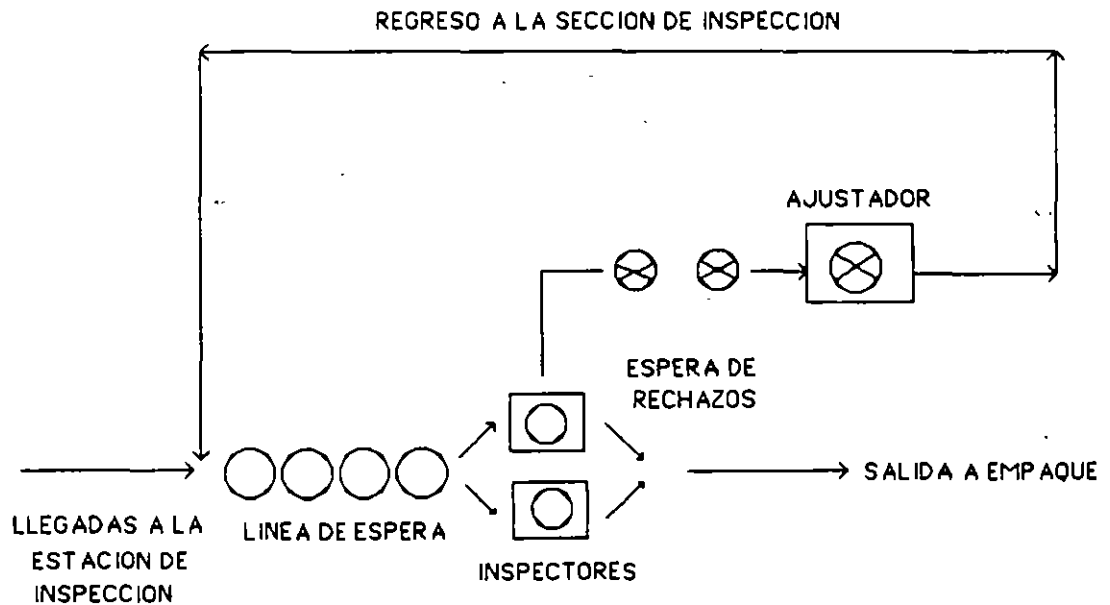
Objetivo. Utilizar el bloque Transfer (este bloque tiene varios usos y varios modos).

Un conjunto de televisores se mueve a través de una serie de estaciones de prueba, al final de estas estaciones el control vertical es probado. Si se encuentra que este control funciona inapropiadamente, se torna a una estación de ajuste. Después de este ajuste los televisores son mandados de una nueva cuenta a la estación de inspección, cuando los televisores pasan las pruebas entonces son mandados a empaquetado.

Los televisores llegan cada 5.5 ± 2 minutos. Dos inspectores trabajan independientemente en la estación final de inspección. El tiempo requerido para inspeccionar un televisor es de 9 ± 3 minutos.

Aproximadamente el 85% de los televisores pasan la inspección y continúan al empaquetado. El otro 15% son turnados a la estación de ajuste. El ajuste de control vertical requiere 30 ± 10 min.

- a) Diseñar un modelo en GPSS para simular este proceso.
- b) Diseñar el modelo de tal forma que se pueda determinar qué tan grandes deben ser los espacios antes de ambas estaciones.
- c) Simular para un día de ocho horas.



PROBLEMA No. 63

Objetivo. Desarrollar la habilidad de transformar o modificar problemas ya resueltos, donde ha habido un pequeño cambio.

Muestra cómo modificar el modelo del caso de la revisión de televisores (visto en clase) para que no más de 4 televisores puedan simultáneamente esperar para servicio en la estación de ajuste.

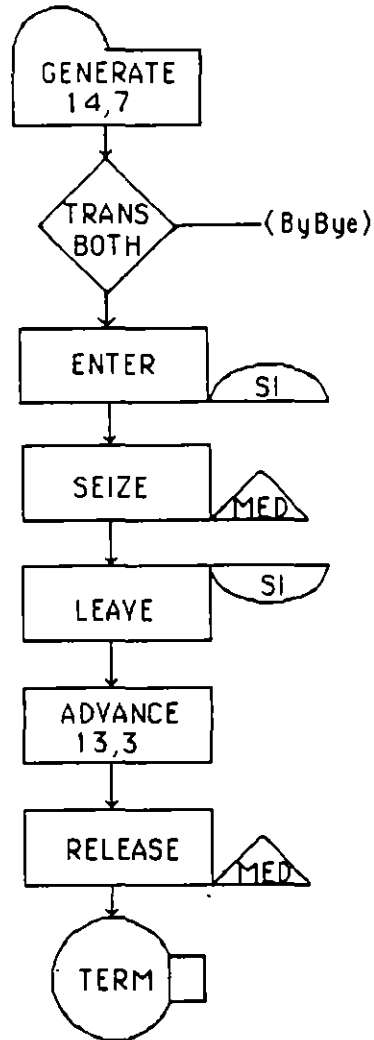
Un trabajador en la estación de servicio no puede empezar la próxima inspección hasta que la unidad precedente haya sido mandada a empaquetado o haya sido transferida a la estación de ajuste. Corre el modelo resultante y muestra las estadísticas de la línea de espera antes de la estación de inspección.

PROBLEMA No. 64

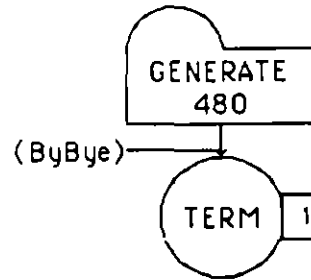
Objetivo. En muchas plantas sucede el siguiente problema (en forma análoga) donde se crea una línea de reparación de artículos, cosa que no debe suceder. Simula la siguiente situación y observa como se duplica el trabajo (volver a inspeccionar artículos ya inspeccionados).

En el mismo caso (revisión) de una línea de producción de televisores), se supuso sin reflejar la realidad ,que cada televisor después de pasar una vez por la estación de inspección tiene una probabilidad de 0.15 de requerir ajuste. Muestra como modificar el modelo bajo la suposición de que después de una o más visitas a la estación de ajuste, la probabilidad de requerir un nuevo ajuste es solo de 0.03

Segmento 1



Segmento 2



PROBLEMA No. 65

Objetivo. Desarrollar la habilidad de modificar un problema ya planteado para obtener nuevos resultados

a) Modifique la figura de la siguiente página para incluir la condición de que el 40% de los pacientes que se retiran del consultorio debido a que no encontraron silla disponible regresan después de 15 ± 5 minutos para probar otra vez. Los pacientes que no encuentran silla en la segunda prueba se retiran para siempre.

b) Modifique la figura de la siguiente página para introducir la siguiente variación en el comportamiento: El 20% de los pacientes que arriban al consultorio permanecen solo si el médico esta inmediatamente disponible. Los otros permanecen solo si encuentran silla disponible en el área de espera.

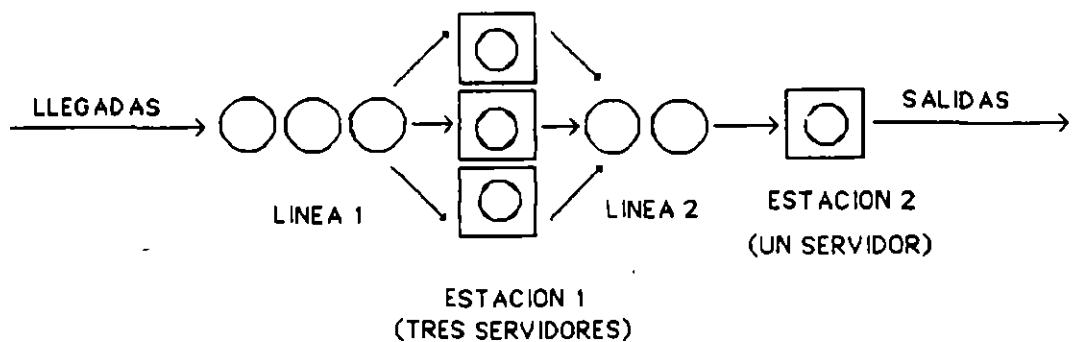
c) Modifique la figura de la siguiente página para incluir un oculista después del médico general, el 25% de los clientes considera la posibilidad de obtener un examen de la vista. Entre estos, solo el 20% esta dispuesto a esperar al oculista, si el oculista esta ocupado en ese momento los otros no esperan, el tiempo para un examen de la vista es de 15.3 ± 3 minutos.

Problema No. 66

Objetivo. Observar cómo un inventario en producción puede afectar a esta, caso a) y b).

a) Construir un modelo para representar el sistema de colas mostrado en la siguiente figura. Los arribos ocurren a la estación cada 115 ± 30 segundos. Los tiempos requeridos para dar el servicio en las estaciones 1 y 2 son 335 ± 60 y 110 ± 25 segundos, respectivamente. Diseña el modelo para medir el comportamiento de la línea de espera para las estaciones 1 y 2. Asuma que hay un espacio ilimitado entre las dos estaciones, así que no hay tamaño máximo en el cual no puedan ser acomodadas las entradas a las dos líneas de espera.

b) Asuma que, debido a las limitaciones de espacio, el número de unidades en la línea de espera 2, no puede exceder a 1. Los servidores de la estación 1 no pueden comenzar el servicio a la siguiente unidad hasta que la unidad precedente haya sido colocada en la línea de espera 2. Modifique su modelo en (a) para tomar esta restricción en cuenta.



PROBLEMA No. 67

Objetivo. Realizar un análisis sobre costos, el cual deberás hacer con la información que el reporte de la simulación proporciona.

Dos tipos de barcos llegan a un puerto donde son descargados. Existen dos remolcadores los cuales dan servicio en el puerto. Los barcos tipo uno, que son pequeños, requieren el uso de uno de estos remolcadores para atracar y desatracar. El tipo 2, los cuales son grandes, requieren el uso de ambos remolcadores para atracar y desatracar. Debido a las diferencias de tamaño, de los dos tipos de barcos, descargan en diferentes muelles y tienen diferente tiempo de descarga. Los datos aparecen en la siguiente tabla.

	Tipo de barco	
	1	2
Tiempo de interarribo (min)	130 \pm 30	390 \pm 60
Tiempo para atracar (min)	30 \pm 7	45 \pm 12
Número de muelles disponibles	6	3
Tiempo de descarga (hrs)	12 \pm 2	18 \pm 4
Tiempo para desatracar (min)	20 \pm 5	35 \pm 10

Simular todos los incisos para un tiempo de un año.

a) Construye un modelo en GPSS para este sistema, diseñándolo para estimar el tiempo agregado de espera de cada tipo de barco en el puerto. Los tiempos agregados de espera consisten del tiempo de espera por cualquier razón, es decir espera por un muelle, y/o espera por un remolcador.

Un barco esperando atracar no captura a un remolcador, hasta que el muelle esté disponible (¿qué puede pasar si un barco pudiera capturar un remolcador antes que un muelle estuviera disponible para el barco?). Además un barco tipo dos no captura una grúa hasta que ambas están disponibles.

b) Si existe un costo de \$350 U.S. y \$500 U.S. por hora, respectivamente, por tener a los barcos tipo uno y tipo dos en espera, y el costo de adicionar un remolcador es de \$250 USC por día ¿puede justificarse adicionar un tercer remolcador en el puerto?

c) Suponiendo que los costos por espera de los barcos son como en el inciso b), ¿a qué costo estimado por día, podría adicionarse otro muelle tipo dos?

PROBLEMA No. 68

Objetivo. Realiza la siguiente simulación y en base a los resultados podrás afirmar si es válido el número de casetas o no.

En la caseta de cobro de la autopista México-Querétaro existen seis estaciones de cobro. Cada una de estas puede atender un carro en 15 ± 3 segundos. Supón que a las tres en punto de la tarde en particular, hay cuatro estaciones abiertas y ningún carro esperando en las estaciones.

La razón media de llegadas a la cual los autos arriban se incrementa para posteriormente disminuir tal como se muestra en la siguiente tabla.

INTERVALO DE TIEMPO	(3:00-4:00)	(4:00-4:30)	(4:30-5:00)	(5:00-5:30)
razón media de llegadas (carros/hora)	1000	1250	1600	2000

INTERVALO DE TIEMPO	(5:30-6:00)	(6:00-6:30)	(6:30-7:00)
razón media de llegadas (carros/hora)	1800	1300	1000

El rango de variación en los tiempos de interarribo (uniformemente distribuidos) para cada intervalo de tiempo es \pm el 25% de la media.

Para compensar el incremento de tráficos durante las horas pico, una quinta caseta se abre a las 4:30 y la restante se abre a las 5:00.

Modela esta situación en GPSS (desarrolla los 7 puntos de un problema estandar en GPSS) para estimar el máximo y número promedio de carros en espera durante cada uno de los intervalos de tiempo dados en la tabla anterior.

Simula con una sola línea de espera y servidores múltiples.

PROBLEMA No. 69

Objetivo. Aprender a utilizar el bloque Assign, ver cuáles son sus ventajas y realizar el problema en GPSS así como la simulación e interpretación de los resultados.

Una tienda super 7 consta de 3 islas y un solo cajero. Los clientes llegan según un proceso de Poisson con un tiempo promedio de interarribo de 75 segundos. Después de llegar, cada cliente toma un carrito y puede ir a una o más islas seleccionando cierta cantidad de artículos como indica la siguiente tabla:

Isla	Probabilidad de ir a esa isla	Tiempo de estancia	No. de artículos seleccionados
1	0.75	120 \pm 60 seg.	3 \pm 1
2	0.55	150 \pm 30 seg.	4 \pm 1
3	0.82	120 \pm 45 seg.	5 \pm 1

Al terminar su recorrido por las islas los clientes pueden seleccionar adicionalmente 2 \pm 1 artículos en la caja. El tiempo para pagar depende del número de artículos comprados. El tiempo es de 3 segundos por artículo.

Construye un modelo en GPSS para simular 8 horas de trabajo de esta tienda. Mide la utilización del cajero y la máxima longitud de la línea de espera. Supón que no hay límite para el número máximo de carritos en uso en cualquier momento.

