Introducción al Despacho en *Optimización de Problemas de Producción*

Docentes: Fernando Islas - Carlos Testuri
Colaboradores: Héctor Cancela - Antonio Mauttone - Pedro Piñeyro

Depto. Investigación Operativa. Instituto de Computación.
Facultad de Ingeniería. UdelaR

2023

Contenido

1. Problema de Despacho de Intervalos
2. Problema de Partición de Intervalos
3. Problema de Despacho de Intervalos Ponderados

Problema de Despacho de Intervalos (PDI)

Dado un *recurso* de uso exclusivo y un conjunto de *intervalos* continuos de tiempo que requieren su uso.

Problema de Despacho de Intervalos (PDI)

Dado un *recurso* de uso exclusivo y un conjunto de *intervalos* continuos de tiempo que requieren su uso.
Ejemplo: un salón podría ser el recurso y clases a dictarse en él los intervalos.
¿Qué otros dominios se pueden modelar con el problema?

Dado un *recurso* de uso exclusivo y un conjunto de *intervalos* continuos de tiempo que requieren su uso.
Ejemplo: un salón podría ser el recurso y clases a dictarse en él los intervalos.
¿Qué otros dominios se pueden modelar con el problema?

Se dice que un *par de intervalos es compatible* si estos no se superponen en el tiempo.

Problema de Despacho de Intervalos (PDI)

Dado un *recurso* de uso exclusivo y un conjunto de *intervalos* continuos de tiempo que requieren su uso.
Ejemplo: un salón podría ser el recurso y clases a dictarse en él los intervalos.
¿Qué otros dominios se pueden modelar con el problema?

Se dice que un *par de intervalos es compatible* si estos no se superponen en el tiempo.
En general se dice que un *subconjunto de los intervalos es compatible* si todo par de intervalos de este es compatible.

Problema de Despacho de Intervalos (PDI)

Dado un *recurso* de uso exclusivo y un conjunto de *intervalos* continuos de tiempo que requieren su uso.
Ejemplo: un salón podría ser el recurso y clases a dictarse en él los intervalos.
¿Qué otros dominios se pueden modelar con el problema?

Se dice que un *par de intervalos es compatible* si estos no se superponen en el tiempo.
En general se dice que un *subconjunto de los intervalos es compatible* si todo par de intervalos de este es compatible.

Problema: Dado un conjunto de intervalos. ¿Cómo determinar un subconjunto de intervalos compatibles de mayor tamaño posible?

Formulación del Problema de Despacho de Intervalos

Sea I el conjunto de los intervalos identificados por $i = 1, ..., n$, con tiempos de comienzo c_i y de finalización f_i tales que $c_i \leq f_i$.

Los intervalos i y j son compatibles si uno finaliza antes que el otro comience:

$$f_i \leq c_j \quad \text{ó} \quad f_j \leq c_i.$$

El subconjunto de intervalos $S \subseteq I$ se dice compatible si para todo par $i, j \in S$ tal que $i \neq j$ se cumple que i y j son compatibles.

Problema: Se busca determinar un subconjunto S de mayor tamaño posible.

Técnica de resolución para el Problema de Despacho de Intervalos

Una técnica de resolución de algunos problemas de optimización consiste en determinar su solución mediante una secuencia de decisiones. Donde en cada etapa de la secuencia *se toma la mejor decisión disponible*, con la esperanza de llegar a una solución óptima. Para esto se necesita una *regla* que determine la mejor decisión disponible. La técnica se denomina *algoritmo ávido (greedy)*.

Técnica de resolución para el Problema de Despacho de Intervalos

Una técnica de resolución de algunos problemas de optimización consiste en determinar su solución mediante una secuencia de decisiones. Donde en cada etapa de la secuencia *se toma la mejor decisión disponible*, con la esperanza de llegar a una solución óptima. Para esto se necesita una *regla* que determine la mejor decisión disponible. La técnica se denomina *algoritmo ávido (greedy)*.

En el PDI la técnica consiste en considerar secuencialmente la compatibilidad de cada intervalo con un subconjunto de intervalos compatibles que se determinó previamente.
A partir de alguna regla se considera un primer intervalo. Luego se van considerando según la regla los restantes intervalos que son compatibles con los que ya se ha determinado que son parte de la solución.

