

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SOLVING A HIGHLY-CONSTRAINED
BI-OBJECTIVE SHIFT SCHEDULING PROBLEM
USING A SPECIALIZED NSGA-II

Bryan Felipe González Ramírez

MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

ENERO 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Solving a Highly-Constrained Bi-objective Shift Scheduling Problem Using a Specialized NSGA-II

Nombre del candidato(a): Bryan Felipe González Ramírez

Carrera / Grado: Magister en Ciencias de la Ingeniería Informática

Campus: Casa Central Valparaiso ; **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Elizabeth Montero, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

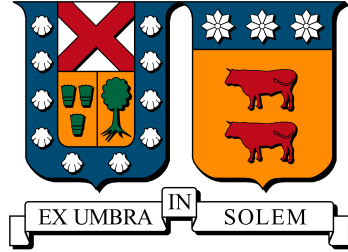
Fecha: 31/12/2025

Firma:

Estudiante o Candidato(a):

Fecha: 31/12/2025; **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA

**SOLVING A HIGHLY-CONSTRAINED
BI-OBJECTIVE SHIFT SCHEDULING
PROBLEM USING A SPECIALIZED NSGA-II**

Tesis de Grado presentada por
BRYAN FELIPE GONZÁLEZ RAMÍREZ

como requisito parcial para optar al grado de
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

PROFESOR GUÍA
ELIZABETH MONTERO

ENERO 2026

TÍTULO DE LA TESIS:

SOLVING A HIGHLY-CONSTRAINED BI-OBJECTIVE SHIFT SCHEDULING PROBLEM USING A SPECIALIZED NSGA-II

AUTOR:

BRYAN FELIPE GONZÁLEZ RAMÍREZ

Trabajo de Tesis, presentado en cumplimiento parcial de los requisitos para el Grado de **Magíster en Ciencias de la Ingeniería Informática** de la Universidad Técnica Federico Santa María.

Elizabeth Montero

Profesor Guía

María Cristina Riff

Profesor Correferente

Ignacio Araya

Profesor Externo

Enero 2026.
Santiago, Chile.

Agradecimientos

Quiero expresar mi sincero agradecimiento a la profesora Elizabeth Montero, profesora guía de esta tesis, por su apoyo constante, orientación académica y valiosas observaciones durante todo el desarrollo de este trabajo, las cuales fueron fundamentales para su calidad y rigor.

Agradezco especialmente a la profesora María Cristina Riff, quien fue una motivación clave para mi ingreso al programa y para el interés en la línea de investigación que dio origen a esta tesis, influyendo de manera significativa en mi formación académica.

Asimismo, agradezco al profesor Sergio Campos por su valiosa retroalimentación en la investigación base que motiva este trabajo, contribuyendo a fortalecer sus fundamentos y enfoque.

Este estudio fue financiado por el proyecto FONDECYT N° 1230365, cuyo apoyo permitió el desarrollo de la investigación. Adicionalmente, agradezco al Departamento de Informática y a la Dirección de Postgrado de la Universidad Técnica Federico Santa María por los fondos otorgados, obtenidos mediante concurso, que hicieron posible la presentación de esta tesis.

Finalmente, agradezco profundamente a mi familia y amigos por su apoyo, paciencia y comprensión a lo largo de este proceso. Asimismo, extendo mi gratitud a todas las personas que, desde el ámbito académico y personal, contribuyeron directa o indirectamente a la realización de este trabajo.

Resumen

La planificación de turnos laborales es un problema clave en la gestión de recursos humanos, especialmente en contextos con múltiples restricciones operativas y preferencias individuales. El *Employee Shift Scheduling Problem* (ESSP) presenta una elevada complejidad combinatoria y objetivos conflictivos, como el cumplimiento de la cobertura requerida y la satisfacción de los empleados. Tradicionalmente, estos aspectos han sido abordados mediante formulaciones mono-objetivo, lo que limita el análisis explícito de los compromisos entre bienestar laboral y eficiencia organizacional.

En este trabajo se aborda el ESSP desde una perspectiva de optimización bi-objetivo, separando la minimización de la insatisfacción de los empleados y la minimización de las penalizaciones por incumplimiento de cobertura. Para ello, se propone una adaptación especializada del algoritmo evolutivo NSGA-II, diseñada para operar en un espacio de búsqueda altamente restringido. La propuesta incorpora una representación híbrida basada en secuencias factibles, que permite manejar implícitamente un subconjunto relevante de restricciones y facilita el diseño de operadores genéticos especializados.

