Preguntas teoricas de coloquio

1-¿Por qué el algoritmo simplex se ha mantenido vigente tanto tiempo a pesar de que no obtiene de una manera
rápida la solución óptima de los modelos de programación lineal continua?

Creo que el método simplex es en teoría ineficiente (el peor caso podría recorrer todos los vertices que sería parecido a probar todas las posibilidades) pero en la práctica converge rápidamente. ¿Es así?

En el caso de la pregunta 1, como vos decís, desde el punto de vista teórico, el método simplex no es un algoritmo que en el peor de los casos termine en un tiempo polinomial (porque recorrería todos los vértices). Sin embargo, la probabilidad de que en un problema real se de el peor caso es bajísima, así que en la práctica funciona muy bien.

2-¿Qué diferentes criterios se pueden usar para clasificar una formulación como "buena"?, ¿cuál fue el criterio adoptado en clase (teórico-prácticas)?: según el criterio adoptado en clase, ¿qué características tiene una formulación de programación lineal continua para ser buena?, ¿qué características tiene una formulación de programación lineal entera para ser buena?

El tema de la pregunta 2 no lo dimos este cuatrimestre, pero te cuento que los criterios son:

- Un buen modelo de PLC tiene pocas restricciones y variables.
- Un buen modelo de PLE tiene un poliedro cercano a la cápsula convexa.

3-En tu opinión, el problema de la mochila que vimos en clase ¿es fácil o difícil? ¿por qué considerás que es así?. Si se permitiera fraccionar los ítems para ponerlos en la mochila (es decir, que no sea necesario poner un ítem entero sino que se pueda poner parte del ítem), ¿cambiaría la complejidad del problema? ¿por qué?

El problema es dificil porque no conocemos ningún algoritmo eficiente para resolverlo (no hay ninguno en tiempo polinomial). Si se pudiera fraccionar creo que sería mucho mas simple porque podríamos elegir cualquiera de los productos y fraccionarlo para que complete la mochila (elegiría el de mejor beneficio con respecto al peso).

En el caso de la pregunta 3, sí, el problema es difícil porque no se conoce un algoritmo que en tiempo polinomial pueda resolverlo, cuando los ítems tienen que entrar en cantidades enteras. Si pudieran entrar en cantidades continuas es trivial, mete enteros

los de mejor relación beneficio/peso y cuando no le alcanza para meter uno entero, mete una fracción de ese producto que no entra entero.

4 - como encaramos un problema si no sabemos si es facil o dificil?

En primer lugar, si no tiene variables enteras no es un problema difícil.
Si tiene variables enteras hay que empezar por caracterizar y tratar de ver si es semejante a alguno de los problemas combinatorios conocidos (viajante, mochila, cobertura de conjuntos, etc.). Si se puede plantear como uno de esos problemas, es difícil.
Sino, hay que resolverlo con algún algoritmo de resolución de problemas enteros y si tarda demasiado tiempo y no finaliza, con una heurística

5 - Para resolver un modelo de Programación Lineal Entera ¿podemos usar el método simplex? ¿qué diferencias hay entre resolver un modelo de Programación Lineal Entera y un modelo de Programación Lineal Continua?

El método Simplex se utiliza para resolver problemas de Programación Lineal Contínua por lo tanto no te garantiza que la solución óptima encontrada sea entera.
Te puede llegar a servir para encontrar el óptimo si el óptimo del problema contínuo y el del entero coinciden, sea por casualidad o porque las características del problema lo garantizan, como ocurre en el Problema de Distribución.
En otros casos, lo podés llegar a usar como parte de un método de resolución que necesite obtener soluciones parciales (por ejemplo, Branch and Bound).

El problema de conjuntos a cubrir ¿es un problema difícil? Definir brevemente el problema e indicar porque es un problema difícil o porque no lo es.

Es un problema difícil ya que si la cantidad de elementos del conjunto es grande, las combinaciones posibles de subconjuntos son muchas. Como lo dice el nombre se trata de un problema donde tengo elementos en uno más conjuntos, y se desea seleccionar los subconjuntos de manera de cubrir todos los elementos con solapamiento o la mayor cantidad posible sin solapamientos.

De los 2 planteos de modelización más conocidos para el viajante (plantear todas las restricciones que evitan subtours, también llamado MZT y el que agrega las restricciones de Ui) ¿Cuál es el mejor en términos de resolución del problema? ¿Por qué piensa que ese es el mejor?

El planteo mediante las restricciones de Ui es mejor ya que con una sola restricción evito los subtours, en cambio con MZT, tengo que establecer una restricción para cada subtour posible. Para casos donde la cantidad de nodos es grande, necesito un número muy grande de restricciones lo que complica el armado del modelo.

