



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia

Ingeniería de Sistemas y Computación

2025969 Modelos estocásticos y simulación en computación y comunicaciones (01)

Tarea 6

El problema de la programación académica en la UNAL

Integrantes:

Javier Andrés Tarazona Jiménez	jtarazonaj@unal.edu.co
Yenifer	@unal.edu.co
Jefferson	@unal.edu.co

Abril 15 de 2024

Tabla de Contenidos

1	Introducción	2
2	Marco Teórico	2
2.1	Ejemplo Documentación	2
2.1.1	Contextualización de la solución	2
2.1.2	Descripción general de la solución	2
2.1.3	Naturaleza de la solución	2
2.1.4	Funcionamiento del sistema	2
2.1.5	Resultados o impacto	2
2.1.6	Referencia	2
2.2	Bunded Probabilistic Serial (BPS): Yenifer	3
2.3	Random Serial Dictatorship (RSD): Jefferson	3
2.4	Probabilistic Serial Rule (PBS)	3
2.4.1	Descripción general de la solución	3
2.4.2	Naturaleza de la solución	3
2.4.3	Funcionamiento del sistema	3
2.4.4	Referencia	4
3	Descripción y Justificación del Problema a Resolver	4
3.1	Objetivo Principal	4
4	Diseño de la solución	5
4.1	Metodología	5
4.1.1	Preparación	5
4.1.2	Modelo Estocástico: Proceso de Asignación	7
5	Código Fuente	9
6	Manual Usuario	9
7	Manual Técnico	9
8	Experimentación	9
8.1	Análisis de resultados	9
8.1.1	Escenario 1	9
8.1.2	Escenario 2	9
8.1.3	Escenario 3	9
9	Referencias	10

1 Introducción

Aquí si empieza el contenido del artículo de la sección 1 y también hay una referencia [1].

2 Marco Teórico

Aquí empieza el contenido del artículo de la sección 1 y también hay una referencia [1].

2.1 Ejemplo Documentación

2.1.1 Contextualización de la solución

Problema identificado: ¿Qué necesidad o dificultad buscaba resolver la universidad? (por ejemplo: congestión en cursos, planificación ineficiente, asignación no equitativa, etc.)

Entorno institucional: ¿Qué tipo de universidad es? ¿Cuántos estudiantes? ¿Qué tan complejo es su sistema académico?

Objetivos del sistema de asignación: (eficiencia, justicia, reducción de tiempos administrativos, etc.)

2.1.2 Descripción general de la solución

Nombre o enfoque de la solución: (por ejemplo: algoritmo de colonia de hormigas, programación entera, RSD, etc.)

Tipo de herramienta: ¿Es una herramienta automatizada, un modelo matemático, un sistema web, un algoritmo implementado?

Actores involucrados: ¿Está orientado a estudiantes, administrativos, docentes?

2.1.3 Naturaleza de la solución

Aquí se refiere a la categoría técnica o metodológica de la solución:

¿Es un modelo de optimización matemática, una heurística, un sistema probabilístico, una metaheurística, un sistema de recomendación?

También puede implicar si es una solución centralizada (administración asigna) o descentralizada (los usuarios eligen con base en preferencias).

2.1.4 Funcionamiento del sistema

¿Cómo se realiza la asignación o planificación?

¿Qué datos se necesitan?

¿Cómo se expresan las preferencias?

¿Qué lógica sigue el algoritmo (en términos generales)?

¿Qué tan automático es?

¿Cómo se gestionan los conflictos o empates?

2.1.5 Resultados o impacto

¿Qué mejoras reportaron?

¿Se adoptó oficialmente?

¿Se continúa utilizando?

2.1.6 Referencia

Nombre del artículo o proyecto, universidad, y si es posible, enlace o fuente bibliográfica.

2.2 Bundled Probabilistic Serial (BPS): Yenifer

2.3 Random Serial Dictatorship (RSD): Jefferson

2.4 Probabilistic Serial Rule (PBS)

2.4.1 Descripción general de la solución

Propuesto por Bogomolnaia y Moulin (2001), el algoritmo Probabilistic Serial Rule, también llamado Probabilistic Serial Mechanism, es un mecanismo de asignación aleatoria para repartir bienes indivisibles (como cupos en asignaturas) respetando las preferencias de los participantes bajo un enfoque equitativo.

Algunos entornos donde el mismo se puede aplicar se encuentran en la asignación de cursos a estudiantes, como en el caso de la Universidad de París y en la Universidad de Tsinghua, reparto de turnos o recursos indivisibles, y asignación de cupos de movilidad internacional o servicios públicos.