Técnica de resolución para el Problema de Despacho de Intervalos

Una técnica de resolución de algunos problemas de optimización consiste en determinar su solución mediante una secuencia de decisiones. Donde en cada etapa de la secuencia *se toma la mejor decisión disponible*, con la esperanza de llegar a una solución óptima. Para esto se necesita una *regla* que determine la mejor decisión disponible. La técnica se denomina *algoritmo ávido (greedy)*.

En el PDI la técnica consiste en considerar secuencialmente la compatibilidad de cada intervalo con un subconjunto de intervalos compatibles que se determinó previamente.
A partir de alguna regla se considera un primer intervalo. Luego se van considerando según la regla los restantes intervalos que son compatibles con los que ya se ha determinado que son parte de la solución.

¿Qué reglas se pueden considerar?

Reglas de selección de intervalos para algoritmo de resol. del PDI (1/2)

Regla 1: *Seleccionar el intervalo disponible que comienza más temprano.*

Regla 1: *Seleccionar el intervalo disponible que comienza más temprano.*
Si el intervalo seleccionado que comienza más temprano tiene una amplitud muy grande, otros intervalos que se superponen con él quedan excluidos de la solución.
Por lo cual se podría terminar con una solución que no es la mejor.

Reglas de selección de intervalos para algoritmo de resol. del PDI (1/2)

Regla 1: *Seleccionar el intervalo disponible que comienza más temprano.*
Si el intervalo seleccionado que comienza más temprano tiene una amplitud muy grande, otros intervalos que se superponen con él quedan excluidos de la solución.
Por lo cual se podría terminar con una solución que no es la mejor.

Regla 2: *Seleccionar el intervalo disponible que tiene la amplitud más pequeña.*

Reglas de selección de intervalos para algoritmo de resol. del PDI (1/2)

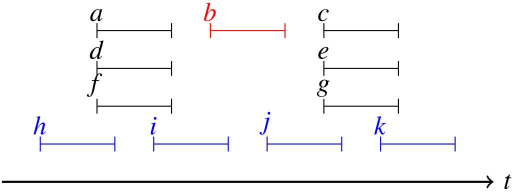
Regla 1: *Seleccionar el intervalo disponible que comienza más temprano.*
Si el intervalo seleccionado que comienza más temprano tiene una amplitud muy grande, otros intervalos que se superponen con él quedan excluidos de la solución.
Por lo cual se podría terminar con una solución que no es la mejor.

Regla 2: *Seleccionar el intervalo disponible que tiene la amplitud más pequeña.*
También, puede llevar a una solución que no es la mejor. A modo de contraejemplo se tienen los intervalos: $(c_1, f_1) = (1, 5)$, $(c_2, f_2) = (6, 10)$, y $(c_3, f_3) = (4, 7)$.
El intervalo 3 se consideraría como parte de la solución, dado que tiene la menor amplitud y no es compatible con los otros dos. Lo cual excluye la solución óptima que forman los otros dos intervalos.

Reglas de selección de intervalos para algoritmo de resol. del PDI (2/2)

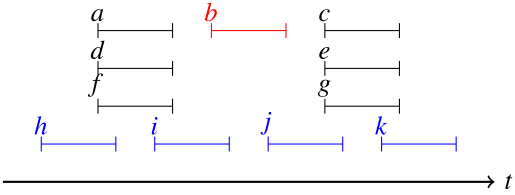
Regla 3: *Seleccionar el intervalo disponible con menor cantidad de intervalos no compatibles.*

Regla 3: *Seleccionar el intervalo disponible con menor cantidad de intervalos no compatibles*.
También, puede llevar a una solución que no es la mejor. A modo de contraejemplo se tienen los intervalos



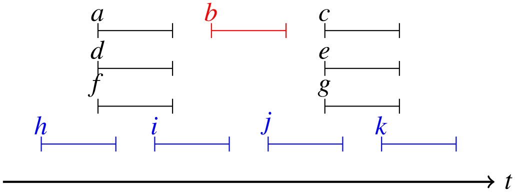
Reglas de selección de intervalos para algoritmo de resol. del PDI (2/2)