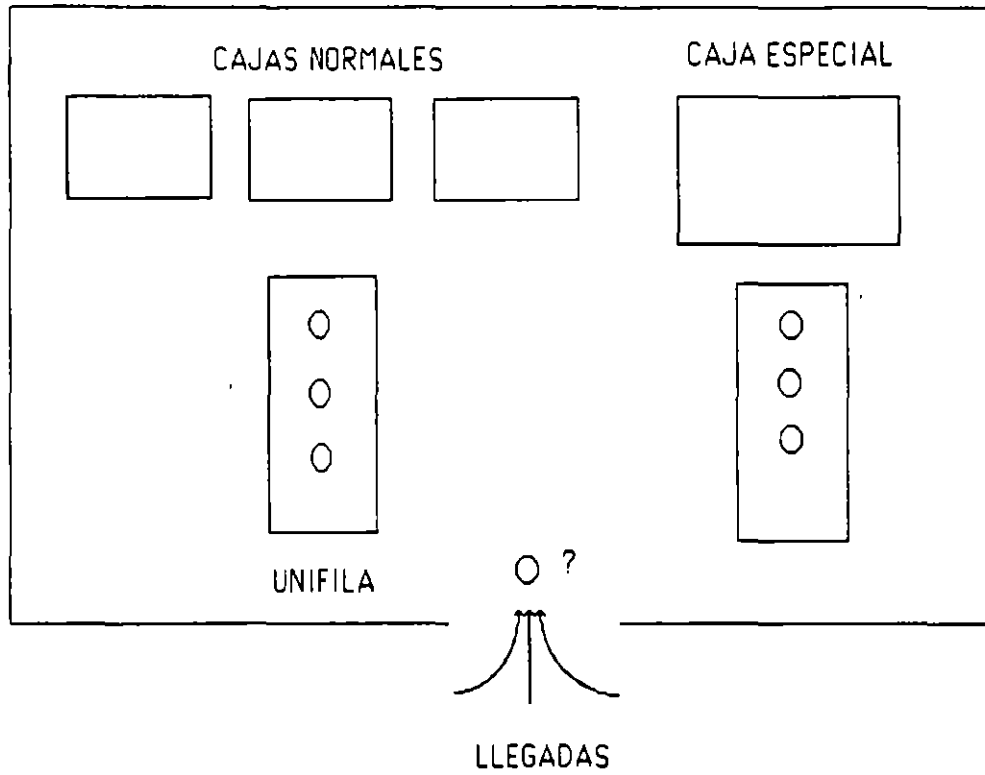
PROBLEMA No. 70

Objetivo. Utilizar los bloques Gate, Logic y Test en el siguiente problema. En este problema la simulación termina cuando el último cliente que está en el banco es terminado de atender, en el pasado fijábamos un tiempo de simulación y al concluir éste terminaba la simulación.

Para dar mejor atención a los clientes de un banco con solo necesidades de "cheques de caja", el banco ha asignado un solo cajero para dar este servicio. Aproximadamente el 40% de los clientes que arriban son de esta naturaleza, al llegar uno de estos clientes al banco es enviado a alguno de los cuatro cajeros para clientes normales sólo si alguno se encuentra en ese momento disponible. Los clientes arriban en intervalos descritos por una distribución uniforme con una media de 66 segundos y un rango de ± 10 segundos.

Los cheques de caja requieren de 120 ± 30 seg y las operaciones normales 360 ± 120 seg. El banco trabaja 7 horas diarias pero debe terminar de atender a las personas que al cumplirse las 7 horas se encuentren dentro del banco (véase la siguiente figura).

UN CASO BANCARIO



PROBLEMA No. 70

Objetivo. Comparar alternativas entre un sistema multilíneas y un sistema unifila

Los clientes arriban a un banco con una distribución de poisson con un promedio de 200 por hora. Existen ocho servidores disponibles en el banco todo el tiempo. Cada servidor tiene su propia línea de espera. Si un servidor esta desocupado cuando un cliente entra al banco, inmediatamente el cliente se va con ese servidor. Si no es así, se forma con el servidor con la menor línea de espera. La atención al cliente se basa en el primero que llega primero en ser atendido, hace su operación y sale del banco.

Las diversas operaciones que se pueden realizar se clasifican en cinco categorías. La frecuencia relativa de cada categoría y su correspondiente tiempo de servicio se muestra en la tabla siguiente. En cada categoría el tiempo de servicio esta exponencialmente distribuido. Ningún cliente puede entrar en dos o más categorías en una visita al banco.

Categoría	Frecuencia Relativa	Tiempo de Servicio (segundos)
1	.10	45
2	.19	75
3	.32	100
4	.24	150
5	.15	300

El gerente del banco ha notado que el tiempo en las líneas de espera es alto. A él le gustaría reducir el tiempo de espera para el cliente, pero preferiría no tener que contratar dos o más empleados más para lograr esto. Él sabe que en otro banco vecino se introdujo un sistema de "Línea rápida". En dicho sistema, el cliente llega al banco y forma una sola línea. Cuando un servidor se desocupa, el cliente de la cabeza de la línea se va con ese servidor. Se presume que el sistema de Línea Rápida reduce el tiempo de espera, eliminando movimientos lentos en la línea cuando un servidor se encuentra con un cliente con el que va a tardar un período de tiempo largo. Ya que esta conjetura es razonable, el gerente decide experimentar con un sistema de Línea Rápida.

Construye un modelo en GPSS con la información de las diversas líneas de espera con las que opera el banco actualmente y con la proposición de cambiarlo a una sola Línea Rápida. Corre el modelo, para cinco diferentes horas al día en cada caso, después compara y analiza resultados. Controla las condiciones experimentales para que cada alternativa sea comparada de la manera más válida posible. Esto significa que el tiempo de interarribo debe ser el mismo para ambos casos y también el tiempo de servicio para los clientes. Por ejemplo, si el cliente veintiuno arriba al sistema multilíneas el tercer día, 16 minutos después de que halla comenzado el día y requiere 165 segundos de servicio; lo mismo debe ser cierto para el cliente veintiuno que llegue al sistema con una sola línea de espera el tercer día. Finalmente, en caso de empate entre llegadas de un cliente y la terminación del servicio, la terminación del servicio debe ocurrir primero. (Con un sistema multilíneas, esto debe tener influencia en los arribos y en el servidor que va a escoger el cliente)

PROBLEMA No. 71

Objetivo. Minimizar los tiempos de recorrido, buscar una mejor distribución de los recursos (cuántos autobombas y vehículos de emergencia deben haber en cada estación).

Una comunidad está considerando donde localizar un nuevo departamento de bomberos. Por propósitos de planeación la comunidad está dividida en 6 áreas. Estimaciones de la frecuencia de los incendios y llamadas para asistencia han sido compiladas de registros pasados. Además, se ha estimado el tiempo de viaje de una área a otra. Cada departamento de bomberos tiene dos autosbomba y dos vehículos de emergencia. Las llamadas son siempre atendidas primeramente por el departamento de bomberos más cercano.

Si un departamento de bomberos no satisface completamente una llamada, la otra estación deberá responder a la misma. Si tampoco puede satisfacer completamente la llamada, se pierde.

El 50% de las llamadas requieren un autobomba, 25% requieren un vehículo de emergencia y 25% requiere ambos, un autobomba y un vehículo de emergencia.

Suponga que el tiempo que una unidad permanece en la escena es 45 minutos y los de viaje entre áreas son:

Tiempo medio de viaje (min)

AREA	1	2	3	4	5	6
1	5	8	10	12	6	7
2	8	5	9	8	7	7
3	10	9	5	15	8	9
4	12	8	15	5	9	13
5	6	7	8	9	5	10
6	7	7	9	13	10	5

Además suponga que el tiempo entre incendios o emergencias para las seis áreas tienen un promedio de 20, 40, 60, 30, 50 y 60 min. respectivamente con distribución exponencial.

La localización de la estación actual de bomberos está en el área 5.

Realiza una simulación de 24 hrs. para conocer donde localizar la nueva planta de bomberos bajo la hipótesis de que lo que se requiere, minimizar, es el tiempo para llegar al siniestro.

PROBLEMA No. 72

Objetivo. Mientras más información tengas de un problema al momento de realizar la simulación, será cada vez más real. En el siguiente problema deberás encontrar el porcentaje de utilización de las cajas (hay dos tipos), el tiempo de espera en la fila, etc.

Un supermercado consiste de cinco pasillos y 11 cajas de las cuales 6 son utilizadas como cajas rápidas y las otras 5 como cajas normales. Los tiempos de interarribo de los clientes se comportan de la siguiente manera:

Hora	7:00 - 8:00	media = 50 seg	rango = 10 seg
	8:00 - 12:00	media = 40 seg	rango = 12 seg
	12:00 - 20:00	media = 10 seg	rango = 10 seg

El supermercado abre sus puertas a las 7:30 y de todos los clientes el 25% de ellos toma un carrito (clientes grandes) de los 500 que hay, si no encuentran carrito abandonan el supermercado, de el 75% restante (clientes chicos) solo el 50% compran algún artículo y el resto permanece curioseando un tiempo aproximado de 300 seg.

De los clientes que sí compran, pasan a los pasillos de la siguiente forma:

Pasillo 1	10%	Pasillo 3	30%
Pasillo 2	40%	Pasillo 4	5%
Pasillo 5	15%		

El número de artículo que compra cada cliente varía dependiendo del pasillo a que se dirija.

Pasillo 1	compra de 5 a 8 artículos
Pasillo 2	compra de 1 a 3 artículos
Pasillo 3	compra de 1 a 8 artículos
Pasillo 4	compra de 2 a 5 artículos
Pasillo 5	compra de 30 a 40 artículos

Se considera un tiempo de 45 segundos para escoger cada artículo, una vez seleccionados sus artículos, los clientes pasan a las cajas, donde hay 2 tipos:

CAJAS RAPIDAS: Solo para los cliente chicos (sin carrito). Estas son 6 y cada una tiene una cola de no más de 10 personas formadas, si un cliente chico llega a esta sección y no encuentra alguna caja con menos de 10 personas en cola, éste pasará a una caja normal (Para cliente con carrito).

CAJAS NORMALES: Estas son 5 y se usan indistintamente para clientes grandes y/o chicos (después de no haber encontrado una caja rápida con menos de 10), si algunos clientes no encuentran alguna caja con una cola menor de 30 personas, entonces abandonarán el supermercado, dejando su carrito sin comprar nada.

Los clientes que pasan a las cajas, tienen un tiempo de cobro de 8 segundos por artículo. El supermercado cierra sus puertas a las 20:00 horas.

4.5

**CASO PRACTICO
DE SIMULACION
EN UNA
EMPRESA
CEMENTERA**

SIMULACIÓN DE UN SISTEMA DE EMBARQUES EN UNA PLANTA CEMENTERA

Marco Antonio Montufar Benítez y Angel Durán Díaz

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México
Departamento de Ingeniería Industrial y de Sistemas

19 de octubre de 1995

1 INTRODUCCION:

La crisis que México empezó a vivir a finales de 1994, esta obligando a las empresas mexicanas de manufactura y servicios a ser más competitivas para conseguir sus metas o permanecer en el mercado. Conseguir tales expectativas, requiere de contar con procedimientos de operación que sean seguros. Ante tal necesidad las compañías deben de hacer uso eficiente de la tecnología disponible.

En el diseño e implementación de un sistema nuevo (de manufactura o de servicios), la Alta Gerencia debe contar con el apoyo de herramientas que le permitan de alguna forma ver hacia el futuro el desempeño de su sistema ante la implementación de alguna forma de operación nueva y que además reduzca el riesgo de llevarla a cabo en la realidad. La herramienta que ha sido usada para tales fines tanto en la ingeniería industrial como en otras áreas del conocimiento es la Simu-

lación. Varios ejemplos de aplicación son dados en Hoover y Perry 1989. Esta herramienta consiste de una serie de pasos que da alguna u otra forma son representados por varios autores (Hoover y Perry 1989, Law y Kelton 1991, McHaney 1991, Kleijnen y Groenendaal 1992, Hoshyar y Nuila 1993).

La industria del cemento en el estado de Hidalgo y estados aledaños se ha desarrollado desde hace varias décadas. Empresas como Tolteca, Cruz Azul y Apaxco son de reconocido renombre en la zona.

Un problema que en la actualidad están enfrentando estas empresas es la insatisfacción de sus clientes en cuanto al tiempo de estancia en la planta, en que incurren sus unidades de transporte como camiones, camionetas, tractocamiones y vagones de ferrocarril.

En este estudio se investigará mediante la simulación varios escenarios en la forma de operación en la zona de embarques en la planta, y con los resultados del análisis estadístico, proponer los cambios que repercutan en un mejor desempeño de la planta en cuanto al tiempo promedio de estancia de los vehículos de transporte.

2 DEFINICION DEL PROBLEMA

Cementos Tolteca (Planta Atotonilco) es una empresa situada en el estado de Hidalgo y se dedica a la producción y comercialización de cemento en bultos de 50 kg. y a granel (polvo) en furgones ferroviarios. Empresas como ésta, actualmente basan sus ventajas competitivas en la calidad del cemento, precio del producto y tiempo de respuesta al cliente. Los dos primeros factores frecuentemente no son flexibles de cambiar dadas las restricciones económicas y tecnológicas de la empresa, pero en cuanto al tiempo de respuesta se puede improvisar mejoras, que pueden ser determinantes para que un cliente decida no cambiar de proveedor. El costo promedio en una empresa de este tipo, se ha estimado en N\$8 000 por camión perdido (venta no realizada).

Encuestas en la empresa mencionada anteriormente han reflejado insatisfacción por parte de sus clientes en cuanto al tiempo de espera en la planta. Datos recabados durante el mes de octubre de 1994

indicaron tiempos promedio máximos de hasta 7.39 hrs. y mínimos de 3.1 horas. Ante tal magnitud promedio de dichos tiempos resulta clara la necesidad de reducirlos.

En este trabajo se analizará diferentes alternativas en la forma de operación en la zona de embarques de la planta que mejoren el desempeño de la empresa al ofrecer menores tiempos de estancia.

La técnica utilizada para conseguir tal propósito será la simulación, en particular se harán modelos en el lenguaje de simulación GPSS (General Purpose System Simulation (Schriber, 1991)). Para evaluar las diferentes alternativas propuestas se hará un análisis estadístico de las variables de interés (Montgomery, 1991).

Los pasos seguidos en el estudio de simulación serán :

1. Formular el problema y planear un modelo:
En este paso debe empezar con un claro establecimiento de los objetivos globales de estudio.
2. Colectar datos y definir un modelo:
La información y datos (si están disponibles) deberán ser colectados sobre el sistema de interés y ser usados para especificar procedimientos de operación y distribuciones de probabilidad de las variables usadas en el modelo.

3. Validar:

La Validación es un proceso que se debe realizar a través del estudio entero de simulación, (aunque generalmente sólo se hace en algunos puntos como en este paso 3). En la construcción del modelo es importante para el modelador involucrar a personas quienes trabajan intimamente con el sistema a simular.

4. Construcción de un programa de computadora y Verificarlo:

El modelador debe decidir cuando programar el modelo en un lenguaje de propósito general como FORTRAN, PASCAL o C, o en un lenguaje especialmente diseñado, tal como: GPSS, SIMAN, SIMSCRIPT II.5 o SLAM (Law y Kelton, 1991, cap.3). Técnicas para verificar los programas deben ser realizadas en este paso.

5. Hacer corridas piloto:

Las corridas piloto del modelo verificado deben ser hechas con propósitos de validación en el paso 6.

6. Validar:

En este paso las corridas piloto pueden ser usadas para dar información sobre la sensibilidad de la salida del modelo a pequeños cambios en los parámetros de entrada. Si la salida cambiara grandemente, una mejor estimación de los parámetros de entrada debe ser obtenida.

7. Diseño de experimentos:

Aquí se debe decidir que diseños de sistemas simular, ya que como frecuentemente ocurre en la práctica hay más alternativas que las que uno puede simular. En su lugar, usando datos de salida de las corridas de producción de ciertos sistemas seleccionados y también técnicas de diseño de experimentos, el analista puede decidir cuáles sistemas adicionales simular.

8. Hacer corridas de producción:

Aquí las corridas de producción son hechas para proporcionar datos del desempeño de los diseños de sistemas de interés.

9. Análisis de datos de salida:

Las técnicas estadísticas son usadas para analizar datos de las corridas de producción. Las metas típicas son construir intervalos de confianza para las medidas de desempeño de un diseño particular de sistema.

10. Documentación, presentación e implementación de resultados:

Dado que los modelos de simulación son frecuentemente usados para más que una aplicación, es importante documentar las suposiciones que fueron hechas en el modelo, así también como el programa mismo.

Si los resultados de un estudio de simulación nunca son implementados es probablemente por un fracaso.