Se podría decir entonces que se usa cuando un grupo de personas desea obtener uno (o varios) recursos limitados y los mismos no se pueden dividir, es decir, son discretos. Y se quiere una asignación que sea justa y eficiente en promedio, sin usar dinero ni intercambios, es decir, una solución equitativa que sea directa.

2.4.2 Naturaleza de la solución

Este algoritmo tiene una intuición muy práctica. Suponga un proceso en el cual se encuentran participantes de un buffet. En el mismo, cada uno empieza a comer de forma simultánea y de forma continua el plato que desee comer, es decir, su preferido. Una vez que dicho plato termina, los integrantes pasan a comer su segundo plato preferido y así sucesivamente hasta que se terminan los platos.

Esta solución cuenta con unas ventajas y otras desventajas.

Ventajas

- El método es eficiente, ya que maximiza el bienestar de los integrantes según preferencias. Ellos mismos buscan el "orden".
- Es justo y equitativo, ya que ningún integrante tiene prioridades.
- No es un método manipulable por los integrantes como el RSD, ya que, en principio, hacer una estrategia es una mala idea por parte de los participantes, pues no hay prioridad. Suponga que la mejor opción es seleccionar lo que más quieren, porque de lo contrario se pueden quedar sin dicho recurso una vez se realice la repartición simultánea (Weak Strategy-Proofness).

Desventajas

- La asignación que se realiza es netamente probabilística y no determinista, pues depende de las elecciones aleatorias de los integrantes y sus preferencias subjetivas. Usualmente, el método viene con una segunda etapa de ronda de sorteos (*lottery decomposition*).
- Su carácter de paralelismo y varias variables hace que sea más difícil de implementar que RSD o BPS.

2.4.3 Funcionamiento del sistema

De forma generalizada, tenemos n integrantes de un evento con m recursos para escoger, y cada uno con una disponibilidad de d_m . Cada integrante va a consumir, de forma paralela, un elemento de m , reduciendo su disponibilidad asociada d_m bajo una velocidad uniforme, es decir, todos a la misma velocidad. Cuando d_m es cero, los integrantes consumen su siguiente recurso m preferido. Y así se continúa hasta que todos los d_m sean iguales a 0.

Como lo puede notar, este proceso no determina una asignación exacta, sino que es una asignación probabilística. Otra forma de verlo es como una lotería justa, porque cada integrante tiene una probabilidad de obtener el recurso según sus preferencias y su comportamiento.

Al final del proceso, hay una interpretación probabilística y una fase de sorteo. En este punto, cada integrante tiene una fracción de cada recurso m ; este valor acumulado es la probabilidad de cada integrante de obtener el recurso en una asignación final. Aquí hay dos casos:

- **Caso 1: Los integrantes pueden recibir varios recursos**

Se asigna cada recurso de forma individual según la probabilidad registrada.

- **Caso 2: Cada integrante debe recibir exactamente un recurso**

Hay que descomponer la matriz de probabilidades en una combinación convexa de matrices de permutación (Ej: Algoritmo de Birkhoff) para generar una asignación determinista a partir de las probabilidades.

Consideraciones

- A pesar de que se señaló que la velocidad debe ser uniforme, se pueden plantear prioridades para ciertos integrantes aumentando su velocidad.
- Se asume que los agentes son neutralmente riesgosos. Es decir, que su satisfacción se mide por el valor esperado de la utilidad obtenida de la asignación final. Es decir, que el mecanismo se basa en las preferencias que expresan los agentes en vez de su actitud frente al riesgo.
- Hay una resistencia a la manipulación si un agente conoce las preferencias de los otros y encuentra satisfacción en el riesgo. Pero este beneficio está acotado a lo sumo al 50% de ganancia adicional, es decir, un factor de $\frac{3}{2}$.
- El método presenta equilibrio de Nash; hay perfiles de preferencias en los que ningún agente puede mejorar unilateralmente su resultado.
- Bajo comparación lexicográfica decreciente, se puede calcular una respuesta óptima en tiempo razonable a la manipulación estratégica. Pero cuando hay que maximizar la utilidad esperada para cada agente (si hay más de dos agentes), el problema se vuelve NP-Hard. Esto hace al método más robusto contra manipulaciones.

2.4.4 Referencia

Vea [2].