Regla 3: *Seleccionar el intervalo disponible con menor cantidad de intervalos no compatibles.*
También, puede llevar a una solución que no es la mejor. A modo de contraejemplo se tienen los intervalos



Regla 4: *Seleccionar el intervalo disponible que finaliza más temprano.*

Regla 3: *Seleccionar el intervalo disponible con menor cantidad de intervalos no compatibles*.
También, puede llevar a una solución que no es la mejor. A modo de contraejemplo se tienen los intervalos



Regla 4: *Seleccionar el intervalo disponible que finaliza más temprano*.
Esta regla asegura que el recurso esté disponible lo antes posible luego de atender un intervalo, por lo cual maximiza el tiempo restante para atender a los intervalos remanentes.

Algoritmo para PDI con selección de intervalo que finaliza más temprano

Algoritmo ávido para resolver el PDI con Regla 4:

```
Datos: I : conjunto de intervalos a considerar
Resultado: S : subconjunto de intervalos solución (inicialmente vacío)
mientras I no es vacío hacer
    Elegir el intervalo i ∈ I con menor tiempo de finalización ;
    Agregar i a S ;
    Eliminar de I el intervalo i y los intervalos que no son compatibles
      con i ;
fin
devolver S;
```

Algoritmo ávido para resolver el PDI con Regla 4:

Datos: I : conjunto de intervalos a considerar
Resultado: S : subconjunto de intervalos solución (inicialmente vacío)
mientras *I no es vacío* **hacer**
  Elegir el intervalo $i \in I$ con menor tiempo de finalización ;
  Agregar i a S ;
  Eliminar de I el intervalo i y los intervalos que no son compatibles con i ;
fin
devolver S;

Análisis del algoritmo

a. ¿Es el resultado S un conjunto compatible?
b. ¿Es el resultado S una solución óptima?

Formulación del PDI como problema de optimización (Ejercicio)

¿Cómo se podría formular el PDI como problema de optimización?

¿Cómo se podría formular el PDI como problema de optimización?

Sugerencia. Asumir que se dispone del conjunto de pares de intervalos que se superponen, $P := \{(i,j) \in I \times I | (i,j)$ no son compatibles e $i \leq j\}$.

Problemas derivados del Problema de Despacho de Intervalos

1. Los intervalos se van presentando a medida que se resuelve el problema, es decir no todos son conocidos previo a la resolución del problema.

Problemas derivados del Problema de Despacho de Intervalos

1. Los intervalos se van presentando a medida que se resuelve el problema, es decir no todos son conocidos previo a la resolución del problema.
2. Se requiere determinar la cantidad mínima de recursos que permite atender a todos los intervalos de forma compatible en cada recurso. Además, se requiere establecer para cada intervalo que recurso lo atiende (Problema de Partición de Intervalos).

Problemas derivados del Problema de Despacho de Intervalos

1. Los intervalos se van presentando a medida que se resuelve el problema, es decir no todos son conocidos previo a la resolución del problema.

2. Se requiere determinar la cantidad mínima de recursos que permite atender a todos los intervalos de forma compatible en cada recurso. Además, se requiere establecer para cada intervalo que recurso lo atiende (Problema de Partición de Intervalos).

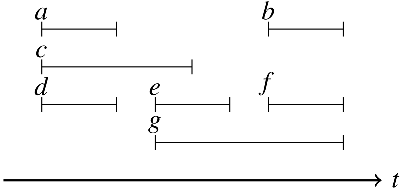
3. Los intervalos están valorados y se busca determinar el subconjunto de intervalos compatibles de mayor suma de valores (Problema de Despacho de Intervalos Ponderados).

Problema de Partición de Intervalos (PPI) (1/3)

Esta variante del problema busca determinar la cantidad mínima de recursos (indistinguibles) que permite atender a todos los intervalos. Es decir se busca dividir el conjunto de intervalos (ej. clases) en recursos (ej. salones) de forma tal que no se superponen en cada recurso.