Además, resultados de modelos altamente creíbles son más probables de ser usados.

3 NUESTRO MARCO DE REFERENCIA

Un doctor examinando un paciente, observa síntomas para determinar si esta enfermo o goza de buena salud. Presión sanguínea, temperatura del cuerpo y pulso son algunas de las cosas a checar. Un analista examinando una fila de un sistema también ve los síntomas de si es saludable o enferma. Estos síntomas son conocidos como medidas de desempeño (MOP).

La observación del sistema de colas es una llave a la formulación del análisis del sistema. Esto contesta directamente a las preguntas: Cuáles son los datos?, y cuáles son las condiciones? Entendiendo como opera actualmente el sistema actual debería ser la base para la elección de una nueva forma de operación. Las observaciones deben proporcionar una idea para ayudar a contestar a las pregunta: Qué es lo desconocido? Esto es, esto ayudaría a identificar, definir y enmarcar el problema.

MEDIDAS DE DESEMPEÑO.

Uno podría considerar dos perspectivas en la medición del desempeño de un sistema de colas. Desde el punto de vista del cliente, el mayor interés es la calidad del servicio recibido. Desde la perspectiva del

servidor (o perspectiva del empleado del servidor), los mayores intereses son los costos de proveer el servicio y el impacto de la calidad del servicio en el negocio (esto es, creación de ingresos). Las medidas de desempeño permiten cuantificar estos intereses.

Los siguientes párrafos dividen las MOPs a lo largo de las líneas de los intereses del cliente y los intereses del servidor. Pero antes de discutir las medidas de desempeño, algunos términos básicos tienen que ser reconocidos:

Tiempo de arribo: el tiempo en el que cliente arriba a la cola.

Tiempo de partida: tiempo en el que cliente completa su servicio y deja el sistema de colas.

Tiempo de partida de la cola: tiempo en que el cliente deja la cola para entrar al servicio.

Tiempo en cola: tiempo de partida de la cola menos el tiempo de arribo.

Tiempo de servicio: tiempo de partida menos tiempo de partida de la cola.

Tiempo en el sistema: tiempo de partida menos tiempo de arribo = tiempo en cola más tiempo de servicio.

MEDIDAS DE DESEMPEÑO DEL CLIENTE

En muchas situaciones, los mayores intereses del cliente son la longitud del tiempo consumido en la cola y en el servicio, y el costo asociado con este tiempo. Estos están reflejados en las siguientes me-

medidas de desempeño:

Tiempo en cola: en esencia, una espera corta es mejor que una espera larga.

Tiempo de servicio: el tiempo de servicio debería ser tratado por separado del tiempo en cola si los clientes encuentran su tiempo de espera en la cola más costosa que su tiempo en el que fueron servidos (o viceversa). Si son percibidos más o menos igual, el tiempo en el sistema puede ser sustituido por el tiempo de servicio y el tiempo en cola.

Costo de espera: el tiempo de espera podría ser más costoso para unos clientes que para otros, en cada caso algunos esquemas de peso podrían usarse para combinar los diversos tiempos de espera. El tiempo combinado de espera puede representar al costo.

Proporción de trabajo terminado a tiempo: los clientes podrían tener fechas límite para ser atendidos. Si no son atendidos en el tiempo límite, se puede incurrir en un castigo.

Tardanza: si un trabajo no se termina en el tiempo límite, es mejor que se retrase un poco y no demasiado. La tardanza es igual a cero si el trabajo es terminado a tiempo y es igual al tiempo de salida menos el tiempo de fecha límite si el trabajo se atrasa.

Debido a que muchos MOPs son variables aleatorias, deben ser definidos más precisamente. No es suficiente decir que los MOPs es el tiempo en cola. El MOP deben ser especificado como el tiempo promedio en la cola, desviación estandar

del tiempo en la cola, o máximo y mínimo tiempo en la cola. Los MOPs pueden también cambiar en un modo previsible (no aleatoriamente) con el tiempo del día, de una semana o algo similar.

En resumen las medidas cuantitativas, tanto los factores cualitativos como la espera del ambiente deben ser examinados. Está el cliente inactivo mientras espera? Los clientes reciben la información adecuada sobre que tanto tienen que esperar?, pueden los clientes sentarse?, está la sala de espera saturada?, Existe ventilación adecuada? Todos estos factores son importantes para el desempeño de la fila.

MEDIDAS DE DESEMPEÑO DEL SERVIDOR

El servidor esta interesado con minimizar del costo del servicio proporcionado y la atracción y permanencia de los clientes. El costo del servicio proporcionado está reflejado en el tiempo de servicio y la utilización del servidor; la atracción y permanencia de los clientes está reflejado en las llegadas y los que desertan de la fila y los que se niegan a entrar. Una lista de MOPs se muestra a continuación:

Tiempo de servicio: Un tiempo de servicio corto es un indicador de una operación más eficiente.

Proporción de utilización: Esta es la proporción del tiempo que los servidores están ocupados atendiendo a clientes. La proporción de utilización es un número en-

tre el cero y el uno, y entre más cercano es el número a uno más eficiente es la operación.

Cantidad procesada: Es la razón a la cual los clientes son servidos. En algunos sistemas, la cantidad procesada incrementa conforme la cola crece porque los clientes están mejor preparados antes de que ellos alcancen al servidor; en otros, el servicio deteriora cuando la cola se hace grande debido a la fatiga del servidor.

Razón de llegadas: La razón de llegadas es una medida de la cantidad de negocios. Una gran razón de llegadas es deseable porque significa mayores ingresos.

Proporción de deserciones: Cada cliente que deserta se traduce en un negocio perdido. Las deserciones pueden ser medidas como la proporción de clientes que desertan después de haber estado en fila, o como la proporción de quienes no deciden entrar a la fila después de ver su longitud. La deserción es el más difícil MOP de medir porque los clientes que desertan puede que nunca hayan sido registrados.

Longitud de fila: Una fila es más difícil de acomodar que una fila corta por el espacio requerido.

Los aspectos cualitativos incluyen los siguientes: Los servidores tienen algo que hacer cuando no están atendiendo a los clientes? Los servidores son completamente productivos? Existen problemas en los servidores? Los servidores son aten-

tos? los servidores están ociosos cuando los clientes están esperando en la fila?

RELACIÓN ENTRE LAS MEDIDAS DE DESEMPEÑO Y LAS METAS DE LA ORGANIZACIÓN

Es importante ser conciente de como las medidas de desempeño se ajustan dentro de las metas globales de la organización. Si la organización es privada, las metas casi siempre maximizan la rentabilidad a largo plazo. En organizaciones del gobierno, las metas son más confusas, pero probablemente contemplan la maximización del bienestar de sus componentes. Cuando evaluamos las alternativas de colas, a uno le gustaría medir directamente el impacto de la alternativa sobre las metas. Raramente es esto posible. En vez de eso, uno debe ver las medidas de desempeño como indicadores del impacto potencial sobre los beneficios y el potencial del impacto sobre el bienestar. Como lo ilustra la figura 1, ambos tiempo en el sistema y longitud de la cola afectan la razón de llegadas y deserciones, que a su vez afecta los ingresos. La utilización, longitud de cola y tiempo de servicio afectan los costos. Combinando costos e ingreso se determina la rentabilidad. Las medidas de desempeño son representantes de las metas globales, las cuales, en la práctica, no puede ser directamente medidas.

4 METODOLOGIA USADA PARA TRATAR EL PROBLEMA

La metodología para abordar el problema está basada en el concepto de Administración Total de la Calidad (TQM) la cual resumimos en las siguientes fases:

FASE 1. Recopilación de datos.

a) Clarificar datos y objetivos:

- Hora de entrada de transportes.
- Hora de salida de transportes.
- Tiempos de estadía.
- Número de camiones por hora.
- Número de camiones por día.
- Número de camiones express.

b) Desarrollar definiciones operativas:

- Medición de tiempos de estadía obteniéndolos con la diferencia entre entrada y salida de cada camión.
- Obtener histogramas de llegada de camiones a través del día y a través de la semana, obteniendo ésto mediante la hora de llegada.

c) Planificar consistencia y estabilidad de datos:

- Obtención de datos y promedios de 2 meses típicos.

d) Comenzar la recopilación de datos.

e) Continuar en la mejora de la consistencia y estabilidad de los datos.

FASE 2: Identificar las causas fundamen-

tales de los problemas.

- a) Identificar causas potenciales.
- b) Verificar las causas con datos.

FASE 3: Desarrollar soluciones apropiadas

- a) Describir la necesidad:
 - Reducir tiempo de estadía de camiones de carga.

- b) Definir metas y criterios:
 - Reducir tiempo de estadía del cliente a 60 minutos mediante la implementación de soluciones haciendo uso de la simulación.

- c) Identificar restricciones:
 - No se dispone de dinero para inversiones fuertes.

- d) Generar alternativas:
 - Sustituir a los operarios que van a desayunar o a comer.
 - Programar camiones de los clientes de una manera uniforme a través del día y de la semana. - Trabajar en domingo.
 - Reducir la venta de cemento envasado y aumentar a granel.
 - Tener un controlador de tráfico.

- e) Evaluar alternativas:
 - Se pueden evaluar las alternativas según su facilidad de implementación.

5 DESCRIPCIÓN DEL PROCESO:

Para conocer el proceso general fue necesario visitar la planta durante una semana, en la cual se obtuvieron datos de las áreas de embarques y envasado.

A partir de un cursograma analítico se identificó la ruta que sigue un camión; para propósitos de nuestro estudio dividimos el sistema en cuatro procesos principales:

- Caseta de Documentación, Entrada y Salida: Es el lugar físico en donde se documenta las horas de entrada y salida de un camión, y entregas de recibos.

- Báscula; Entrada y Salida: es el área en donde se registra el peso de los camiones (al entrar y al salir), para verificar que la carga con la que sale sea correcta (por diferencia de pesos).

- Envasadoras 1 y 2: Este tipo de envasadoras tiene una capacidad de envasado de 16.67 costales/min, sólo se pueden envasar 4 costales a la vez y el peso de cada costal es de 50 kg, las bandas de carga son retráctiles, evitando que el camión se tenga que mover al ser cargado. Debido a que el alimentador de cemento es fijo, el operario se tiene que desplazar enfrente de los 4 alimentadores (en paralelo) en un asiento móvil para acomodar el costal vacío.

- Envasadora 3: Esta envasadora tiene capacidad de 33.33 costales/min (envasa 8 costales a la vez), esta envasadora cuenta con dos bandas fijas. Cabe mencionar que las 2 bandas tienen un mismo alimentador y sólo puede ser cargando un camión al mismo tiempo mientras otro se acomoda en la otra banda. Los camiones se deben mover constantemente al ser cargados. El alimentador de cemento es giratorio, por lo cual, el operador se encuentra de pie acomodando un costal vacío.

RUTEO DE CAMIONES:

Un camión llega, se estaciona, y se detiene a documentar en la caseta. Posteriormente espera a ser llamado para pasar por la báscula y ser pesado; el peso se registra e inmediatamente pasa a formarse a alguna de las envasadoras que él mismo escoge. Una vez que llega a la envasadora es cargado con el número de costales solicitados. Ya cargado, espera su turno para pasar a la báscula, para verificar si la carga es correcta, sino tiene que regresar a la envasadora para que se le agregue o se le quite carga. Una vez con el peso correcto, el camión puede retirarse.

SITUACIÓN ACTUAL DE LA PLANTA.

Características:

Los camiones son diferenciados por el servicio requerido en tres categorías:

a) Normales: Son camiones que llegan sin cita previa.

b) Express: Estos camiones llegan con una cita.

c) Granel: Estos cargan más de 40 toneladas en otro lugar dentro de la planta, por lo cual no afectarán nuestra área de estudio.

d) Proveedores: Dichos camiones tienen prioridad en la entrada sobre los camiones de carga.

PROCESOS

En proceso por el que pasan todas las unidades es el siguiente: Una vez documentado el camión entra a la fila de la báscula que es la única para todos, tardan un tiempo aproximado de 2 minutos por camión en ser registrados: éste tiempo puede variar dependiendo de otros factores.

Los camiones de materia prima pasan a la planta donde la descargan, posteriormente, no vuelven a pasar por la báscula si llegaran a entrar otra vez, a excepción de los proveedores de combustóleo, los cuales afectan la fila de la báscula por lo que contribuyen a aumentar el tiempo de espera de los camiones que van a las envasadoras o regresan de ellas.

Los camiones express, a diferencia de los normales, evitan la espera para pasar a la báscula; pero dentro de la planta se les proporciona un servicio normal.

En el estudio de tiempos y movimien-

tos se consideraron datos históricos de dos meses (septiembre y octubre de 1994), las muestras fueron obtenidas durante las visitas efectuadas a la planta. Los tiempos se tomaron en fracciones de hora.

Para el análisis de los datos de los tiempos de interarribo tanto de camiones normales como express, tiempo de interarribo de camiones de materia prima, número de costales cargados por camión, tiempos de servicio en báscula, y tiempo de estancia de camiones de materia prima, se utilizó el paquete estadístico SAS. Se realizaron pruebas de bondad de ajuste para las variables mencionadas anteriormente, los resultados arrojados por este análisis fueron los siguientes:

a) Tiempo de interarribo de camiones normales: Con una media de 14.07 min. y una distribución Weibull con ($\Theta=0$, $\text{Shape}=0.68$, $\text{Scale}=0.11$)

b) Tiempo de interarribo de camiones express: Se presentó una distribución empírica debido a que la media fluctuaba entre dos horas diferentes del día.

c) Tiempo de interarribo de proveedores: Se registró de 9.91 min. con una distribución Weibull con una $\Theta=0.01$, $\text{Shape}=0.64$, $\text{Scale}=0.11$.

d) Tiempo de envasado: Es de 0.033 min. por costal.

e) Cantidad de costales cargados: Con un valor de la moda de 800 costales.

6 APLICACION DE LA SIMULACION AL CASO "TOLTECA":

En este trabajo la simulación fue aplicada a través de la serie de pasos que la componen. Nuestro primer paso consistió en definir adecuadamente nuestro problema, poniendo énfasis en los objetivos y alcances del trabajo, los cuales fueron: desarrollar un modelo de simulación que fuera capaz de reflejar posibles cambios en los procedimientos y métodos de operación de la planta e identificar los factores que redundaran en mejoras de los tiempos de estadía. Ya identificados los problemas principales que producían las largas estadías de los camiones cargeros tales como los horarios fijos de comida, descomposturas de las envasadoras y el control de tráfico interno, se procedió al diseño de experimentos para realizar un análisis estadístico sobre las interacciones entre factores y de los diferentes niveles dentro de un mismo factor.

Nuestro análisis consistió de un diseño 2X2X2, donde se manejó la siguiente nomenclatura:

- CONTROL DE TRAFICO con(A), sin(a)
- ROTACION DE COMIDAS con(B), sin(b)
- FUNCIONAMIENTO CORRECTO con(C), sin(c)

De este diseño resultaron por lo tanto

8 posibles combinaciones de los niveles de los factores, con cada una de estas combinaciones se realizaron 10 réplicas de la simulación (las 10 réplicas arrojaron una potencia de la prueba mayor de 0.8, por lo cual se consideró un tamaño de muestra confiable). Los 80 resultados arrojados por la simulación fueron introducidos para su análisis en el paquete estadístico SAS.

Para cada una de las ocho diferentes situaciones se realizó un programa de simulación en el lenguaje GPSSH, los cuales se corrieron sobre una microcomputadora.

