

Control 3 - Diseño de Sistemas de Transporte

Semestre Primavera 2023

P1 *Toretto's (Global) Auto Parts*

Dom Toretto ha decidido expandir su operación de "Toretto's Auto Parts" a nivel mundial para apoyar redes de carreras clandestinas y talleres de modificación de autos. Existen N ciudades clave en el mundo, cada una con una demanda proyectada de D_i piezas de alto rendimiento por año.

La Familia de Toretto ha identificado J ubicaciones estratégicas para establecer almacenes de distribución de partes de autos. El costo de asegurar un local en la ubicación j es de C_j dólares, y el costo de establecer un almacén de tamaño k —pequeño, mediano o grande— es de F_{jk} dólares. Un almacén de tamaño k tiene la capacidad de almacenar G_k piezas al año. Solo puede haber un almacén por ubicación.

Además, el costo de distribuir las partes a cada ciudad i desde un almacén en j es de T_{ij} dólares por pieza, y una ciudad puede ser atendida por cualquier número de almacenes.

Dom espera que lo ayudes a modelar los siguientes problemas de programación entera:

- (2 puntos) Minimizar el costo total para "Toretto's Auto Parts" de satisfacer toda la demanda mundial.
- (2 puntos) Maximizar la cantidad de piezas distribuidas, considerando un presupuesto total de B dólares.

Dom Toretto te considera parte de la familia y te ha confiado una tarea crucial: revisar los containers que transportan piezas de alto rendimiento para "Toretto's Global Auto Parts". Para él, cada pieza y cada miembro del equipo son igualmente importantes, y quiere asegurarse de que la logística de distribución refleje ese cuidado y compromiso.

- (2 puntos) Como miembro de la familia y parte del equipo de logística, tu misión es aplicar estrategias *greedy* para el **CRP (Container relocation problem)**.

Dom te pide que utilices la regla *Min-Max* para determinar qué container se debe mover a continuación. Además, te anima a emplear otro método de tu elección para que realices una comparación.

Aquí se presenta un ejemplo con 7 *stacks* y 3 *tiers*. Los números representan el día en que se debe retirar el container correspondiente.

s_0	s_1	s_2	s_3	s_4	s_5	s_6
				8		
6	5	7	13	11		9
10	2	1	3	12	4	14

Recuerda, en la familia Toretto, tu opinión y estrategias son valoradas. ¡Manos a la obra!

P2 *Relajaciones, Ramificaciones y Acotamiento*

- a) (1 punto) Demuestre que si RP es una relajación del problema de programación entera IP , entonces $z^R \geq z$.
- b) (1 punto) Demuestre que si una relajación RP es infactible, entonces el problema original IP también lo es.
- c) (1 punto) Sea x^* una solución óptima de RP . Demuestre que si $x^* \in X$ y $f(x^*) = c(x^*)$, entonces x^* es una solución óptima de IP .
- d) (3 puntos) Explique el algoritmo de Ramificación y Acotamiento, y sus condicionantes de poda.

P3 *TransLogix*

Usted es el nuevo empleado de TransLogix, empresa de transporte global que compite arduamente con los mejores del sector.

Actualmente TransLogix se encuentran con un desafío logístico. La empresa cuenta con un camión con una capacidad de carga limitada y múltiples paquetes, cada uno con su propio peso y valor asociado que refleja su importancia para la entrega. Su primera misión es determinar la forma más eficiente de llenar el camión para maximizar la utilidad total sin superar la capacidad de carga.

Tiene que tener en consideración que el camión tiene una restricción de peso total que puede transportar, limitada a 7 unidades. Además, se tienen cuatro paquetes disponibles para elegir, detallados a continuación:

- Paquete 1: Peso = 2, Valor = 10.
- Paquete 2: Peso = 3, Valor = 21.
- Paquete 3: Peso = 5, Valor = 50.
- Paquete 4: Peso = 6, Valor = 51.

Usted sabe que para abordar este desafío se deben realizar las siguientes acciones:

- a) (1.5 puntos) Modelar el desafío de TransLogix como un problema de programación entera.
- b) (1.5 puntos) Calcular el óptimo de la relajación del problema de programación entera.
- c) (3 puntos) Analizar los valores que toman las distintas variables de decisión en la solución relajada, y en caso de ser necesario, utilizar el algoritmo de *Branch & Bound* para obtener el valor óptimo del problema entero.