En las preguntas 2 y 3 te aclaro que el planteo con las Ui tiene muchas menos restricciones en problemas grandes (como bien decís) pero desde el punto de vista de resolución (en lo referente a teoría de complejidad de algoritmos) el planteo MZT da una cápsula convexa que está más cerca de la cápsula convexa entera del problema que el planteo con las Ui. Es decir, según el criterio de planteo con menos restricciones es mejor el de las Ui, pero según el criterio de resolución es mejor el MZT

En las clases teórico prácticas se vieron dos maneras de hacer el modelo para el problema del viajante: una usando variables Ui y la otra armando las restricciones (combinando las Yij) que evitan todos los posibles subtours. ¿Según qué criterio el modelo de las Ui es el mejor de los dos y según qué criterio el modelo de evitar subtours es el mejor de los dos?

El armado de las restricciones combinando las Yij para evitar los subtours puede ser un mejor cuando se puede observar a simple vista cual o cuales son los subtours que se podrían formar y la cantidad de restricciones a escribir es pequeña. En cambio usando las variables Ui, se puede utilizar para un número más grande de problemas del viajante usando solo una restricción.

En las preguntas 2 y 3 te aclaro que el planteo con las Ui tiene muchas menos restricciones en problemas grandes (como bien decís) pero desde el punto de vista de resolución (en lo referente a teoría de complejidad de algoritmos) el planteo MZT da una cápsula convexa que está más cerca de la cápsula convexa entera del problema que el planteo con las Ui. Es decir, según el criterio de planteo con menos restricciones es mejor el de las Ui, pero según el criterio de resolución es mejor el MZT

El problema de asignación ¿es un problema difícil?

No, no es un problema dificil ya que puedo resolverlo utilizando programacion lineal continua y la solución obtenida sera factible? (siempre que sea el caso de asignacion clasica y no el de asignacion cuadratica).

C2 En el problema de coloreo de grafos, si se resuelve suponiendo que las variables pueden tomar valores no enteros (es decir, se lo resuelve como un problema continuo) ¿se puede usar esa solución como base para resolver el problema en el cual las variables son enteras? Indique las limitaciones de la solución continua (si las tiene).

No ya que la solucion podria incluir pintar un pais la mitad de un color y la mitad de otro, y algun pais limitrofe estar pintado de los mismos 2 colores, dando asi una solucion inconsistente en terminos de lo que queremos calcular, pero sera factible para el modelo ya que la sumatoria de ambos colores sera igual a 1 pero no se cumplira que un pais este pintado de un solo color y que Xij + Xkj <= wj

Xij = 1🡐el vértice i se colorea con color j

wj si se utiliza el color j

C1 ¿En qué afecta a la resolución de modelos de programación lineal entera que no se haya podido probar aún si P es igual o distinto que NP?

Nos afecta en que tenemos incertidumbre acerca de si podemos encontrar algun algoritmo capaz de resolverlo en tiempo polinomial o no

C2 De las heurísticas que conoce para el problema del viajante ¿cuál le parece que da mejores resultados y por qué?.

C1 Si existen procedimientos como Branch and Bound, Branch and Cut y otros, que resuelven de manera exacta un problema de programación lineal entera ¿por qué algunos problemas se resuelven de manera aproximada con un algoritmo heurístico?

C2 El problema de la mochila ¿es un problema difícil?. Definir brevemente el problema e indicar por qué es un problema difícil o por qué no lo es.

C2 Si el problema de distribución se puede resolver de manera exacta como un problema de programación lineal continua ¿por qué se utilizan heurísticas de construcción para ese problema?

Porque, como vimos, todas las restricciones del problema son igualdades, y si lo resolvemos por el método simplex tenemos que agregar una variable artificial por cada una, con lo que tendrían que pasar varias tablas hasta que encontremos una solución válida. En problemas de tamaño mayor es peor. Así que se usa una modificación del método partiendo de una primera solución válida

C1 Para resolver un problema de Programación Lineal Entera, uno de los procedimientos que se pueden utilizar es Branch & Bound ¿Cómo se puede acelerar la resolución por Branch & Bound para que termine antes?

Para optimizar en el caso de tener soluciones alternativas lo que se puede hacer es encuentro una solucion y le agrego un % y su una rama no es mayor a la sol+ % no la analizo para no perder tiempo analizando soluciones similares.

C1 ¿Para qué utiliza el método Branch and Bound la mejor solución entera encontrada hasta el momento?

C2 Para resolver un modelo de Programación Lineal Entera ¿podemos usar el método simplex? ¿qué diferencias hay entre resolver un modelo de Programación Lineal Entera y un modelo de Programación Lineal Continua?

No ya que en el metodo simplex voy recorriendo el poliedro de soluciones moviendome por las restricciones, en el caso de tener una solucion entera que no pertenezca a los vertices del poliedro, no podre hallar la solucion.

Cuando un problema es dificil?

Un problema se cataloga como inherentemente difcil si su
solucion requiere de una cantidad signi cativa de recursos computacionales,
sin importar el algoritmo utilizado