7 CONCLUSIONES

1) La simulación aportó una forma diferente de llevar a cabo un experimento real.

2) La filosofía de liderazgo de calidad permitió atacar el problema de la empresa de una manera metódica y eficiente.

3) Los resultados de la conjunción de la Calidad Total, Simulación y Diseño de Experimentos permitieron visualizar diferentes soluciones al mismo problema.

Referencias

- [COP 93] Hoshyar, A. and Victor Nuila: Required Steps For Successful Design and Implementation of Simulation. Computers and Operations Research., 25: 53-59 (1993).

- [Law 91] Law, A.M., and W. D. Kelton:
Simulation Modeling & Analysis, 2d
ed., McGraw-Hill International Edi-
tions, (1991).
- [Hov 89] Hoover S. V., and R. F. Perry:
Simulation: A Problem-Solving Ap-
proach, Addison-Wesley Publishing
Company, (1989).
- [Kl 92] Kleijnen, J., and W. V. Groenen-
daal: Simulation: A statistical Per-
spective, John Wiley & Sons, (1992).
- [Mac 91] McHaney, R.: Computer Simu-
lation: A Practical Perspective, Aca-
demic Press, Inc., (1991).

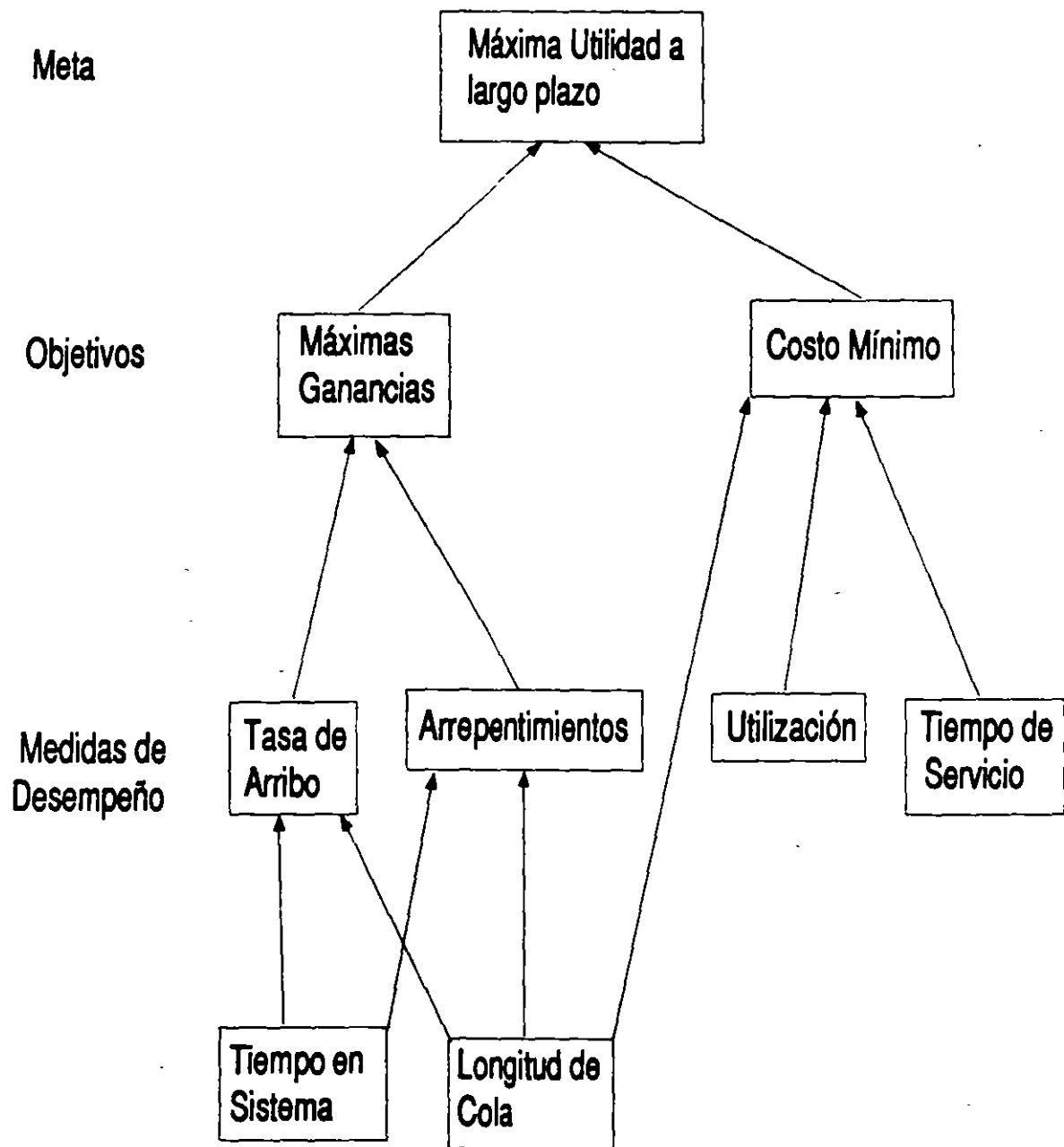


Figura 1 .Las Medidas de desempeño indican cuando los objetivos y metas del sistema son alcanzados.

4.6

VENTAJAS

DE LA

ANIMACION EN

LA OBTENCION

DE

CREDIBILIDAD

PROOF ANIMATION

I.- GENERALIDADES

Proof Animation es una aplicación de tipo general cuya finalidad es desarrollar y presentar en forma animada cualquier sistema que esté siendo simulado. Las principales características de Proof son:

- Es una animación fuera de línea que requiere de un proceso de simulación previo. No está conectada a ningún lenguaje de simulación o herramienta mientras está ejecutando la animación.
- Soporta una resolución de 640 x 350 pixels o bien monitores EGA y VGA.
- Permite la animación de sistemas más grandes que el tamaño de una pantalla de monitor. Cualquier parte del sistema puede ser vista a cualquier escala sin perder resolución usando la función *Zoom*.
- Soporta dibujos en dos dimensiones y para lograr un efecto de pseudo-3D, Proof soporta un punto de vista isométrico.
- Permite la captura de dibujos realizados previamente en sistemas CAD.
- Provee una velocidad de tiempo de animación ajustable.
- Permite crear presentaciones a cualquier nivel de sofisticación, incluyendo páginas estáticas mezcladas con segmentos de animación.
- La versión estudiantil tiene tres limitaciones intencionales: (a) El archivo de movimientos termina después de leer 1250 líneas, (b) El tamaño máximo de layout que puede ser guardado es de 12500 bytes y (c) La ejecución soporta solamente 2 minutos de tiempo de ejecución a tiempo real.

Archivos de entrada y salida utilizados en Proof.

Cuatro tipos de archivos son utilizados para la entrada de datos en una animación:

Archivo Layout

Este archivo contiene todas las gráficas y textos que sirven como el fondo o escenario de una animación. También contiene la definición de la forma de los objetos (Object Classes), los campos de texto variables (Messages), los niveles variables (Bars) y las trayectorias de movimiento guiado (Paths).

Los archivos layout normalmente se construyen, editan y salvan dentro del Proof Animation, sin embargo pueden derivarse de datos provenientes de otros programas basados en dibujos de CAD.

El formato de este archivo es ASCII e intenta ser simple de observar y decifrar por parte del usuario.

Archivo de movimientos

Este archivo es el que maneja la animación. Contiene la secuencia de eventos en el tiempo. Este archivo es generalmente escrito directamente por un modelo o programa que ha sido diseñado para generar sintácticamente los comandos del Proof.

Archivo de presentación

Cónsiste en una secuencia de comandos que muestran hojas y animaciones. Este archivo es creado a mano utilizando un editor de textos.

Archivos de imágenes

Estos archivos contienen imágenes fijas utilizables solamente en el módulo de Presentación. La fuente de estos archivos puede ser el Proof Animation <Alt-G> o cualquier otro software que

soporte el formato de archivos .PCX, por ejemplo el Paintbrush de Windows.

Cuatro tipos de archivos son utilizados para la salida de datos:

Archivo Layout

Este archivo es tanto una salida como una entrada de datos. El archivo layout es típicamente creado siguiendo una sesión de dibujo. El archivo contiene información de la geometría del escenario.

Archivo Linkage

Desde el Draw Mode o del Path Mode el Proof genera un archivo de encadenamiento cuya función principal es ayudar a encadenar la Animación con la Simulación. Este archivo contiene la descripción de todos los caminos que se han definido. El modelo de simulación puede extraer información importante de las distancias a partir de este archivo.

Archivo Log

De vez en cuando el Proof muestra "Señales de Alerta", avisos, errores, etc. en la pantalla. Esta información junto con una descripción más detallada es automáticamente archivadas en el archivo "archmovim.log".

Archivos de imágenes

Es posible en cualquier momento salvar como una imagen el layout actual para usarse posteriormente como un archivo de entrada. Para hacer eso, teclear <Alt-G>.

Interfase con el usuario.

La interfase con el usuario del Proof Animation es a base de Modos y Menús. El sistema tiene siete Modos de operación:

Run Mode

Es el módulo utilizado para ejecutar las animaciones. Las funciones principales son: arranque, paro, control de velocidad y control del tiempo.

Debug Mode

Es similar a Run Mode pero agregando algunas

herramientas para llevar la animación paso por paso y para analizar el movimiento de los objetos.

Draw Mode

Este módulo es utilizado con el fin de dibujar y editar los layouts a utilizar en la animación.

Path Mode

Este módulo permite al usuario definir trayectorias o caminos por los cuales los objetos se pueden mover. Las trayectorias son estructuras lógicas formadas por la unión de líneas y arcos.

Class Mode

Es el módulo en el cual se define la forma o dibujos de los objetos que se moverán a través del layout.

Presentation Mode

Es el módulo a utilizar para controlar el libreto de una presentación. Una presentación consiste en una o más secuencias de segmentos de animación y páginas estáticas, separadas por efectos especiales.

Setup Mode

El módulo permite alterar la configuración de los datos del sistema tales como velocidad del ratón o la paleta de colores.

II.- ARCHIVO LAYOUT

Este archivo contiene toda la información acerca del escenario, texto y gráficas de una animación, siendo todos ellos elementos inmóviles dentro de la animación. Además el archivo contiene la definición de las propiedades de los objetos, los mensajes y las trayectorias. El archivo layout generalmente se construye desde el Proof Animation y el formato se basa en código ASCII. La extensión de este archivo es .LAY.

Para crear un layout, entrar al sistema tecleando desde el prompt del DOS: SPA, PA, PP (dependiendo del tipo y versión del sistema), ya dentro del sistema el Proof Animation consta de 3 opciones en el menú:

FILE

Esta opción realiza el manejo de archivos y permite:

Open Layout & Trace	Abrir un layout y cargar un archivo de movimientos previamente creados, para usar esta opción es necesario que tanto el archivo LAY como el ATF tengan el mismo nombre.
Open Layout Only	Seleccionar un archivo con el layout LAY.
Open Trace Only	Seleccionar un archivo de movimientos ATF.
Open Presentation	Seleccionar un archivo de presentación PSF.
Show Slide	Seleccionar un archivo de imágenes PCX.
Save Layout	Guardar un layout.
Write Linkage	Crear un archivo de datos de las distancias de las trayectorias.
Change Directory	Cambiar de directorio.
Change Disk	Cambiar de disco o drive.

VIEW

La opción VIEW permite manejar diferentes vistas de una animación, las opciones existentes son:

Select View	Seleccionar una vista previamente guardada que se desea cargar en pantalla.
Define View	Definir la imagen actual en pantalla como una nueva vista. Esta opción permite actualizar una vista, borrar una vista o salvar una vista. El usuario nombra cada una de las vistas que desee, teniendo como opción el actualizar cualquier vista como

(Home), esta vista es la principal, de tal manera que cada vez que se llame a un layout, en la pantalla aparecerá esta vista.

Pan	Mover la imagen hacia arriba, abajo, derecha, izquierda en porcentajes de 25%.
Zoom	Acercar o alejar la imagen actual en porcentajes que van desde 0.3 hasta 3.3 de la imagen actual.
Zoom Box	Acercar la imagen actual de acuerdo a las dimensiones de un recuadro, el recuadro se puede cerrar o abrir al gusto y al oprimir de nuevo VIEW la imagen que se encuentre dentro del recuadro será ampliada.
Zoom-to-Fit	Modificar la imagen actual para que todos los elementos del layout quepan en una sola pantalla.
Out + Back	Mover la imagen actual en pantalla en función de la definición de un nuevo centro utilizando el ratón para fijarlo.
Rotate	Girar la imagen actual cierto número de grados o convertir la imagen a coordenadas Isométricas o bien Ortogonales.
Grid	Definir o eliminar la malla de coordenadas como ayuda para dibujar.
Refresh	Redibujar la pantalla.

MODE

La opción MODE permite el acceso a otros módulos para creación del layout, específicamente a:

Run	Permite el arranque y pausa en la
-----	-----------------------------------

	ejecución de una animación, controlar la velocidad de animación y saltar en el tiempo.
Debug	Permite ejecutar una animación paso por paso.
Draw	Permite dibujar cualquier geometría correspondiente a objetos inmóviles, el escenario, textos, mensajes y barras. Para dibujar utiliza los primitivos de líneas, polilíneas, arcos, esquinas y llenado.
Class	Permite dibujar la geometría de los objetos móviles usando los mismos primitivos que en DRAW, así como dar de alta las propiedades de: velocidad, volumen, color y direccionalidad.
Path	Permite definir las trayectorias a ser usadas por los objetos durante la animación. Al definir una trayectoria es necesario fijar principalmente: <ul style="list-style-type: none"> - Los segmentos que forman su longitud total. - El tiempo o la velocidad en cruzarla. - Si es circular. - Si es acumulativa.
Present	Ejecutar una presentación o mostrar una imagen.
Setup	Fijar los parámetros de: Color, Aspecto, Pantalla, Tolerancias, Frecuencia y Velocidad del Ratón.
Exit	Salir de Proof Animation.

III.- ARCHIVO DE MOVIMIENTOS

Este archivo es el generador de la animación ya que contiene toda la secuencia de eventos a través del tiempo. Este archivo es generalmente creado a partir de un modelo o programa que haya sido diseñado para ese fin. El archivo es de tipo texto y la extensión de dicho archivo es ATF.

Estructura

La estructura de un archivo ATF es equivalente a un libreto, en donde es necesario colocar los eventos que deseamos que ocurran a través del tiempo. De esta manera tenemos el archivo con el siguiente formato:

Tiempo t1

-

-

Lista de eventos que ocurren en t1

-

-

Tiempo t2

-

-

Lista de eventos que ocurren en $t2 > t1$

-

-

Tiempo t3

-

-

Lista de eventos que ocurren en $t3 > t2$

-

-

Tiempo t4

...

Tiempo tn

Fin

A continuación se presenta la lista de instrucciones de animación de eventos:

CREATE

El comando para crear un objeto individual es CREATE. La sintaxis es:

CREATE *c n*

Crear el objeto *n* perteneciente a la clase *c*

CREATE mono 10

CREATE maq 4

CREATE pieza 1

TIME

El archivo de movimientos siempre debe contener comandos que muevan el reloj de la animación. De otra manera, nada pasará en la pantalla. El comando TIME es el más utilizado para mover el reloj hacia adelante. La sintaxis es:

TIME *t*

Fijar el tiempo *t* en el que ocurrirán los eventos. El valor *t* debe ser positivo y siempre mayor o igual a valor definido anteriormente. Se aceptan valores enteros o de punto flotante.