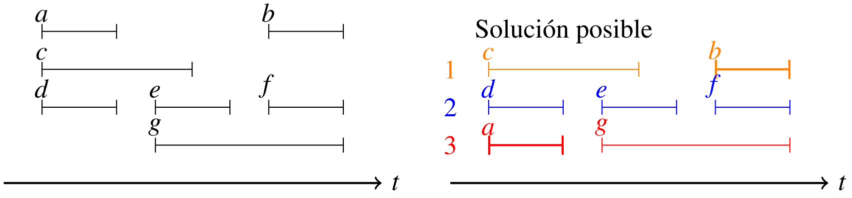
Esta variante del problema busca determinar la cantidad mínima de recursos (indistinguibles) que permite atender a todos los intervalos. Es decir se busca dividir el conjunto de intervalos (ej. clases) en recursos (ej. salones) de forma tal que no se superponen en cada recurso.

Ejemplo con intervalos identificados por letras y recursos por números:



Esta variante del problema busca determinar la cantidad mínima de recursos (indistinguibles) que permite atender a todos los intervalos. Es decir se busca dividir el conjunto de intervalos (ej. clases) en recursos (ej. salones) de forma tal que no se superponen en cada recurso.

Ejemplo con intervalos identificados por letras y recursos por números:



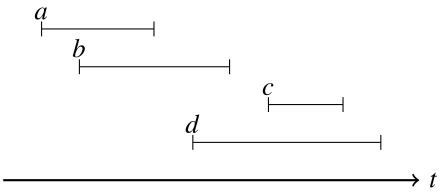
¿Es posible usar menos de tres recursos?

Problema de Partición de Intervalos (PPI) (2/3)

En cada instancia del PPI, la cantidad de recursos necesaria es al menos la cantidad máxima de intervalos que se superponen en el tiempo.
Esta cantidad, denotada h, es una cota mínima que toda solución debe tener.
Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

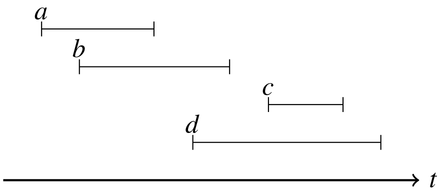
En cada instancia del PPI, la cantidad de recursos necesaria es al menos la cantidad máxima de intervalos que se superponen en el tiempo.
Esta cantidad, denotada h, es una cota mínima que toda solución debe tener.
Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

Dada la instancia del problema:



En cada instancia del PPI, la cantidad de recursos necesaria es al menos la cantidad máxima de intervalos que se superponen en el tiempo.
Esta cantidad, denotada h, es una cota mínima que toda solución debe tener.
Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

Dada la instancia del problema:



¿Cuál es su mínima cantidad de recursos necesarios, h?

Problema de Partición de Intervalos (PPI) (3/3)

Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

Problema de Partición de Intervalos (PPI) (3/3)

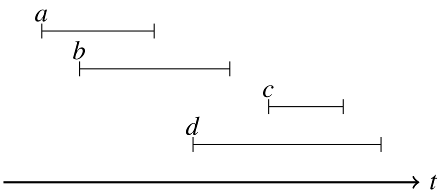
Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

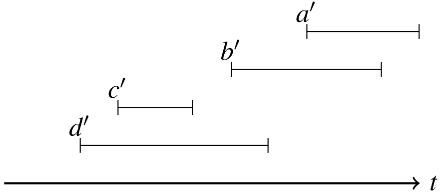
¿Qué regla de selección de intervalos permite resolver el problema?

Problema de Partición de Intervalos (PPI) (3/3)

Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

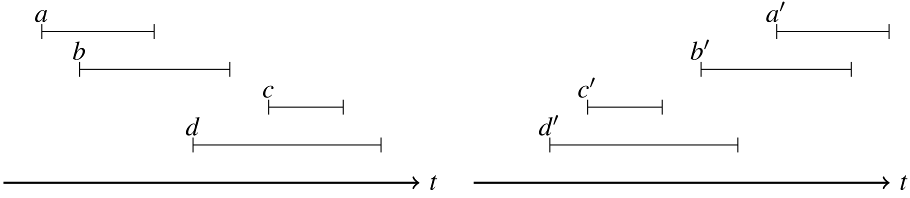
¿Qué regla de selección de intervalos permite resolver el problema? Considerar la regla propuesta en las instancias:





Para resolver el problema se necesita asignarle a cada intervalo un recurso, es decir asignarle un elemento del conjunto $\{1, 2, ..., h\}$.