3 Descripción y Justificación del Problema a Resolver

La Universidad Nacional de Colombia, como uno de sus fines misionales, tiene por objetivo impartir conocimiento al país, en particular a su comunidad de estudiantes mediante la docencia. Para realizarlo, se ofrecen cursos (asignaturas) como actividades académicas; sin embargo, dicha oferta requiere recursos y la sincronización de horarios, docentes y aulas. Y al mismo tiempo, dicho proceso se ve entorpecido por la falta de recursos, la pésima administración, las garantías académicas que se usan con poca conciencia por integrantes de la comunidad estudiantil, como la libre inscripción y cancelación, o el aprovechamiento del sistema mediante estrategias de manipulación.

Ahora bien, dicho problema se puede abordar mediante modelación estocástica, planteando una simulación que emplee algoritmos y las condiciones de la Universidad para encontrar una estrategia y solución óptima al problema real.

3.1 Objetivo Principal

- Formular *un modelo estocástico* que describa el problema de planificación académica en la Universidad Nacional.

- *Simular* ese modelo para encontrar al menos *una solución óptima* que contribuya a mitigar el caos descrito.

Variables que afectan:

- Mala administración.
- Insuficiencia de recursos (humanos, físicos, logísticos).
- Comportamiento estratégico o irresponsable de estudiantes (inscripción y cancelación masiva).

4 Diseño de la solución

Para abordar la solución, se presentan las áreas y abstracciones que se van a abordar.

- La oferta.
 - Cursos.
 - Docentes.
 - Salones (y laboratorios).
 - Horarios.
- Demanda fluctuante y mal gestionada.
 - Preferencias no explícitas.
 - Manipulación.
 - Cancelaciones y sobreinscripción.
- Solución.Problemas
 - Falta de coordinación.
 - Falta de mecanismos eficientes de asignación.
- **Objetivo esperado.**
 - Obtener una planificación justa, óptima y automatizada.

4.1 Metodología

Para el desarrollo se abordará el problema en fases. Y el programa se realizará a través de módulos bajo una arquitectura de componentes.

4.1.1 Preparación

Módulo de Oferta

Este módulo tiene como objetivo generar una oferta académica simulada de cursos a partir de una base de datos inicial, seleccionando aleatoriamente docentes, aulas y horarios bajo una distribución uniforme. Esta etapa corresponde a una primera aproximación estocástica del sistema, sin aún aplicar criterios de optimización o restricciones complejas.

Constantes del programa (Se pueden tomar de una base de datos o emularlas):

- **Cursos:** Lista de diccionarios, donde cada elemento representa un curso con los siguientes atributos:
 - **codigo** (int): identificador único del curso.
 - **nombre** (str): nombre del curso.

- **creditos** (int): número de créditos académicos del curso.
- **programado** (bool): si ya se programó el curso.
- **Docentes:** Lista de diccionarios, cada uno con:
 - **cedula** (int): identificador del docente.
 - **nombre** (str): nombre completo del docente.
 - **programado** (bool): si ya se agendó a un curso.
- **Aulas:** Lista de diccionarios, cada uno con:
 - **edificio** (int): número del edificio.
 - **aula_id** (str): identificador del aula.
 - **tipo** (str): tipo de espacio (e.g., “laboratorio”, “salón”).
 - **programado** (bool): si ya se agendó a un curso.
- **Horarios disponibles:** Lista de diccionarios con:
 - **dia** (str): día de la semana, de lunes a sábado.
 - **franja** (str): intervalo de tiempo disponible (e.g., 7-9, 9-11, 11-13, 14-16, 16-18, 18-20, 7-13, 14-18).

Para cada curso, el módulo va a realizar lo siguiente:

1. **Selección de docente:** Se elige aleatoriamente un docente disponible (que no esté asignado a otro curso en el mismo horario) y se marca como programado.
2. **Selección de aula:** Se selecciona aleatoriamente un aula disponible (que no esté ocupada en el horario asignado) y se marca como ocupada.
3. **Selección de horario:** Se selecciona aleatoriamente un horario. Para garantizar una distribución balanceada de clases a lo largo de la semana, se propone el siguiente esquema de asignación adaptativa:
 - Llevar un conteo de asignaciones por cada día de la semana.
 - Definir una función de probabilidad que asigne mayor peso a los días menos cargados.
 - Aplicar una probabilidad ponderada inversamente a la carga actual de cada día:
$$P(dia_i) \propto \frac{1}{1 + asignaciones : dia_i}$$
 - Excluir los días viernes y sábado inicialmente. Estos se incluirán únicamente cuando reste el 5% de los cursos por asignar.
4. **Creación de la clase:** Se registra la asignación como un nuevo diccionario en la colección **clases**, la cual contiene los siguientes campos: **codigo_clase** generado aleatoriamente, **codigo_curso**, **cedula_docente**, **horario** y **aula** (identificada como **edificio+aula_id**). Además, se asigna el número de **cupos** como un valor aleatorio entre 30 y 60, calibrado de modo que el promedio de cupos se acerque al valor proporcionado por el usuario.