TIME 10
TIME 234.7
TIME 0

PLACE .. AT

Esta instrucción causa que un objeto activo o existente aparezca en cierto punto en el espacio dentro del sistema de coordenadas del Proof. La sintaxis es:

PLACE *n* AT *x y*

Colocar el objeto *n* en la posición *x,y* de la pantalla

PLACE 1 AT 20 20
PLACE 10 AT 40 30
PLACE 4 AT 10 10

Un objeto se mantendrá en reposo inmediatamente después de la instrucción. El objeto puede no ser visible si las coordenadas *x*, *y* están fuera de la pantalla, si el color del objeto es BACKDROP o si coinciden un número impar de objetos idénticos en el mismo punto.

PLACE .. IN

Este comando coloca un objeto en el mismo lugar que otro objeto. La sintaxis es:

PLACE *n1* IN *n2*

Colocar el objeto *n1* sobre el objeto *n2*.

PLACE 1 IN 2
PLACE 4 IN 10

La instrucción coloca el objeto *n1* de tal forma que coincida su punto (0,0) con el punto (0,0) del objeto *n2*. La instrucción deja al objeto después de su ejecución en estado de reposo.

Subsecuentes movimientos, relocalización, destrucción, rotación del objeto n2 no tienen efecto sobre n1.

PLACE .. ON

La instrucción coloca un objeto sobre una trayectoria. El objeto comenzará a moverse inmediatamente si puede hacerlo. La sintaxis de la instrucción es:

PLACE *n* ON *p* AT *x*

Colocar el objeto *n* sobre la trayectoria *p* a partir de *x*; el objeto se moverá sobre esa trayectoria de acuerdo a la velocidad del objeto o de la trayectoria.

PLACE 100 ON camino
PLACE 14 ON cam1
PLACE 50 ON cam4 AT 10
PLACE 80 ON tray AT END

A menos que se defina el AT, el objeto será colocado al inicio del primer segmento de la trayectoria seleccionada. En caso de definir al AT, el objeto será colocado a *x* distancia del inicio del primer segmento de la trayectoria. Si la trayectoria es de tipo acumulativa y uno o más objetos ya se encuentran acumulados al final de la misma la instrucción PLACE .. ON .. AT END causará un aviso de alerta de choque (Encroachment Alert).

MOVE

La ejecución de esta instrucción causa que un objeto se mueva en línea recta desde su posición actual a la posición especificada como destino. La sintaxis del MOVE es:

MOVE *n* t *x* *y*

Mover el objeto *n* en línea recta en cierto tiempo *t* desde su posición actual hasta la posición final definida por *x,y*.

MOVE 1 40 30 30
MOVE 1 30 20 10

La duración *t* determina que tan rápido se moverá el objeto. Hay que notar que un objeto que ha sido creado pero que no ha sido

colocado en cierta posición no puede ser movido con la instrucción MOVE; entonces, antes del MOVE debe existir al menos una instrucción PLACE.

ROTATE

Esta instrucción gira a un punto con referencia a su punto (0,0). Esta instrucción tiene dos formas. La primera causa que un objeto gire un ángulo especificado. La sintaxis es:

ROTATE *n* [TO] *a* [TIME *h*] [SPEED *s*] [STEP *p*]

Girar el objeto *n* de su posición actual al ángulo *a* en cierto tiempo *h* o a cierta velocidad *s*, la rotación se realiza en incrementos de tamaño *p*.

ROTATE 44 TO 45 TIME 10 STEP 5
ROTATE 60 TO 90
ROTATE 2 TO 180 SPEED 180 STEP 1

La segunda causa que un objeto rote indefinidamente sin parar. La sintaxis de esta segunda forma es:

ROTATE *n* SPEED *s* [STEP *p*]

Girar el objeto *n* indefinidamente desde su posición actual a cierta velocidad *h*, la rotación se realiza en incrementos de tamaño *p*.

ROTATE 20 SPEED -40
ROTATE 10 SPEED 10 STEP 1

SET OBJECT .. COLOR

Inmediatamente cambia el color del objeto. El color inicial es determinado por el color con el que fue creado el objeto. La manipulación del color puede realizarse después de haber sido creado el objeto individual. La sintaxis es:

SET [OBJECT] *n* COLOR *cl*

Cambiar al color *cl* el objeto *n*; los colores disponibles son: white, blue, red, yellow, green, tan, backdrop (negro), layout (gris).

SET 10 COLOR blue

SET 15 COLOR backdrop
SET 100 COLOR 5

La identificación de los colores puede hacerse con números del 0 al 8, de acuerdo con la tabla inferior:

Color	Número
BACKDROP	0
BLUE	1
RED	2
WHITE	3
YELLOW	4
PINK	5
TAN	6
GREEN	7
LAYOUT	8

SET OBJECT .. CLASS

Esta instrucción cambia la forma de un objeto durante la animación. También cambia las propiedades del objeto tales como color, volumen, direccionalidad, etc. Lógicamente SET OBJECT x CLASS es lo mismo que DESTROY x seguido de un CREATE x, excepto por que la posición y el estado del movimiento del objeto se mantienen. La sintaxis de la instrucción es:

SET [OBJECT] *n* CLASS *c*

Cambiar las propiedades del objeto *n* mediante el cambio a la clase *c*.

SET 18 CLASS pieza2
SET 19 CLASS producto

SET OBJECT .. SPEED

La instrucción controla la velocidad de un objeto moviéndose sobre una trayectoria. Debido a que el movimiento del objeto puede estar influenciado por SET PATH .. SPEED, la palabra OBJECT en este comando no se recomienda omitir en la sintaxis. Como regla general si a un objeto no se le define una velocidad, tomará la velocidad de la trayectoria. La sintaxis es:

SET [OBJECT] *n* SPEED [*v* | PATH | CLASS]

Fijar la velocidad fijada por el parámetro *v* o por el camino *PATH* o de acuerdo a la clase *CLASS* a la que pertenece el objeto *n*.

```
SET 4 SPEED 100
SET 14 SPEED PATH
SET 13 SPEED CLASS
```

SET OBJECT .. TRAVEL

Esta instrucción tiene el mismo efecto que **SET OBJECT .. SPEED** cuando se utiliza con una velocidad numérica. La velocidad del objeto se calcula dividiendo la distancia aún por recorrer de cierta trayectoria por el tiempo definido en la instrucción. La sintaxis es:

```
SET [ OBJECT ] n TRAVEL t
```

Fijar el tiempo de movimiento *t* del objeto *n*.

```
SET 13 TRAVEL 30
```

Esto causa que Proof divida la longitud aún por recorrer de la trayectoria sobre la que se encuentre el objeto entre 30 dando como resultado la velocidad que tomará el objeto.

SET PATH .. SPEED

Esta instrucción cambia la propiedad de velocidad de una trayectoria. El cambio afecta a todos los objetos que se estén moviendo sobre la trayectoria. La sintaxis es:

```
SET PATH p SPEED s
```

Definir para la trayectoria *p* una velocidad de *s*.

```
SET 10 SPEED PATH
SET 100 SPEED CLASS
```

SET OBJECT .. DIRECTIONAL

SET OBJECT .. NONDIRECTIONAL

La direccionalidad de un objeto puede ser cambiada utilizando la instrucción cuya sintaxis se muestra a continuación:

SET [OBJECT] *n* [DIRECTIONAL | NONDIRECTIONAL]

Cambiar el estado de la direccionalidad del objeto *n*.

SET 50 DIR
SET 100 NODIR

SET OBJECT .. CLEARANCE

La instrucción cambia el claro libre de un objeto. Esta instrucción se utiliza si los objetos necesitan acumularse con diferentes claros dependiendo de su orientación rotacional. La sintaxis es:

SET [OBJECT] *n* CLEARANCE *at ad*

Cambiar el claro libre atrás *at* y adelante *ad* del objeto *n*.

SET 1 CLEARANCE 2 2

El nuevo claro de la pieza 1 será dos unidades lineales hacia adelante y dos hacia atrás.

SET BAR

Esta instrucción controla el tamaño y color actuales de un elemento barra individual. En el control del tamaño, las cuatro esquinas son controlables. La sintaxis es:

SET BAR *b* [COLOR *cl*] [[*ed1*] |1] [[*ed2*] |2] [[*ed3*] |3] [[*ed4*] |4]

Cambiar la dimensión *edj* de la barra *b* hacia la dirección *lj* fijando el color *cl*.

SET BAR util TOP 10
SET BAR prod BOTTOM 20
SET BAR area TOP 5 RIGHT 5
SET BAR uno TOP 2 BOTTOM 2 LEFT 2 RIGHT 2
SET BAR dos COLOR 1 TOP 5
SET BAR tres COLOR blue

El valor por default es TOP, entonces se puede teclear para el primer ejemplo:

SET BAR util 10

DESTROY

La instrucción sirve para destruir un objeto que no sea ya necesario en la animación y permite liberar la memoria que está consumiendo y también desaparece de la pantalla. La sintaxis es:

DESTROY n

Destruir el objeto n y desaparecerlo de la pantalla.

DESTROY 33
DESTROY 100

Una vez que un objeto es destruido, el número de su identificación puede ser utilizado nuevamente.

DT

La instrucción DT (Delta Time) incrementa el valor del reloj de animación una cantidad especificada. Es similar al TIME, excepto que el TIME utiliza al valor actual del reloj y DT utiliza la diferencia entre el tiempo actual y el deseado. La sintaxis es:

DT t

Incrementar el tiempo t unidades a partir del tiempo actual.

DT 5
DT 10

Equivale a:

TIME 5
TIME 15

La mayoría de las animaciones utilizan la instrucción TIME. DT es útil para crear composiciones musicales con la instrucción SOUND.

SET VIEW

Esta instrucción permite cambiar la vista del layout durante la ejecución de la animación. La sintaxis es:

SET VIEW nombre

Cambiar a la vista nombre previamente definida en el layout. Algunos ejemplos son:

```
SET VIEW " (Home) "  
SET VIEW " (Class) "  
SET VIEW embarque
```

WRITE

Esta instrucción modifica el contenido de un mensaje (Message), los cuales son caracteres de texto dinámicos que aparecen y desaparecen del pantalla. La sintaxis es:

```
WRITE m t
```

Escribir en pantalla el texto t en la posición donde se encuentra el mensaje m.

```
WRITE mss 100  
WRITE dato 0.567  
WRITE estado descompuesta
```

Es permitido utilizar espacios en blanco dentro del mensaje y termina donde se encuentre un salto de línea.

SOUND

Esta instrucción agrega la dimensión del sonido a la animación utilizando el generador de tonos que forma parte de cualquier PC. Un tono permanecerá audible hasta que otro tono sea activado o hasta que se produzca un SOUND 0. La sintaxis es:

```
SOUND f
```

Iniciar un sonido con cierta frecuencia f.

```
TIME 5  
SOUND 400  
DT 5  
SOUND 500  
DT 10  
SOUND 0
```

END

La instrucción END termina todo el proceso de animación. Proof permanece activo pero no lee en el archivo de movimientos. La sintaxis es:

END

Fin de archivo

Ejemplo.

TIME 0	En el tiempo 0 ...
SOUND 400	Se obtiene un sonido con frecuencia de 400 hertz y
CREATE PIEZA 1	Se crea un objeto de clase PIEZA numerado como 1
PLACE 1 AT 30 30	Se coloca en la posición 30 30 de la pantalla.
TIME 10	En el tiempo 10 ...
SET 1 COLOR BLUE	Se cambia el objeto 1 a color azul y ...
SOUND 0	Se apaga el sonido.
TIME 20	En el tiempo 20 ...
MOVE 1 40 50 50	Se mueve el objeto 1 a una velocidad de 40 hacia 50 50.
TIME 60	En el tiempo 60 ...
DESTROY 1	Se destruye el objeto 1.
TIME 65	En el tiempo 65 ...
END	Termina la animación.

IV.- ARCHIVO DE PRESENTACION

El archivo de presentación es un archivo texto, generalmente creado en forma manual en un editor, consiste en una secuencia de comandos que presentan imágenes estáticas y animaciones. Este archivo debe tener la extensión PSF.

Comandos:

SLIDE

Este comando permite mostrar imagen previamente creada con formato PCX. Las imágenes creadas deben realizarse con una resolución de 640 x 350 o 640 x 480 a 16 colores. La sintaxis es:

`SLIDE [DISSOLVE | SUNRISE] nombre de la imagen.PCX`

Ejemplos:

`SLIDE resultados.pcx`

`SLIDE DISSOLVE resultados.pcx`

Mostrar en pantalla la imagen "resultados" hasta que sea ejecutado otro comando de Proof.

RUN

Este comando permite ejecutar la animación de un layout con un archivo de movimientos. La sintaxis es:

`RUN [DISSOLVE | SUNRISE] archivo.LAY archivo.ATF
VIEW vista tiempo1 tiempo2`

Ejemplo.

`RUN mach.LAY rep1.ATF VIEW superior 100 200`

Ejecutar mach.LAY con rep1.ATF utilizando la vista superior del tiempo 100 al tiempo 200

WAIT

Se utiliza este comando para retrasar o incluir una pausa dentro de la presentación. Generalmente WAIT se utiliza inmediatamente después de un SLIDE. La sintaxis es:

`WAIT [tiempo de retraso] [aviso]`

Si se omite el tiempo de retraso, se hará una pausa y permitirá al usuario moverse a la velocidad deseada. En el

caso de usar el campo aviso, en pantalla aparecerá un mensaje en color rojo mostrando el texto.

Ejemplos:

WAIT 10

WAIT

WAIT 5 Este mensaje se autodestruirá en 5 segundos.

MENU

Un libreto puede o no utilizar el comando MENU. En caso de no usarlo, la presentación se realizará secuencialmente de acuerdo a lo escrito en el libreto. En caso de usar el comando MENU se debe acompañar de uno o más comandos CASE. Cada CASE a su vez pueden formar un carrusel que incluya el resto de los comandos para formar una subpresentación. Al final de un MENU debe existir un ENDMENU. La sintaxis es:

MENU { aviso }

Donde lo expresado en aviso aparecerá en pantalla en color rojo

Ejemplo:

MENU Selecciona una opción.

CASE

En presentaciones con menú, el comando CASE aparece en el libreto al inicio de cada carrusel (un carrusel es una subpresentación independiente accesible desde una lista de menú). Un carrusel termina en el siguiente CASE o con el comando ENDMENU.

Cada comando CASE debe aparecer después de un comando MENU y antes de un comando ENDMENU. Es un error incluir CASE sin tener comando MENU. La sintaxis del CASE es:

CASE *título*

Donde *título* es el nombre del carrusel en la lista del menú. Puede contener hasta 40 caracteres incluyendo espacios en blanco.

CASE Introduccion

CASE 35 llegadas/hora

ENDMENU

Las presentaciones con MENU deben tener un comando ENDMENU. Este comando es utilizado para finalizar el carrusel que sigue al comando MENU. La sintaxis es:

ENDMENU

WIPE

Utilizada para la transición de una imagen a un color sólido. Este color puede ser cualquiera de los definidos en el Proof. La sintaxis es:

WIPE color { FADE | DISSOLVE | VENETIAN | CURTAIN
{UP, DOWN, UPDOWN } | RANDOM }

Este comando permite diferentes efectos, por ejemplo al usar VENETIAN se observa una transición entre imágenes con un efecto de cierre de persianas venecianas. Si se utiliza DISSOLVE la transición tiene un efecto de formación de puntos o motas. La opción CURTAIN realiza la transición como una cortina que baja, sube o se cierra en ambos sentidos. La opción RANDOM ocasiona que el Proof seleccione aleatoriamente cualquiera de las opciones anteriores.