¿Qué regla de selección de intervalos permite resolver el problema?
Considerar la regla propuesta en las instancias:



Se propone un algoritmo de resolución que recorre el conjunto de intervalos ordenados por su *tiempo de comienzo creciente*. Luego para cada intervalo intenta asignar un recurso que no haya sido asignado previamente a otro intervalo que se superpone con él.

Algoritmo ávido para el Problema de Partición de Intervalos (PPI)

Datos: I : conjunto de intervalos a considerar
Datos: h : cantidad de recursos necesaria
Resultado: s_i : recurso asignado al intervalo $i = 1, ..., n$ (inicialmente vacíos)
Ordenar el conjunto de intervalos I según tiempo de comienzo creciente. Sea $\{I_1, I_2, ..., I_n\}$ el conjunto ordenado ;
para $i = 1, 2, ..., n$ **hacer**
    Sea $R = \{1, 2, ..., h\}$ el conjunto de recursos en consideración ;
    para $j = 1, 2, ..., i - 1$ **hacer**
        si *I_j se superpone con I_i* **entonces**
            Se excluye el recurso asignado a I_j de consideración: $R := R \setminus \{s_j\}$;
        fin
    fin
    si *existe algún recurso r que no ha sido excluido de R* **entonces**
        Asignar el recurso r al intervalo i: $s_i := r$;
    fin
fin
devolver $s_i, i = 1, ..., n$; /* Según intervalos ordenados por tiempo comienzo */ ;

¿Cómo se podría formular el PPI como problema de optimización?

Formulación del PPI como problema de optimización (Ejercicio)

¿Cómo se podría formular el PPI como problema de optimización?

Sugerencia. Asumir que se dispone del conjunto de pares de intervalos que se superponen, $P := \{(i,j) \in I \times I | (i,j) \text{ no son compatibles e } i \neq j\}$.

Problema de Despacho de Intervalos Ponderados (PDIP)

Es una generalización del Problema de Despacho de Intervalos (PDI) en la cual a cada intervalo $i = 1, ..., n$ se le asocia un valor $v_i > 0$.

¿Cómo determinar el subconjunto de intervalos compatibles de mayor suma de valores?

Problema de Despacho de Intervalos Ponderados (PDIP)

Es una generalización del Problema de Despacho de Intervalos (PDI) en la cual a cada intervalo $i = 1, ..., n$ se le asocia un valor $v_i > 0$.

¿Cómo determinar el subconjunto de intervalos compatibles de mayor suma de valores?

Notar que el PDI es un caso especial de este, en el cual el valor de todos los intervalos es uno.

Problema de Despacho de Intervalos Ponderados (PDIP)

Es una generalización del Problema de Despacho de Intervalos (PDI) en la cual a cada intervalo $i = 1, ..., n$ se le asocia un valor $v_i > 0$.

¿Cómo determinar el subconjunto de intervalos compatibles de mayor suma de valores?

Notar que el PDI es un caso especial de este, en el cual el valor de todos los intervalos es uno.

El algoritmo ávido para el PDI (que selecciona el intervalo que finaliza más temprano) no funciona para el PDIP (Ejercicio: determinar un contraejemplo).

Problema de Despacho de Intervalos Ponderados (PDIP)

Es una generalización del Problema de Despacho de Intervalos (PDI) en la cual a cada intervalo $i = 1, ..., n$ se le asocia un valor $v_i > 0$.

¿Cómo determinar el subconjunto de intervalos compatibles de mayor suma de valores?

Notar que el PDI es un caso especial de este, en el cual el valor de todos los intervalos es uno.

El algoritmo ávido para el PDI (que selecciona el intervalo que finaliza más temprano) no funciona para el PDIP (Ejercicio: determinar un contraejemplo).

Actualmente no se conoce un algoritmo ávido que resuelva el PDIP .

