



PROGRAMA DE ASIGNATURA

Nombre del curso o actividad	OPTIMIZACIÓN AVANZADA DE PROCESOS
Nombre del programa	Doctorado en Ciencias de la Ingeniería mención Automática
Descripción	<i>Código: 12502, Nivel: 1, Créditos SCT-Chile: 10, Tipo: Obligatoria, Plan de Estudios: Resolución N°3618 del 15/07/25.</i> Este curso ofrece una introducción general a la modelación matemática en programación lineal, no lineal y problemas combinatorios. Los estudiantes aprenderán conceptos clave de optimización determinista y estocástica, tanto en problemas continuos como combinatoriales. Además, se explorarán aplicaciones en ingeniería y diversas técnicas de resolución, incluyendo algoritmos exactos y aproximados. El curso también enfatiza la implementación computacional de modelos y algoritmos mediante software especializado, permitiendo evaluar su efectividad. Se fomentará la exploración de nuevos enfoques algorítmicos y su aplicación en temas de tesis.
Objetivos	● Comprender conceptos básicos y avanzados sobre técnicas modernas de optimización lineal y no-lineal para problemas de naturaleza continua, convexa y combinatorial en su forma determinista y estocástica en el ámbito de la ingeniería. ● Modelar matemáticamente nuevas aplicaciones usando técnicas modernas de optimización determinista y estocástica. ● Explorar nuevas aplicaciones usando los modelos, métodos y algoritmos estudiados relacionados con los temas de tesis de cada estudiante en particular. ● Realizar implementaciones computacionales que permitan medir la efectividad de los algoritmos propuestos usando software de optimización especializado. Realizar un aporte personal científico y/o de aplicación práctica mediante la implementación de técnicas de optimización.
Contenidos	Unidad 1: Introducción a la programación matemática lineal, no lineal, convexa y combinatoria. ● Programación Lineal (PL) ○ Definición y formulación de problemas de PL. ○ Teoremas fundamentales: existencia de soluciones óptimas, dualidad y complementariedad. ○ Método Simplex. ○ Resolución con software (CPLEX, GLPK, Matlab, Gurobi). ● Programación No Lineal (PNL) ○ Diferencias entre programación lineal y no lineal. ○ Funciones objetivo y restricciones no lineales. ○ Condiciones de optimalidad de Karush-Kuhn-Tucker (KKT). ○ Métodos de solución: - Métodos de gradiente. - Métodos de Newton y cuasi-Newton. ● Optimización Convexa ○ Definición y propiedades de conjuntos y funciones convexas. ○ Programación convexa vs. no convexa. ○ Condiciones de optimalidad en problemas convexas. ● Optimización Combinatorial ○ Definición de problemas combinatoriales y su complejidad computacional. ○ Ejemplos clásicos:



- Problema del Viajante de Comercio (TSP).
- Problema de Asignación.
- Problema de la Mochila.
- o Métodos de solución:
 - Programación entera y mixta-entera.
 - Métodos exactos: Ramificación y Acotamiento (Branch and Bound), Benders.

Unidad 2: Modelos de optimización basados en optimización robusta y estocástica.

- Introducción a la Optimización Bajo Incertidumbre
- Diferencia entre optimización determinista, robusta y estocástica.
- Fuentes de incertidumbre en problemas de optimización.
- Aplicaciones en ingeniería, logística, redes.
- Optimización Robusta (RO - Robust Optimization)
 - o Definición y fundamentos de optimización robusta.
 - o Conjuntos de incertidumbre (poliedrales, elipsoidales, intersección de conjuntos).
- Formulación de modelos robustos:
 - o Programación Lineal Robusta (RLP).
 - o Programación Entera Robusta (RIP).
 - o Programación No Lineal Robusta (RNLP).
- Introducción a la modelación probabilística en optimización.
- Programación Estocástica con Restricciones de Probabilidad (Chance-Constrained Programming).
- Aplicaciones en asignación de recursos, diseño de redes y logística.

Unidad 3: Algoritmos exactos, heurísticos y de aproximación.

- Algoritmos Exactos
 - o Branch and Bound: Técnica basada en la exploración de un árbol de soluciones con poda de ramas no prometedoras.
 - o Branch and Cut: Extensión de Branch and Bound que incorpora cortes para mejorar la eficiencia en problemas de Programación Entera.
- Algoritmos Heurísticos aproximados
 - o Los heurísticos buscan soluciones buenas en tiempos razonables, sin garantizar optimalidad. Son útiles en problemas grandes o difíciles.
 - o Métodos Greedy (Voraces): Construcción de soluciones paso a paso, eligiendo la mejor opción localmente.
 - o Búsqueda Local: Mejoras iterativas a una solución inicial (Ejemplo: Hill Climbing, Simulated Annealing).
 - o Tabu Search: Extensión de la búsqueda local que evita ciclos mediante una memoria de soluciones visitadas.
 - o Algoritmos Genéticos: Inspirados en la evolución natural, combinan soluciones para mejorar resultados.