Ejemplos:

WIPE 2 CURTAIN UP

WIPE GREEN VENETIAN

WIPE WHITE RANDOM

END

El procesamiento de una presentación finaliza cuando se encuentra el comando END. La sintaxis es:

END [siguiente archivo PSF]

En una presentación sin menú, END sin "siguiente archivo PSF" regresa al usuario al DOS. END con " siguiente archivo PSF" causa que Proof se encadene con otra presentación.

END otra.PSF

END cambio

EXIT

El comando EXIT puede aparecer en cualquier parte del libreto, la sintaxis es:

EXIT [siguiente archivo PSF]

Un EXIT sin definición posterior regresa al usuario al DOS. Un EXIT con definición posterior causa un encadenamiento con otro archivo PSF. En presentaciones con MENU debe existir la posibilidad de salir del menú vía el comando EXIT.

Ejemplo.

```
SLIDE INTROD.PCX
WAIT 20
WIPE RED VENETIAN
RUN CEMEX.LAY CEMEX.ATF
WIPE GREEN CURTAIN DOWN
SLIDE RESULT.PCX
WAIT
END
```

Ejemplo.

```
SLIDE INTROD.PCX
WAIT 20
MENU Selecciona una opción
CASE Cemex
RUN CEMEX.LAY CEMEX.ATF
WIPE RED VENTIAN
SLIDE RES.PCX
WAIT 50
CASE Caterpillar
RUN CATER.LAY CATER.ATF VIEW UNO 0 500
WIPE WHITE RANDOM
SLIDE RES2.PCX
WAIT 50
CASE Salir a DOS
EXIT
ENDMENU
END
```

BIBLIOGRAFIA

Using Proof Animation. Wolverine Software Corporation.
Annandale, VA USA. 1994



FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA

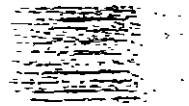
CURSOS INSTITUCIONALES

INSTITUTO MEXICANO DE SISTEMAS
E INVESTIGACION DE OPERACIONES, A.C.

TOPICOS DE ACTUALIDAD DE LA INVESTIGACION
DE OPERACIONES Y LA SIMULACION

MATERIAL COMPLEMENTARIO

Del 23 al 25 de septiembre de 1996



PALACIO DE MINERIA
1996

Orden de Producción

N° orden de producción:		Código de pedido:	
Fecha:	secuencia	operación	terminado
Producto:		control de calidad 1	
		control de calidad 2	
		corte de chasis	
		corte de la cabina	
		corte de la caja	
Tamaño del lote		montaje de la cabina	
		montaje de la caja	
		montaje luces traseras	
		montaje luces delante	
		mon. luces de sirena	
		montar ruedas	
		montaje camioneta	
		pegatina "Just-in time"	

Solicitud de materias primas

N° de solicitud:		Código del pedido:	
material	cantidad	material	cantidad
luz frontal		plancha base 4 cm.	
luz trasera		plancha base 2 cm	
luz de sirena		accesorio camioneta	
ejes de ruedas		pagatina "Just-in-Time"	
ruedas		cinta de montaje	

Orden de Producción

N° orden de producción:		Código de pedido	
Fecha:	secuencia	operación	terminado
Producto		control de calidad 1	
		control de calidad 2	
		corte de chasis	
		corte de la cabina	
		corte de la caja	
Tamaño del lote		montaje de la cabina	
		montaje de la caja	
		montaje luces traseras	
		montaje luces delante	
		mon. luces de sirena	
		montar ruedas	
		montaje camioneta	
		pegatina "Just-in time"	

Solicitud de materias primas

N° de solicitud:		Código del pedido:	
material	cantidad	material	cantidad
luz frontal		plancha base 4 cm.	
luz trasera		plancha base 2 cm	
luz de sirena		accesorio camioneta	
ejes de ruedas		pagatina "Just-in-Time"	
ruedas		cinta de montaje	

Orden de Producción

N° orden de producción		Código de pedido:	
Fecha:	secuencia	operación	terminado
Producto:		control de calidad 1	
		control de calidad 2	
		corte de chasis	
		corte de la cabina	
		corte de la caja	
Tamaño del lote		montaje de la cabina	
		montaje de la caja	
		montaje luces traseras	
		montaje luces delante	
		mon. luces de sirena	
		montar ruedas	
		montaje camioneta	
		pegatina "Just-in time"	

Solicitud de materias primas

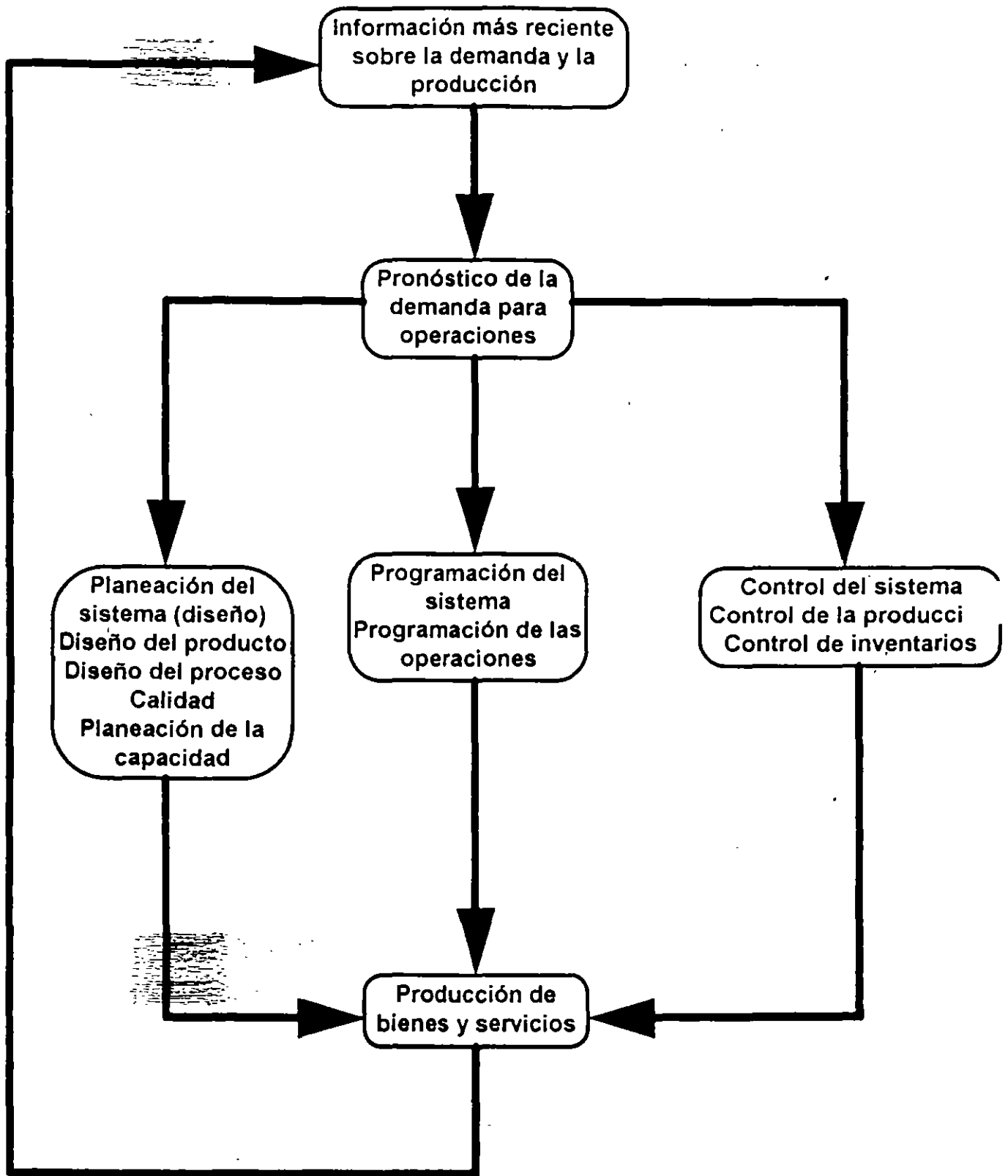
N° de solicitud		Código del pedido	
material	cantidad	material	cantidad
luz frontal		plancha base 4 cm	
luz trasera		plancha base 2 cm	
luz de sirena		accesorio camioneta	
ejes de ruedas		pagatina "Just-in-Time"	
ruedas		cinta de montaje	

Orden de Producción

N° orden de producción		Código de pedido:	
Fecha	secuencia	operación	terminado
Producto		control de calidad 1	
		control de calidad 2	
		corte de chasis	
		corte de la cabina	
		corte de la caja	
Tamaño del lote		montaje de la cabina	
		montaje de la caja	
		montaje luces traseras	
		montaje luces delante	
		mon. luces de sirena	
		montar ruedas	
		montaje camioneta	
		pegatina "Just-in time"	

Solicitud de materias primas

N° de solicitud		Código del pedido:	
material	cantidad	material	cantidad
luz frontal		plancha base 4 cm	
luz trasera		plancha base 2 cm	
luz de sirena		accesorio camioneta	
ejes de ruedas		pagatina "Just-in-Time"	
ruedas		cinta de montaje	



PLANEACIÓN DE REQUERIMIENTO DE MATERIALES (MRP)

- PLANEACIÓN PARA ANTICIPAR PROBLEMAS
- PUEDE SER MUY FLEXIBLE, PUES ESTÁ FUNDAMENTADO EN:
 - ◆ PRONOSTICO DE LA DEMANDA
 - ◆ TIEMPOS PREDETERMINADOS DE ENTREGA Y PRODUCCIÓN
- SINTETIZA ADMINISTRACIÓN DE INVENTARIO Y PLANEACIÓN DE PRODUCCIÓN
- PUEDE SER UN ELEMENTO INTEGRADOR MUY IMPORTANTE, EL ENCADENAR LA DEMANDA FINAL CON LOS REQUERIMIENTOS DE MATERIALES
- DEBE COMBINARSE CON BUSCAR AHORROS AL MOMENTO DE PROGRAMAR COMPRAS, BALANCEANDO COSTOS DE INVENTARIOS
- IDEALMENTE, CON PRONOSTICOS Y TIEMPOS EXACTOS, ELIMINARÍA INVENTARIOS

SISTEMAS JUSTO A TIEMPO (JIT)

- UNA FILOSOFÍA DE MANUFACTURA BASADA EN LA ELIMINACIÓN PLANEADA DE CUALQUIER DESPERDICIO Y LA MEJORA CONTINUA DE LA PRODUCTIVIDAD. ABARCA TODAS LAS ACTIVIDADES DE MANUFACTURA., DESDE EL DISEÑO HASTA LA ENTREGA Y ENFATIZA CERO INVENTARIOS, CERO DEFECTOS Y REDUCCIÓN DE TIEMPOS DE ENTREGA
- TODO DESPERDICIO DEBE ELIMINARSE
- EL INVENTARIO ES UN DESPERDICIO
- FLEXIBILIDAD EN MANUFACTURA INDISPENSABLE
- RESPETO Y APOYO DE TODAS LAS AREAS
- INTEGRACIÓN CON PROVEEDORES
- PARTICIPACIÓN DE TODO EL PERSONAL
- EJECUCIÓN PARA IDENTIFICAR PROBLEMAS

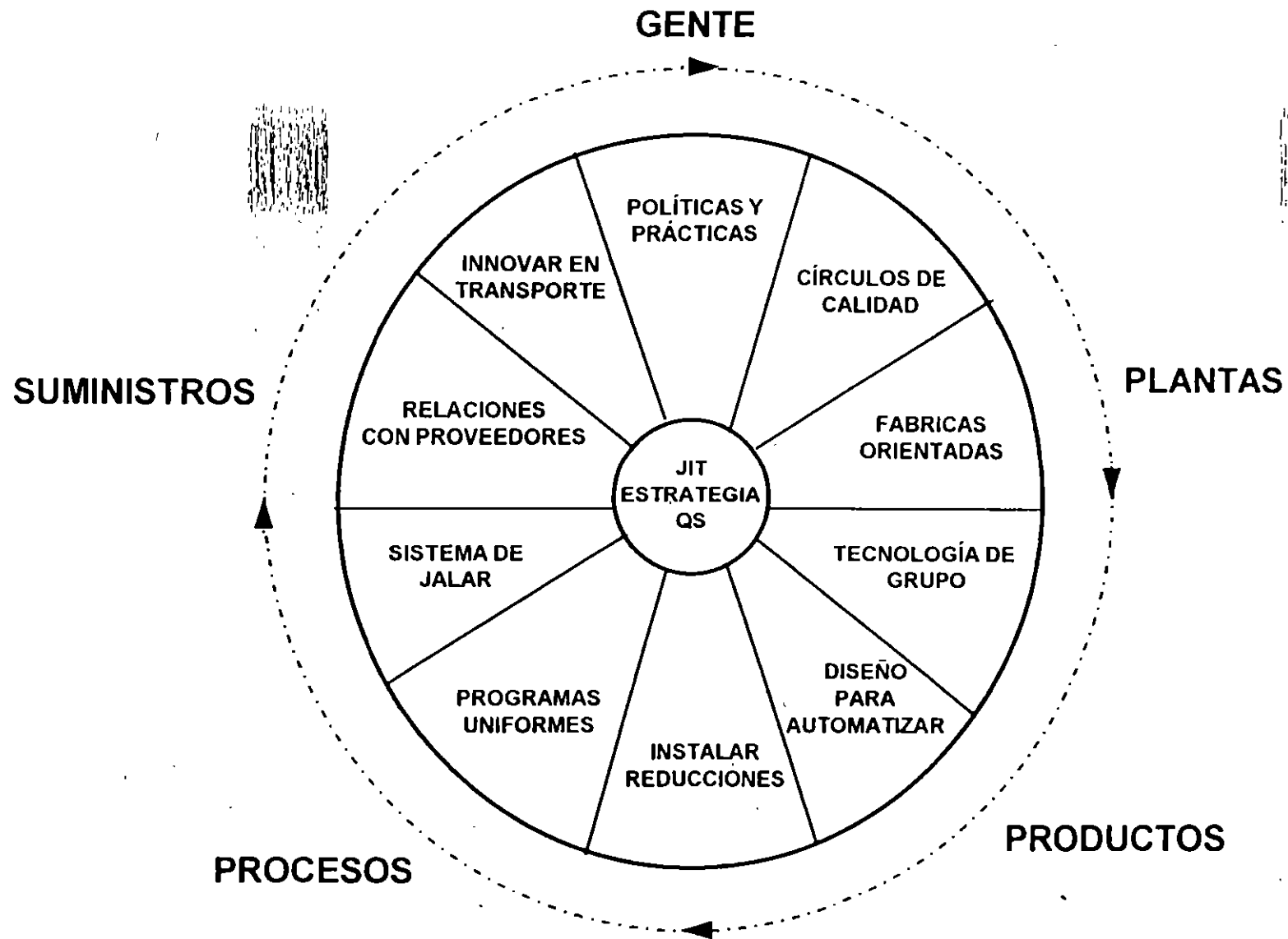
JIT

También el JIT puede ser entendido a través de un conjunto de cambios que propone y que conllevan a una alteración radical de la manera en que trabaja una empresa, estas modificaciones implican acciones encaminadas a resolver problemas, considerados estos como las causas reales de ineficacia industrial, que se traducen como fuentes potenciales de desperdicios.

