



# Modelos de optimización y lenguajes algebraicos de modelado

Andrés Ramos

# Modelo

---

Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja (por ejemplo, la evolución económica de un país), que se elabora para facilitar su comprensión y el estudio de su comportamiento.

Diccionario de la lengua española. Real Academia Española.

# Modelo

---

- Representación precisa de una realidad
- Herramienta de ayuda a la toma de decisiones
- Puede involucrar equipo multidisciplinar
- *Modelador*: especifica y desarrolla el modelo
- *Experto*: conoce el problema real

# Modelado

---

- Ciencia
  - Análisis y detección de relaciones entre datos
  - Suposiciones y aproximaciones a los problemas
  - Algoritmos específicos de solución
- Arte
  - Visión o interpretación de la realidad
  - Estilo en modelo y documentación
  - Elegancia y simplicidad en desarrollo
  - Uso creativo de herramientas

# Beneficios del modelado

---

- Diálogo entre modelador y experto
- Organiza información disponible
- Estructura el pensamiento sobre problema
- Internaliza estructura organizativa de empresa
- Compartir supuestos entre modelador y experto
- Indicar dirección de mejora en decisiones

# Etapas en el desarrollo de un modelo

---

- Identificación del problema
- Especificación matemática y formulación
- Resolución
- Verificación, validación y refinamiento
- Interpretación y análisis de resultados
- Uso extensivo

# Identificación del problema

---

- Recolección de información relevante
- Definición del problema en términos vagos
- Interpretación y traducción a términos precisos
- Datos son vitales, suelen ser cuello de botella
- Etapa fundamental para que decisiones sean útiles

# Especificación matemática y formulación

- Definición de variables, ecuaciones, función objetivo, parámetros
- Análisis de tamaño y estructura del problema
- Identificación de tipo de problema (LP, MIP, NLP)
- Énfasis en precisión y belleza en la formulación



# Tipos de problemas LP según su tamaño

	Restricciones	Variables
• Caso ejemplo	100	100
• Tamaño medio	10000	10000
• Gran tamaño	100000	100000
• Muy gran tamaño	> 100000	> 100000

# Resolución

---

- Algoritmo de obtención de solución óptima, cuasióptima, satisfactoria
- Diferentes métodos de solución
- Diferentes implantaciones del algoritmo elegido



# Verificación, validación y refinamiento

- Eliminación de errores en codificación
- Comprobar validez de simplificaciones adoptadas
- Comprobación de adaptación a la realidad
- Ampliación en el modelado por nuevas necesidades

# Interpretación y análisis de resultados

---

- Análisis de sensibilidad en parámetros de entrada
- Robustez de la solución óptima
- Detección de soluciones cuasióptimas atractivas



# Uso, documentación y mantenimiento

- Etapa fundamental para el éxito de un modelo
- Documentación clara, precisa y completa
- Manual de usuario con especificación técnica funcional, matemática e informática
- Formación de posibles usuarios

# Alternativas desarrollo modelos optimiz

- Lenguajes de programación de propósito general (C, FORTRAN 90)
- Lenguajes o entornos de cálculo numérico o simbólico (hojas de cálculo, Matlab, Mathematica)
- Lenguajes algebraicos de modelado (GAMS, AMPL, AIMMS, XPRESS-MP)

# Ventajas lenguajes algebraicos

---

- Lenguajes de alto nivel para formulación compacta de modelos grandes y complejos
- Facilitan desarrollo de prototipos
- Mejorar productividad de modeladores
- Estructuran buenos hábitos de modelado
- Separan datos de estructura de optimizadores
- Formulación independiente del tamaño
- Modelo independiente de optimizadores

# Ventajas lenguajes algebraicos (cont.)

- Facilitan reformulación continua
- Documentación simultánea al modelo
- Permiten implantación de algoritmos avanzados
- Implantación fácil de problemas NLP, MIP, MCP
- Portabilidad entre plataformas y sistemas operativos



# Desventajas lenguajes algebraicos

- No son adecuados para usos esporádicos con problemas de pequeño tamaño
- No son adecuados para resolución directa problemas de tamaño gigantesco (1.000.000 x 1.000.000)



# Tendencias futuras

---

- Interfaz visual en formulación
- Interfaz más estrecha con hojas de cálculo y bases de datos
- Interfaz con funciones externas escritas en lenguajes de propósito general
- Resolución directa de problemas optimización estocástica (OSL)
- Selección automática de método y optimizador

# Aplicaciones reales

---

- En IIT se pasó de utilizar FORTRAN a utilizar GAMS exclusivamente
- Problemas de hasta 117.000 restricciones, 225.000 variables y 655.000 elementos no nulos resueltos con facilidad
- Incorporación de algoritmos avanzados (descomposición estocástica anidada de Benders) en modelos

# Programación

---

- Disciplina cuyo dominio es básico en muchos proyectos de ingeniería
  - Ciencia: pensamiento, disciplina, rigurosidad y experimentación
  - Arte: belleza y elegancia
- Aprendizaje mediante la lectura
- Un buen diseño es fundamental
- Programar por refinamiento gradual
- Utilizar una maqueta para el desarrollo, depuración y verificación de un modelo
- Cuidado con los detalles