Módulo de Estudiantes

Aquí se definirán los deseos de los estudiantes.

- **Estudiante:** Lista de diccionarios con
 - Cédula estudiante (int)
 - Nombre estudiante (str)
 - Lista de deseos de tamaño n donde cada elemento es un código de clase.
 - Lista de asignaciones, las asignaturas que tiene inscritas

Para cada estudiante se crea una lista de deseos seleccionando aleatoriamente, bajo una distribución uniforme, un código de clase y agregándolo a dicha lista.

4.1.2 Modelo Estocástico: Proceso de Asignación

Ahora bien, una vez que la oferta de clases ya está construida y los estudiantes también están definidos junto con sus preferencias, sigue el proceso de asignación y selección de asignaturas. Para realizar dicho proceso, se plantean tres escenarios donde la asignación se realiza mediante los algoritmos Bundled Probabilistic Serial, Random Serial Dictatorship y Probabilistic Serial Rule.

Bundled Probabilistic Serial (BPS)

- **Inicialización**

- Se crea la lista de asignaciones vacía para cada estudiante
- Se crea un diccionario de probabilidades de asignar la asignatura en su lista de deseos. La probabilidad para la asignatura a_i en la lista de deseos será

$$P[a_i] = \frac{1}{\text{posicion}(a_i, \text{listaDeseos})}$$

Donde $\text{posicion}(a_i, \text{listaDeseos})$ es la posición de dicha asignatura en la lista de deseos.

Dicha probabilidad se guarda así:

$$\text{probabilidadesAsignaturas}[(\text{cedulaEstudiante}, \text{codClase})] = P[a_i]$$

- Se inicializa en 0 la suma de estudiantes que no pueden inscribir

- **Iteración**

- La iteración se realiza para cada estudiante y se vuelve a repetir hasta que no hayan mas cupos, todos los estudiantes ya no tengan materias en sus listas de deseos o el número de estudiantes que no pudieron inscribir sea igual al número de estudiantes.
- En el estudiante i , va a consumir su materia con mayor probabilidad si tiene cupos disponibles (si no, pasa a la siguiente y suma un estudiante que no pudo inscribir). Entonces, se reducen los cupos en uno, se elimina la clase de la lista de deseos y se agrega a la lista de inscritas. La probabilidad de sus materias deseadas se actualiza nuevamente así como en la inicialización y la de la materia inscrita se pone en 0.

Bundled Probabilistic Serial (BPS) Ajustado

- **Inicialización**

- Para cada estudiante, se crea una lista vacía de asignaciones y una lista de *bundles* (grupos de clases deseadas a partir de su lista de deseos de asignaturas -¿ crear la combinación respectiva de 1 a n materias: parámetro de entrada).
- Se asigna una función de utilidad o prioridad a cada bundle, normalmente derivada de las preferencias del estudiante. Donde b_i es un bundle, esto es:

$$P[b_i] = \frac{1}{\text{posicion}(a_i, \text{listaDeseosBundle})}$$

- Se inicializa una matriz de consumo $C[e, b] = 0$ donde e es el estudiante y b un bundle.

- **Consumo continuo**

- Se simula un avance en el tiempo continuo. En cada instante, cada estudiante consume su bundle preferido disponible a una tasa uniforme.
- Si el bundle contiene varias clases, sólo se puede consumir si todas las clases del bundle tienen cupos disponibles.

- A medida que un bundle es consumido por varios estudiantes, los cupos de sus clases se reducen proporcionalmente. Si algún curso del bundle se agota, se detiene su consumo para todos.

- **Finalización y redondeo**

- Al terminar la fase de consumo, se aplica una técnica de *rounding* para convertir las fracciones consumidas en asignaciones definitivas (por ejemplo, mediante el método de Birkhoff von Neumann).

Probabilistic Serial Rule (PS)

- **Inicialización**

- Para cada estudiante, se crea una lista vacía de asignaciones (inicialmente fraccional).
- Cada estudiante dispone de una lista ordenada de preferencias de asignaturas.
- Para cada asignatura, se conoce la capacidad (número de cupos disponibles).
- Se inicializa un diccionario de consumo $C[e, a] = 0$, donde e denota al estudiante y a a la asignatura.