Estos cambios se denominarán como "cambios del JIT" y se enlistan a continuación:

- Mejor suministro.
- Cambio en la configuración de la planta.
- La reducción de los tiempos de montaje.
- El sistema de "jalar" la producción, denominado Kanban.
- El mantenimiento total.

Para lograr sus propósitos, el JIT exige calidad, y particularmente aquella ~~que~~ consiste en hacer las cosas bien la primera vez, conocida como Calidad en el Origen.



KANBAN

- HERRAMIENTA DE INFORMACIÓN PARA PRODUCCIÓN Y TRANSPORTE (MOVIMIENTO) BASADA EN IDENTIFICADORES (KANBANES)
 - ◆ KANBAN DE TRANSPORTE: ESPECIFICA CLASE Y CANTIDAD DE PRODUCTO QUE HA SIDO RETIRADA POR EL PROCESO SUBSECUENTE DEL PROCESO PRECEDENTE
 - ◆ KANBAN DE PRODUCCIÓN: ESPECIFICA CLASE Y CANTIDAD DE PRODUCTO QUE EL PROCESO PRECEDENTE DEBE PRODUCIR
- NO ENVIAR PRODUCTOS DEFECTUOSOS
- RETIRAR SOLO LO QUE SE NECESITA, Y SOLO ACOMPAÑADO DEL KANBAN
- PRODUCIR SOLO LA CANTIDAD Y SECUENCIA REQUERIDA POR LA SIGUIENTE ESTACIÓN
- EQUILIBRAR LA PRODUCCIÓN (MAYOR CAPACIDAD AL INICIO)
- KANBAN ES AJUSTE FINO
- ESTABILIZAR Y RACIONALIZAR TODA LA CADENA

KANBAN

En el JIT se propone una forma de operación que resulta ser totalmente contraria, la idea fue tomada de la manera como se desempeñan los supermercados americanos y consta de dos etapas.

La primera etapa del proceso inicia cuando un cliente compra artículos, la información de la cantidad sustraída es controlada por una caja registradora.

En la segunda etapa del Kanban, el departamento de compras del supermercado utiliza los datos de la caja registradora para reponer las cantidades y las diferentes variedades de artículos vendidos. Estos artículos de reposición se encuentran en los almacenes.

CÁLCULO DEL NÚMERO DE TARJETAS KANBAN A EMITIR

Una de las fórmulas empleadas para calcular el número de tarjetas Kanban entre dos operaciones sucesivas conectadas es la siguiente:

$$D = \frac{\text{Demanda promedio durante el tiempo principal + las existencias de seguridad}}{\text{No. de piezas transferidas en cada contenedor}}$$

$$D = \frac{D(TE+TP)(1+\alpha)}{C}$$

Donde:

N = Número de tarjetas.

D = Nivel de producción diario (demanda diaria, en unidades).

TE = Tiempo de espera para el contenedor antes de empezar su procesamiento (decimales de día).

TP = Tiempo de procesamiento por contenedor (decimales de día).

~~C~~ = Número de piezas transferidas en cada contenedor (no más de 10% de la demanda diaria)

α = Coeficiente de seguridad. Variable política de no más de 10%, que refleja la eficiencia de las estaciones de trabajo.

11-11-11

11-11-11



FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA

CURSOS ABIERTOS

*TOPICOS DE ACTUALIDAD DE LA
INVESTIGACION DE OPERACIONES Y LA SIMULACION*

23, 24 y 25 de Septiembre de 1996

*Expositor: Dr. Hamdy A. Taha.
Palacio de Minería
México, D.F.*

11-11-11

11-11-11



**EL ESTADO ACTUAL DE LA
INVESTIGACION DE OPERACIONES**

**Dr. Hamdy A. Taha
Departamento de Ingeniería Industrial
Universidad de Arkansas, EEUU**



Septiembre de 1996

¿Qué es la Investigación de Operaciones?

La aplicación en grupos interdisciplinarios del método científico a problemas relacionados con el control de las sistemas organizacionales, a fin de que produzcan soluciones que mejor sirvan a los objetivos de toda la organización.

HERRAMIENTAS DE LA I.O.

Optimización:

Max ó min (f.o.)
s.a.
restricciones

- .Prog. Lineal
- .Prog. No lineal
- .Prog. Entera
- .Prog. de Redes
- .Prog. Estocástica
- .Prog. Geométrica
- .Prog. Dinámica
- .Prog. de Metas
- .Teoría de decisiones y Juegos

Análisis:

Evaluación de las medidas de desempeño

- .Teoría de colas
- .Teoría de estadística
- .Cadenas de Markov
- .Simulación

Aplicaciones:

- .Inventarios
- .Lineas de Espera
- .Producción
- .Transporte
- .Manufactura
- .Planeación Estratégica
- .Servicios Públicos
- .Balance de Lineas

EL AVANCE DE LA PROGRAMACION LINEAL

- .La PL es la técnica más importante de I.O.
- .La PL es el indicador principal del progreso que hemos logrado en I.O.
- .La historia de todas las técnicas de I.O. es en realidad la historia de la PL.

Fechas Importantes en la Historia de la PL y I.O.:

1820: Gauss descubrió el método de resolver ecuaciones lineales. Al principio del siglo XVI, se presentó la solución de la ecuación

$$ax - b = 0$$

como un problema formidable.

Ejemplo: $5x - 12 = 0$

$x_1 = 5, 5x - 12 = 13$

$x_2 = 4, 5x - 12 = 8$

Entonces,

$$x = \frac{4 \times 13 - 5 \times 8}{13 - 8} = \frac{12}{5}$$

1947: George Dantzig desarrolló la técnica de programación lineal y el método SIMPLEX.

1945-1950: El desarrollo de la computadora.

En 1935, el economista norteamericano Leontiev desarrolló un modelo de 44 ecuaciones y 44 variables para representar los sectores industriales de la economía norteamericana. Resultó imposible resolver las ecuaciones y tuvo que reducir el modelo a solo 10 ecuaciones por 10 variables.

En 1948, se necesitó gastar más de 120 horas para resolver un problema de programación lineal de tamaño muy chico.

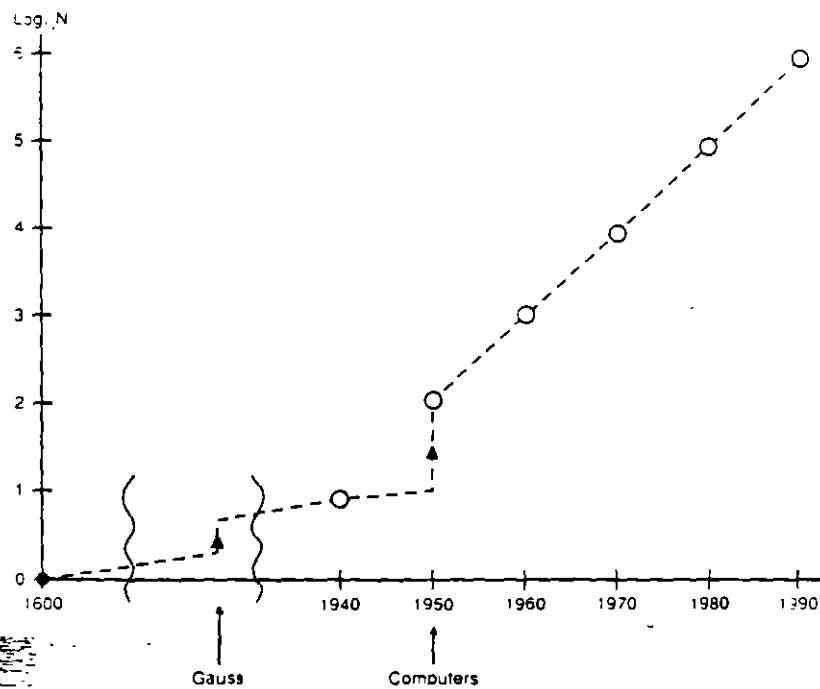
1951: Khun y Tucker desarrollaron las condiciones básicas para optimizar los problemas no lineales.

1958: Gomory desarrolló las teorías que tratan de programación entera.

1984: Karmarkar desarrolló un método nuevo para resolver programación lineal.

EL TAMAÑO DEL MODELO EN LA PROGRAMACION LINEAL

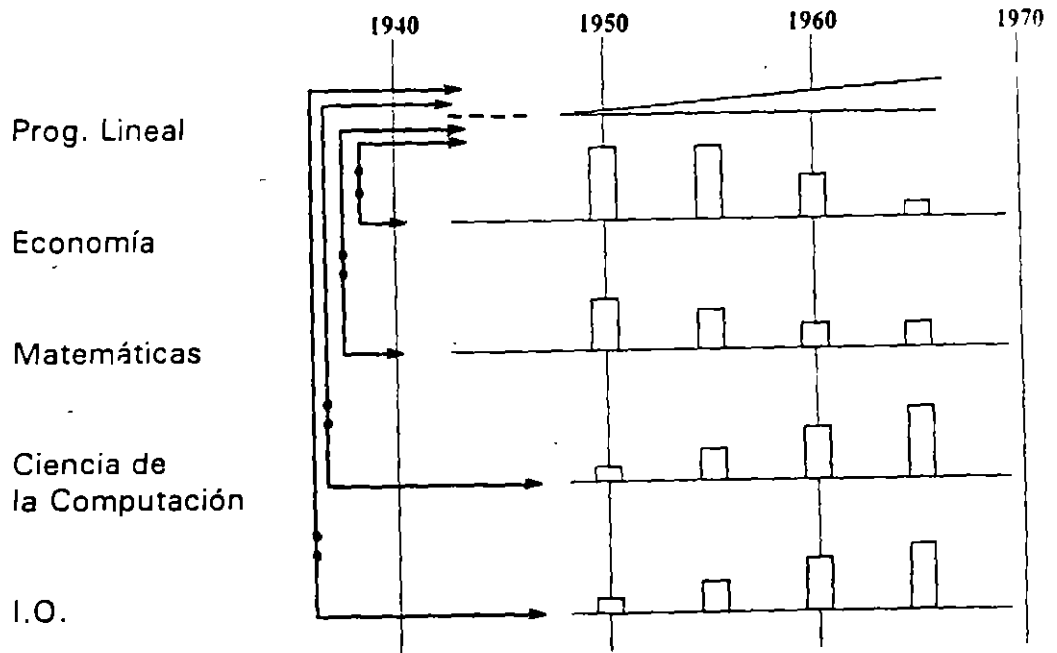
1950: 200 restricciones
1960: 1,000 restricciones
1970: 10,000 restricciones
1980: 100,000 restricciones
1990: Hasta 1,000,000 de restricciones



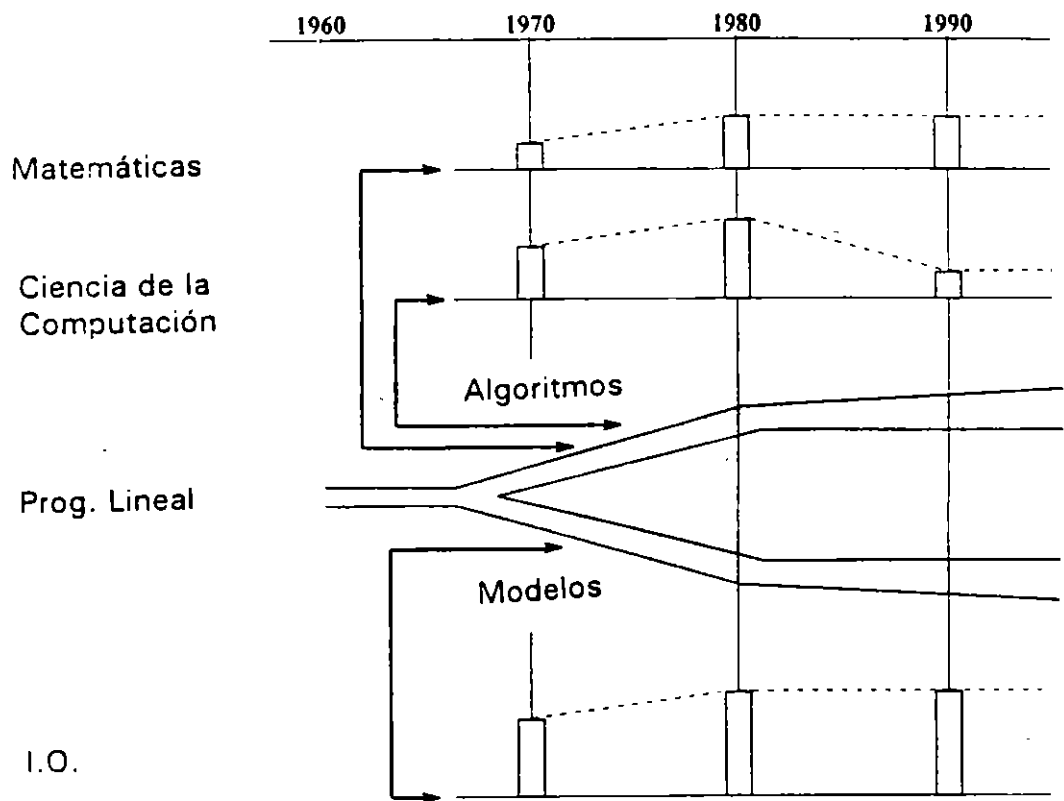
GANADORES DEL PREMIO NOBEL CUYOS TRABAJOS INCLUYEN PROGRAMACION LINEAL

Year	Nobel Laureate	Year	Title of a publication in the LP area
1969	Ragnar Frisch	1954	"Principles of linear programming with particular reference to the double gradient form of the logarithmic potential network"
1970	Paul Samuelson	1958	<i>Linear Programming and Economic Analysis</i> (Dorfman, Samuelson, Solow)
1972	Kenneth Arrow	1958	<i>Studies in Linear and Nonlinear Programming</i> (Arrow, Hurwicz, Uzawa)
1973	Wassily Leontiev	1951	<i>The Structure of the American Economy, 1919-1939</i>
1975	Leonid Kantorovich	1939	<i>Mathematical Methods for Organizing and Planning Production</i>
1975	Tjalling Koopmans, ed	1951	<i>Activity Analysis of Production and Allocation</i>
1978	Herbert Simon	1956	"Derivation of a linear decision rule for production and employment" (Holt, Modigliani, Muth, Simon)
1982	George Stigler	1945	"The cost of subsistence"
1985	Franco Modigliani	1956	"Derivation of a linear decision rule for production and employment" (Holt, Modigliani, Muth, Simon)
1987	Robert Solow	1958	<i>Linear Programming and Economic Analysis</i> (Dorfman, Samuelson, Solow)
1990	Harry Markowitz	1957	"The elimination form of the inverse and its application in linear programming"
1990	Merton Miller	1959	"An application of linear programming to financial budgeting and the costing of funds" (Charnes, Cooper, Miller)
1990	William Sharpe	1967	"A linear programming algorithm for mutual fund portfolio selection"

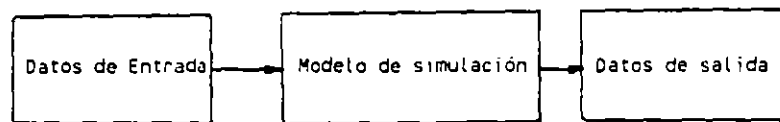
EL DESARROLLO DE LA PROGRAMACION LINEAL EN VARIOS CAMPOS DE LA CIENCIA



EL ESTADO ACTUAL DE LA PROGRAMACION LINEAL



LOS ELEMENTOS DE UN ESTUDIO DE LA SIMULACION

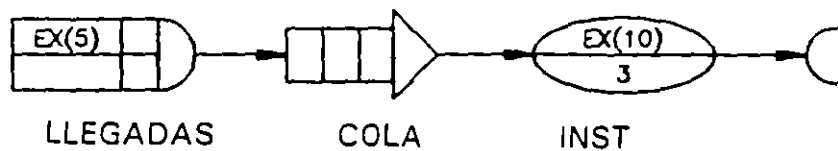


Puntos Importantes a Recordar:

- La implantación de la simulación en la práctica llega más allá del desarrollo del modelo utilizando un lenguaje de simulación.
- La tarea principal del analista de simulación consiste en preparar los datos de entrada, recolectar las observaciones estadísticas, y analizar los datos de salida.
- La técnica de simulación consiste en un experimento estadístico muy particular. La implantación formal de la simulación requiere experiencia adecuada en las técnicas de estadística.