Unidad 4: Implementación de modelos matemáticos en software especializados.

- Introducción a la Implementación de Modelos Matemáticos
 - o Importancia de la implementación computacional en optimización.
 - o Diferencias entre formulación teórica y modelado computacional.
 - o Elección del software adecuado según el tipo de problema.
- Herramientas de Modelado y Software de Optimización



	○ Lenguajes de modelado: AMPL, GAMS, Pyomo, JuMP. ○ Solvers comerciales: Gurobi, CPLEX, MOSEK, FICO Xpress. ○ Solvers de código abierto: CBC, GLPK, SCIP. ○ Otros entornos: MATLAB (Optimization Toolbox), R (lpSolve, ompr), Python (PuLP, OR-Tools). ● Algoritmos Exactos y Aproximados en la Práctica ○ Implementación de algoritmos de Branch & Bound, Branch & Cut ○ Métodos heurísticos y metaheurísticos (ej. algoritmos genéticos, búsqueda tabú, recocido simulado). Unidad 5: Aplicación de conceptos teóricos y prácticos a problemáticas específicas relacionados con la tesis de los estudiantes. ● Modelación Matemática aplicada a problemas de tesis ○ Formulación de modelos matemáticos específicos (PL, PNL, enteros, estocásticos, etc.). ○ Elección de restricciones y funciones objetivo-relevantes. ● Métodos exactos y heurísticos en problemas específicos ○ Uso de heurísticas y metaheurísticas para problemas complejos (GRASP, Algoritmos Genéticos, Búsqueda Tabú). Análisis de enfoques exactos y aproximados en los problemas de tesis.
Modalidad de evaluación	Resultado de Aprendizaje General: Implementar modelos y algoritmos de optimización mediante software especializado, evaluando su efectividad en aplicaciones de ingeniería, para explorar enfoques algorítmicos innovadores en el contexto de sus investigaciones y trabajos de tesis. Resultados de Aprendizaje Específicos: ● Construir modelos matemáticos de programación matemática lineal, no lineal, convexa y combinatoria formulando adecuadamente los problemas y utilizando las técnicas apropiadas para representar situaciones reales en contextos de optimización. ● Formular modelos de optimización basados en optimización robusta y estocástica para la construcción de modelos que garanticen soluciones eficientes y viables en condiciones de incertidumbre. ● Aplicar distintos algoritmos exactos, heurísticos y de aproximación utilizando las estrategias más adecuadas según las características del problema de optimización, para obtener soluciones precisas, eficientes o cercanas a la óptima en tiempo razonable. ● Implementar modelos matemáticos y algoritmos en software especializados (Matlab, Python, GAMS, etc) desarrollando programas eficientes para la resolución de problemas de optimización, evaluando su efectividad y capacidad de adaptación a diferentes tipos de modelos y condiciones de incertidumbre. ● Aplicar conocimientos adquiridos a sus problemas de investigación relacionados con la tesis a desarrollar formulando y resolviendo modelos de optimización adecuados, utilizando algoritmos y software especializado, y proponiendo soluciones innovadoras a desafíos específicos en su área de estudio.
Resultados de aprendizajes esperados	Estrategias Metodológicas El aprendizaje se lleva a cabo mediante una metodología activa que integra teoría y práctica a través de simulaciones y ejercicios interactivos. Se emplean herramientas digitales para modelar escenarios reales y analizar distintas estrategias de solución. Durante las sesiones, los estudiantes participan en experimentos controlados y análisis de datos con software especializado, lo que facilita la comprensión y aplicación de los conceptos clave del curso.



	Evaluación Las evaluaciones se realizarán por medio de: • 2 trabajos de investigación –documento escrito, código fuente de la implementación y presentación- (corresponde al 50 % de la evaluación cada uno) • Trabajo adicional en caso de no cumplir con la expectativa de algún trabajo en modalidad POR.
Bibliografía	Básica: • Hillier, F. & Lieberman, G. (2015). <i>Introduction to Operations Research</i>. Versión en Español: <i>Introducción a la Investigación de Operaciones</i>. Mc Graw Hill. 9th edition • Luenberger, D. G. & Ye, Y. (2015). <i>Introduction to Linear and Nonlinear Programming</i>. Addison Wesley. Versión en Español. “<i>Programación Lineal y No-Lineal</i>”. Addison Wesley. 4th Edition. • Papadimitriou, C. H. (2003). <i>Computational complexity</i>. John Wiley and Sons Ltd. • Glover, F. & Kochenberger, G. (2003). <i>Handbook of Metaheuristics</i>. Kluwers International Series, New York. • Papadimitriou, C. H. & Steiglitz, K. (1998). <i>Combinatorial optimization: algorithms and complexity</i>. Courier Corporation. Recomendada: Literatura reciente consultada en bases de datos: IEEE Xplore, Springer, Science Direct, Hindawi, MDPI, etc.