# Estructura general de un modelo de optimización escrito en GAMS

---

- Declaración de sets y parámetros
- Variables
- Ecuaciones
- Modelo
- Inclusión y manipulación de datos de entrada
- Acotación e inicialización de variables,
- Resolución del problema
- Lectura y presentación de resultados

# Elementos de estilo de programación (calidad)

---

- Modularidad: datos, estructura modelo y resultados en diferentes ficheros
- Mismo uso en diferentes partes del código
- Comprobación del dominio de los índices



# Elementos de estilo de programación (mantenibilidad)

---

- Código limpio, cuidar la estética en la escritura
- Mantener la coherencia de las reglas de escritura (indentación en instrucciones repetitivas)
- Facilitar su lectura (paralelismo entre instrucciones consecutivas similares)
- Comentarios ilustrativos y bien localizados
- Nombres de entidades descriptivos
- Definiciones con dimensiones físicas
- Uso sistemático y coherente de mayúsculas y minúsculas

# Uso avanzado de GAMS

---

- minimización del tiempo de ejecución y/o de la memoria
- importante cuando se trata de problemas de muy gran tamaño ( $> 100.000 \times 100.000$ ) o resolución iterativa de numerosos problemas (más de 100)
- aparece al usar simulación de Monte Carlo o técnicas de descomposición



# Tiempo de ejecución de modelos escritos en GAMS

---

- tiempo de creación
  - formulación del problema específico
- tiempo de interfaz
  - comunicación entre lenguaje GAMS y optimizador
- tiempo de optimización
  - resolución del problema por el optimizador

# Análisis de consumos de tiempo/memoria

- dependiente del tamaño y estructura de la matriz de restricciones
- número de resoluciones (iteraciones)
- variación entre soluciones sucesivas de los parámetros estocásticos

# Direcciones de mejora

---

- informáticas (asociadas al lenguaje GAMS)
- matemáticas (reformulación del problema)
  
- afectan conjuntamente al tiempo de ejecución
- criterios dependientes del problema, indican direcciones a explorar

# Direcciones a explorar

---

- uso de disco virtual
- uso de últimas versiones
- cambio en instrucciones de asignación
- reformulación manual del problema
- modelado de conjunto ordenados SOSn
- selección de optimizador y algoritmo de optimización

## Direcciones a explorar (cont.)

---

- ajuste de parámetros del optimizador
- uso de bases previas
- mejoras en el barrido de las numerosas optimizaciones
- detección de infactibilidades
- análisis de sensibilidad

# Cambio en instrucciones de asignación

- orden de los índices consistente en todos los elementos del modelo
- orden de barrido en instrucciones reiterativas coherente
- hacer uso extensivo de condiciones de exclusión mediante el uso de conjuntos dinámicos

# Preprocesamiento

- reducción de los tamaños de dos problemas con la opción de preprocesamiento de CPLEX 6.0

	R	V	E	R	V	E
Sin presolve	19047	27262	81215	48971	63935	187059
Con presolve	15744	21982	51079	40794	56133	135361
Decremento	17,3%	19,4%	37,1%	16,7%	12,2%	27,6%

# Mejoras en la formulación

---

- cálculo analítico de número de restricciones y variables. Detección de la estructura de la matriz del problema
- no crear variables ni ecuaciones superfluas
- reducción de número de restricciones y/o elementos
- escalación alrededor de 1
- acotación de variables



# Selección de optimizador y método de optimización

---

- hasta 5.000 x 5.000 método simplex
- desde 5.000 x 5.000 hasta 100.000 x 100.000 método de punto interior
- más de 100.000 x 100.000 métodos de descomposición
- análisis de sensibilidad método simplex

# Comparación entre preprocesamientos

	R	V	E	R	V	E
Sin preproc	19047	27847	82295	49715	64679	189477
Preproc CPLEX	-14,8%	-19,3%	-36,2%	-17,9%	-13,2%	-28,6%
Preproc OSL	-4,9%	0,0%	-2,4%	-15,6%	0,0%	-9,1%



# Comparación entre optimizadores y método de optimización

	Tiempo p.u.			Iter		
<b>CPLEX 6.0</b>						
Punto interior	41.8	1.0	32	237.3	1.0	35
Simplex dual	99.8	1.4	12692	1812.6	6.6	48695
Simplex primal	156.2	3.7	21622	1217.5	5.1	50280
<b>MINOS 5.3</b>						
Simplex primal	1863.6	44.6	23927	---	---	---
<b>OSL 2.1</b>						
Punto interior	163.9	3.9	10798	774.4	3.3	19524
Simplex primal	530.9	12.7	12685	7426.6	31.3	62019

# Récords actuales en LP

---

- problema de 117.000 x 225.000 x 655.000 resuelto en 240 s
- problema de 20.000 x 25.000 x 80.000 resuelto numerosas veces en un tiempo medio de 3 s
- problema de 50.000 x 65.000 x 200.000 resuelto numerosas veces en un tiempo medio de 8 s