- **Consumo continuo**

- Se simula un avance en el tiempo continuo. Al inicio, cada estudiante comienza a "consumir" su asignatura más preferida que aún disponga de cupos, a una tasa unitaria.
- Mientras la asignatura no se agote (es decir, mientras su capacidad sea mayor que cero), el estudiante sigue consumiéndola.
- Cuando una asignatura se consume completamente (cupos = 0), todos los estudiantes que la estaban consumiendo pasan a consumir su siguiente asignatura preferida con cupos disponibles.
- Durante este proceso se actualiza el diccionario de consumo, de modo que, para cada asignatura a , la suma de los consumos de todos los estudiantes satisface:

$$\sum_e C[e, a] = \text{capacidadInicial} : a.$$

- **Finalización y redondeo**

- Al finalizar el proceso, cada entrada $C[e, a]$ representa la fracción de la asignatura a "consumida" por el estudiante e . Dichas fracciones se interpretan como las probabilidades de asignación.
- Se aplicará una técnica de *rounding* (por ejemplo, mediante el método de Birkhoff von Neumann) para convertir la asignación fraccional en asignaciones definitivas, asignando a cada estudiante una o más asignaturas según se requiera.

Random Serial Dictatorship (RSD)

- **Inicialización**

- Para cada estudiante, se crea una lista vacía de asignaciones.
- Cada estudiante dispone de una lista ordenada de preferencias de asignaturas.
- Para cada asignatura, se conoce la capacidad inicial (número de cupos disponibles).

- **Determinación del orden aleatorio**

- Se genera una permutación aleatoria de todos los estudiantes, definiendo un orden secuencial.

- **Asignación secuencial**

- Siguiendo el orden aleatorio, cada estudiante elige la asignatura de mayor preferencia que aún disponga de cupos.

- Una vez seleccionada una asignatura, se asigna al estudiante y se reduce la capacidad de dicha asignatura en uno.
- Si la asignatura preferida del estudiante ya no tiene cupos disponibles, se evalúan sus siguientes preferencias, en orden, hasta encontrar una asignatura con cupos.

- **Finalización**

- El proceso se repite hasta que se hayan evaluado todos los estudiantes o hasta que ninguno pueda seleccionar una asignatura (por agotarse las capacidades).

5 Código Fuente

Aquí empieza el contenido del artículo de la sección 1 y también hay una referencia [1].

6 Manual Usuario

Datos de entrada

- Número de cursos
- Número de docentes, debe ser mayor o igual al número de cursos
- Número de aulas, debe ser mayor o igual al número de cursos
- Número de estudiantes
- Máximo número de materias deseadas de un estudiante (Predeterminado=9).
- Promedio de número de cupos de las clases (Predeterminado 40).
- Método de asignación (BPS, RSD, PBS).
- Si elige BPS, debe elegir, el número máximo de asignaturas por Bundle.

7 Manual Técnico

Aquí empieza el contenido del artículo de la sección 1 y también hay una referencia [1].

8 Experimentación

Aquí empieza el contenido del artículo de la sección 1 y también hay una referencia [1].

8.1 Análisis de resultados

8.1.1 Escenario 1

8.1.2 Escenario 2

8.1.3 Escenario 3

Tabla 2: Nivel de Agua en los Embalses

Fecha	Estado	Nivel de los embalses/Capacidad
Abril 2024	Inicio Racionamiento	14%
Junio 2024	Condiciones Actuales Escasez	30%
Ideal	Capacidad Completa	100%

Tabla 1: Ejemplo de tabla 4x3 con contorno marcado

Encabezado 1	Encabezado 2	Encabezado 3	Encabezado 4
Celda 1	Celda 2	Celda 3	Celda 4
Celda 5	Celda 6	Celda 7	Celda 8
Celda 9	Celda 10	Celda 11	Celda 12

Así mismo hay una tabla 1.

Tabla larga y personalizable en 3

Tabla 3: Análisis Stakeholders

Stakeholder	Necesidades	Intereses	Impacto
CD1	CD2	CD3	CD4
CD5	CD6	CD7	CD8

Y hay una figura 1.



Figura 1: Descripción de la figura.

Por último una lista:

- Item 1
- Item 2
- Item 3

Use la siguiente página para documentar las referencias en formato IEEE: <https://www.citethisforme.com/>

9 Referencias

- [1] D. N. de B. Universitario, “Dirección Nacional de Bienestar Universitario,” contador de visitas. [Online]. Available: <https://bienestar.unal.edu.co/index.php>. [Accessed: 05-Mar-2023].
- [2] A. Bogomolnaia and H. Moulin, “A new solution to the random assignment problem,” *Journal of Economic Theory*, vol. 100, no. 2, pp. 295–328, 2001, doi: 10.1006/jeth.2000.2710.