El Impacto de los Datos de Entrada Sobre los Resultados del Modelo

Modelo de una Línea de Espera con un Servidor



Tiempo entre llegadas = $EX(5)$

Se determina el tiempo de servicio utilizando datos históricos.

Resultados basados en 100 replicaciones

Distribución del tiempo de servicio	Promedio del tiempo de demora en COLA	Promedio del tamaño de COLA
exponencial	6.71	6.78
gamma	4.54	4.60
Weibull	4.36	4.41
lognormal	7.19	7.30
normal	6.04	6.13

Desarrollo del Modelo

Los sistemas en donde se utiliza la simulación son del tipo de líneas de espera:

- .Una fuente para crear clientes
- .Una cola donde los clientes pueden esperar
- .Una instalación donde se ofrece el servicio

Tipos de los Lenguajes de la Simulación

.Todos los lenguajes son basados en el aspecto de eventos discretos.

.Nodos o bloques básicos

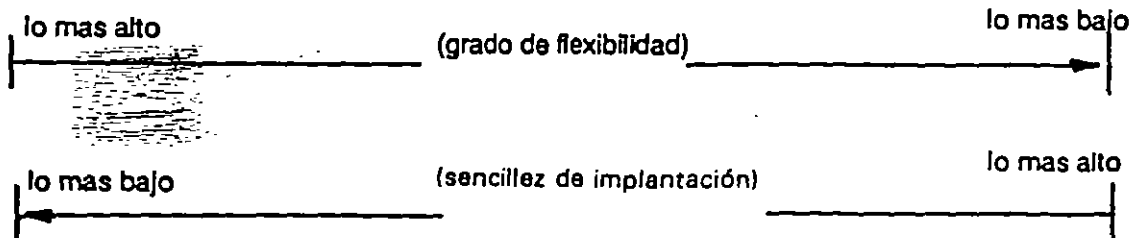
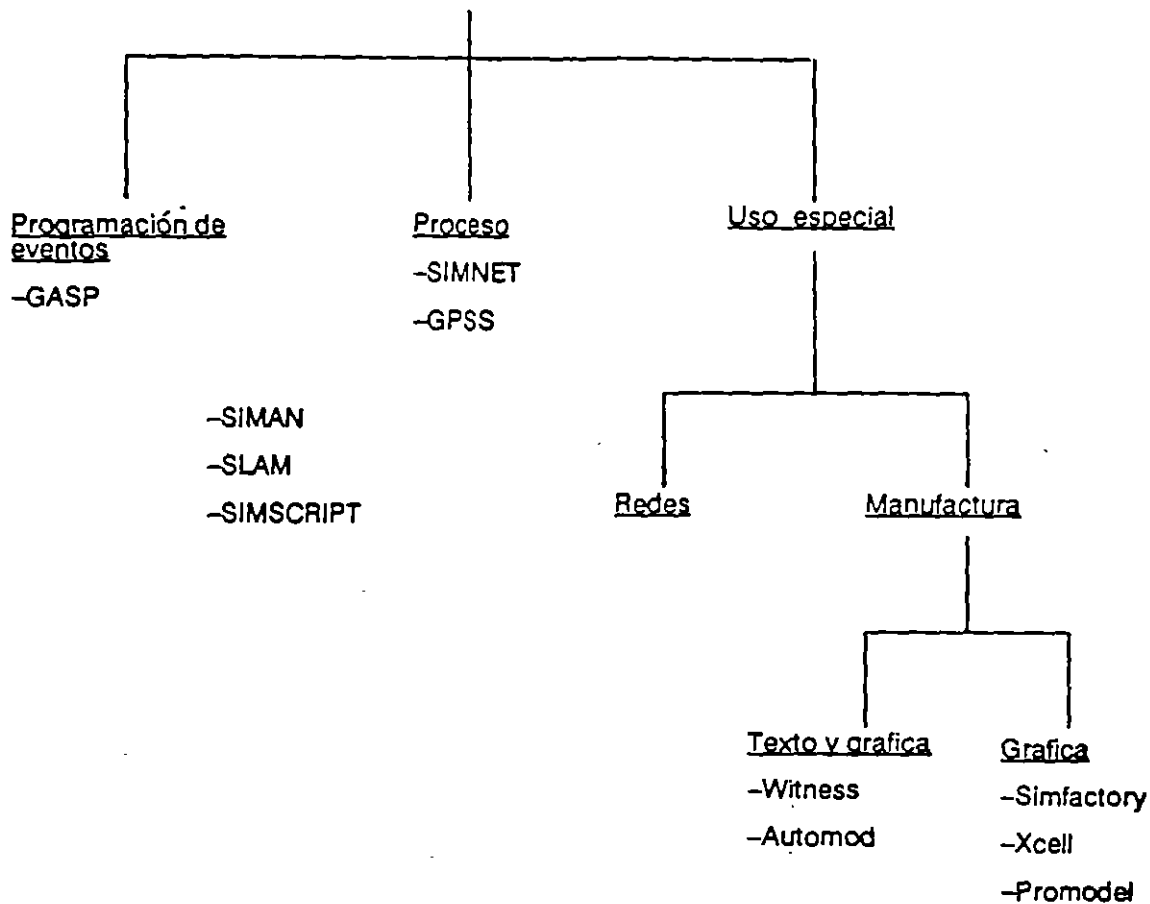
- Una fuente
- Una cola
- Una instalación

.Nodos o bloques especiales

.Control los movimientos de la transacciones

.Llevar a cabo las funciones de manufactura o de manejo de materiales

Los Lenguajes de la Simulación



DISTRIBUCION DE LA UTILIZACION DE LOS LENGUAJES DE LA SIMULACION

FORTRAN, C, BASIC	42%
Lenguajes de simulación	43%
Lenguajes especializados	15%

ANÁLISIS DE LOS DATOS DE SALIDA

● **Es indispensable tratar la simulación como un experimento estadístico.**

- Hacer varias corridas con sucesiones distintas de números aleatorios nos da distintos resultados de salida.

● **El experimento de simulación es de tipo particular porque no satisface ningunas de las condiciones del experimento estadístico común.**

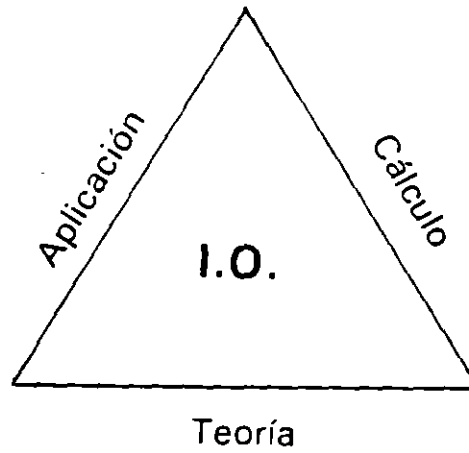
- Las observaciones no son independientes.

- Las distribuciones de las variables de simulación no son estacionarias.

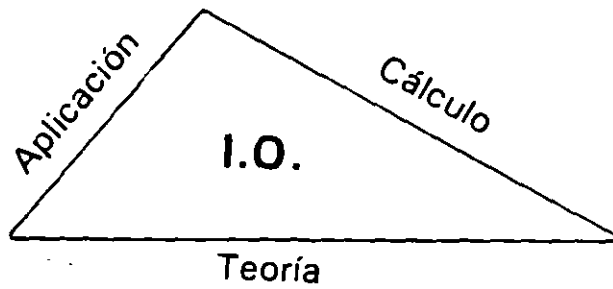
- Las observaciones no siguen la distribución normal.

LOS ELEMENTOS BASICOS DE LA INVESTIGACION DE OPERACIONES

1. El Estado Ideal:



2. El Estado Actual:



¿DONDE ESTAMOS?

. Los algoritmos y la teoría de I.O. han alcanzado un nivel de madurez muy respetable con la excepción de las áreas de

- Programación entera
- Programación no lineal

. El avance enorme de las computadoras y las técnicas de computación nos brinda ventajas en la solución de problemas.

. Por otro lado, abundan programas de computación en el mercado que tratan de convencernos de que podamos resolver problemas de I.O. sin mucho conocimiento de la teoría de las técnicas que estamos utilizando.

. Nos quedamos atrasados en el área de la implantación de I.O. en la práctica.

. En las universidades y los institutos académicos, la mayoría de los investigadores tratan de problemas de tipo teórico que quizás no tiene nada que ver con lo que pasa en la vida real. La lista abajo nos da prueba al respecto:

Revistas Científicas de I.O.

Teoría:

1. *Management Science*
2. *Operations Research*
3. *Mathematical Programming*
4. *Decision Science*
5. *Omega*
6. *Naval Research Logistics*
7. *IIE Transactions*
8. *European Journal of OR*
9. *Mathematics of OR*

Aplicación:

1. *Interfaces*

DILEMA EN LA IMPLANTACION DE LA I.O.

● Los académicos tienen base fuerte en la teoría de la I.O. pero les falta implantarla en la práctica.

● Las personas que trabajan en la industria y el sector público se les presenta la situación contraria: Tienen la experiencia práctica, pero les falta el entrenamiento adecuado en las herramientas de la I.O.

Sugerencias:

1. Formar equipos de I.O dentro de las compañías y darles el apoyo que se necesita para funcionar bien como una entidad integrada dentro de la organización.
2. La academia puede ofrecer sus experiencias a los equipos de la I.O. en forma de consultorías.
3. La academia debe involucrarse a los alumnos en resolver problemas en la industria y el sector público.
4. Los académicos deben involucrarse más en la investigación de técnicas con fin práctico.
5. Para mejor difundir las experiencias de aquellas personas que están involucradas en resolver problemas prácticos, es importante aumentar el número de revistas científicas en que se puede publicar estas experiencias.

¿A DONDE VAMOS?

1. Hay que reconocer que existe una diferencia entre enseñar modelos y enseñar el enfoque para la formulación de modelos.
2. La mayoría de los libros de la I.O. se concentran en el área de enseñar modelos.
3. Es difícil escribir un libro que presenta adecuadamente los puntos finos del arte de la formulación de modelos.
4. Es sumamente importante que el alumno aprenda adecuadamente la teoría de la I.O. Este tipo de conocimiento se puede adquirir solo en las instituciones académicas.
5. Al mismo tiempo, en la clase no tenemos suficiente tiempo para la enseñanza de modelos y el arte de su formulación.

UN PLAN ACADÉMICO PARA OFRECER UN PROGRAMA DE LA I.O.

Licenciatura:

Introducción de las
técnicas de la I.O.
(2 cursos)

Maestría y
Doctorado:

Teoría

Cálculos

Aplicación

HEURÍSTICOS PARA ENSEÑAR LA FORMULACION DE MODELOS

Heurístico °1: Determine los módulos básicos del problema (Divide y vencerás).

Ejemplo: El análisis económico de dos alternativas para suministrar agua potable a Arabia Saudita:

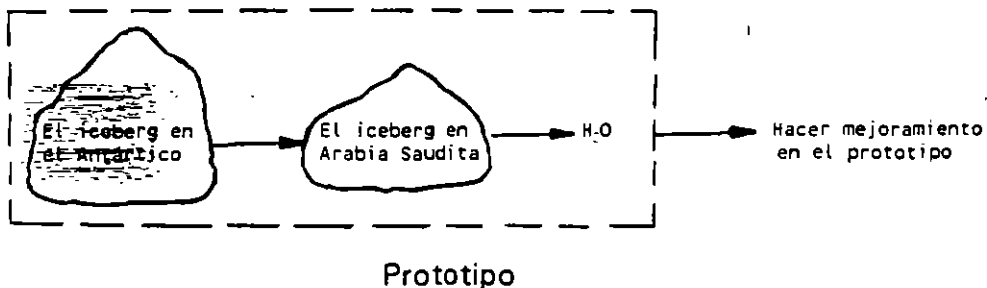
1. Desalinización del agua del Mar Rojo.
2. Transporte de iceberg desde el Antártico hasta Arabia Saudita.

Módulos del problema:

1. Forma y tamaño del iceberg en el Antártico.
2. Tamaño del iceberg al llegar a Arabia Saudita.
3. Transformamiento del iceberg a agua potable.

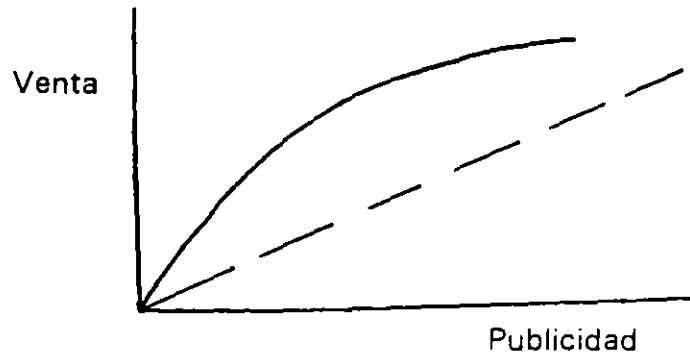
Heurístico °2: Desarrolle un prototipo que abarca todos los módulos.

1. Se utiliza el prototipo para contestar preguntas sobre el problema original.
2. Se utiliza el análisis de sensibilidad para llevar a cabo la evaluación de cada módulo.
3. El análisis del prototipo nos da indicio de los módulos en que tenemos que concentrarnos con el fin de realizar los mejores resultados.



HEURÍSTICOS PARA ENSEÑAR LA FORMULACION DE MODELOS

Heurístico °3: Utilice gráficas y dibujos.



Heurístico °4: Utilice parámetros cuando sea posible para representar las relaciones entre las variables del problema.

Tamaño neto del iceberg = $f(p_1, p_2, \dots, p_n)$ x Tamaño original

o

Tamaño neto del iceberg = r x Tamaño original

Heurístico °5: Distinga entre la generación de ideas agresivas y la evaluación de las alternativas de decisiones.

-El problema del elevador

Heurístico °6: Aproxime los datos cuando sea necesario.

-Sobran los datos en las situaciones sencillas, y faltan en las situaciones importantes.

-El uso de la distribución triangular en simulación.

LOS REQUISITOS PARA LA IMPLANTACION I.O. EN LA PRACTICA

1. 30 años de experiencia en la solución de problemas prácticos.
2. Un doctorado en la I.O. para que se entienda todas las teorías de las herramientas y técnicas del campo.
3. Una habilidad excepcional de ser creativo y original en apreciar y comprender el aspecto práctico del problema.

O SIEN

Todo lo anterior se puede sustituir por ser un individuo con
sentido comun.