Esquema de resolución para el PDIP (1/2)

Se asume que los intervalos están ordenados según tiempo de finalización no decreciente: $f_1 \leq f_2 \leq \ldots \leq f_n$. A partir de lo cual se dice que el intervalo i *tiene lugar antes* que el intervalo j si $i < j$. Para un intervalo j, se define $p(j)$ como el *intervalo más próximo* que finaliza antes que j comience; es decir como el intervalo i de mayor valor de índice tal que $i < j$ con i y j compatibles. Se define $p(j) = 0$ si no hay intervalo compatible con j.

Esquema de resolución para el PDIP (1/2)

Se asume que los intervalos están ordenados según tiempo de finalización no decreciente: $f_1 \leq f_2 \leq \ldots \leq f_n$. A partir de lo cual se dice que el intervalo i *tiene lugar antes* que el intervalo j si $i < j$. Para un intervalo j, se define $p(j)$ como el *intervalo más próximo* que finaliza antes que j comience; es decir como el intervalo i de mayor valor de índice tal que $i < j$ con i y j compatibles. Se define $p(j) = 0$ si no hay intervalo compatible con j.

Si se asume como solución óptima de una instancia del problema a S. Puede ocurrir que el intervalo n sea parte de S o no lo sea. Si n pertenece a S entonces ningún intervalo indexado entre $p(n) + 1$ y n pertenece a S, debido que se superponen con n. Además, alguno de los intervalos indexados entre 1 y $p(n)$ pertenece a S, dado que no se superpone con n. Por otra parte, si n no pertenece a S, entonces S esta compuesto solo por intervalos entre 1 y $n - 1$.

Esquema de resolución para el PDIP (2/2)

Por lo cual determinar la solución óptima a un problema de intervalos $\{1, ..., n\}$ puede descomponerse en determinar la solución a problemas de intervalos $\{1, ..., i\}$, con $i = 1, ..., n$. Sea S_i la solución óptima de cada problema i, y $O(i)$ el valor de la solución (Se asume que la solución vacía tiene valor cero, $O(0) = 0$).

Esquema de resolución para el PDIP (2/2)

Por lo cual determinar la solución óptima a un problema de intervalos $\{1, ..., n\}$ puede descomponerse en determinar la solución a problemas de intervalos $\{1, ..., i\}$, con $i = 1, ..., n$. Sea S_i la solución óptima de cada problema i, y $O(i)$ el valor de la solución (Se asume que la solución vacía tiene valor cero, $O(0) = 0$).
El objetivo es determinar la solución S_n cuyo valor $O(n)$ es máximo. Por lo visto anteriormente, para la solución óptima S_i, en $\{1, ..., i\}$ se cumple, si $i \in S_i$ entonces $O(i) = v_i + O(p(i))$, ó en caso contrario, si $i \notin S_i$ entonces $O(i) = O(i-1)$. Por lo cual la búsqueda del valor óptimo queda establecida por

$$O(i) = \max(v_i + O(p(i)), O(i-1)).$$

Ante lo cual el intervalo i pertenece a la solución S_i en $\{1, ..., i\}$ si y solo si

$$v_i + O(p(i)) \geq O(i-1).$$

Este esquema, basado en una ecuación recurrente del valor óptimo del problema a partir de subproblemas, se denomina *programación dinámica*.

Algoritmo recurrente de resolución para el PDIP

La recurrencia sobre $O(i)$ permite resolver el problema al invocar el algoritmo recursivo *Calcular_O*(n),

Algoritmo *Calcular_O*(i) ;
si i = *0* **entonces**
  devolver 0 ;
en otro caso
  devolver $\max(v_j + \mathit{Calcular\_O}(p(i)), \mathit{Calcular\_O}(i-1))$;
fin

Formulación del PDIP como problema de optimización (Ejercicio)

¿Cómo se podría formular el PDIP como problema de optimización?

Formulación del PDIP como problema de optimización (Ejercicio)

¿Cómo se podría formular el PDIP como problema de optimización?

Sugerencia. Asumir que se dispone del conjunto de pares de intervalos que se superponen, $P := \{(i,j) \in I \times I | (i,j) \text{ no son compatibles e } i \neq j\}$.