El algoritmo incluye una fase de construcción inicial orientada a favorecer la factibilidad y una parametrización automática mediante ParamILS. La validación experimental se realiza sobre diez instancias de referencia del estado del arte, comparando los resultados con un método exacto basado en programación lineal entera (AMPL/Gurobi). La evaluación se lleva a cabo mediante el indicador de hipervolumen y el análisis de frentes de Pareto. Los resultados muestran que el enfoque propuesto genera soluciones competitivas, especialmente en instancias de mayor tamaño, y evidencian que las características estructurales del problema influyen de manera significativa en el desempeño algorítmico.

Abstract

Workforce shift scheduling is a central problem in human resource management, particularly in contexts characterized by multiple operational constraints and individual preferences. The *Employee Shift Scheduling Problem* (ESSP) exhibits high combinatorial complexity and inherently conflicting objectives, such as meeting coverage requirements while preserving employee satisfaction. Traditional approaches have typically addressed these aspects through single-objective formulations, which limit the explicit analysis of trade-offs between workforce well-being and operational efficiency.

This thesis addresses the ESSP from a bi-objective optimization perspective, explicitly separating the minimization of employee dissatisfaction from the minimization of penalties associated with coverage deviations. To this end, a specialized adaptation of the evolutionary algorithm NSGA-II is proposed, specifically designed to operate within a highly constrained search space. The approach incorporates a hybrid representation based on feasible work sequences, allowing a relevant subset of constraints to be handled implicitly and enabling the design of problem-specific genetic operators.

The algorithm includes a feasibility-oriented initialization phase and an automatic parameter tuning process using ParamILS. Experimental validation is conducted on ten benchmark instances from the state of the art, with results compared against an exact method based on integer linear programming implemented in AMPL and solved with Gurobi. Performance is evaluated using the hypervolume indicator and Pareto front analysis. The results show that the proposed approach is capable of generating competitive solution sets, particularly for larger instances, and highlight the significant influence of instance structural characteristics on algorithmic performance.

Índice de Contenidos

Agradecimientos	III
Resumen	IV
Abstract	V
Índice de Contenidos	VI
Lista de Tablas	X
Lista de Algoritmos	XI
Lista de Figuras	XIII
1. Introducción	1
1.1. Hipótesis y Objetivos	3
1.1.1. Objetivo general	3
1.1.2. Objetivos específicos	3
1.2. Estructura de la tesis	4
2. Marco Teórico	6
2.1. Enfoques de Objetivo Único	6

2.1.1.	Mejores resultados a la fecha en la versión mono-objetivo del ESSP	9
2.1.2.	Otras técnicas relevantes	11
2.2.	Enfoques Multiobjetivo	12
2.2.1.	Principales objetivos considerados	12
2.2.2.	Metodologías empleadas	13
3.	Employee Shift Scheduling Problem	15
3.1.	Motivación y objetivos	15
3.2.	Problemas relacionados	16
3.3.	Modelamiento del problema	16
3.3.1.	Enfoque Multiobjetivo	17
4.	Modelo Matemático	18
5.	Algoritmos y propuesta de solución	23
5.1.	Acercamiento propuesto	24
5.2.	Adaptaciones realizadas	25
5.3.	Representación	26
5.3.1.	Nivel genotípico.	26
5.3.2.	Nivel fenotípico.	27
5.3.3.	Motivación y ventajas de la representación híbrida.	27
5.3.4.	Ejemplo ilustrativo.	28
5.4.	Evaluación de soluciones	28
5.4.1.	Funciones objetivo	29
5.4.2.	Manejo de restricciones	29
5.5.	Estructura general	30
5.6.	Operadores	31

5.6.1.	Mutación	31
5.6.2.	Crossover operators	37
5.7.	Fase de Construcción	39
5.7.1.	Sequene Gather	39
5.7.2.	Feasible Employees	40
5.7.3.	Population Construction	41
5.8.	Parámetros	42
6.	Experimentos	44
6.1.	Experimentos	44
6.1.1.	Configuración Experimental	44
6.1.2.	Descripción de las Instancias	45
6.1.3.	Análisis de características de las instancias	46
6.1.4.	Metodología de evaluación	53
7.	Resultados y análisis	57
7.1.	Resultados de la parametrización	57
7.1.1.	Correlación de características y valores de parámetros	59
7.2.	Resultados del método exacto	62
7.2.1.	Resumen de resultados del método exacto (AMPL)	62
7.2.2.	Cálculo del indicador Hipervolumen	63
7.3.	Resultados de NSGA-II	65
7.3.1.	Comparación de frentes de Pareto	65
7.3.2.	Visualización de frentes de Pareto representativos	67
8.	Conclusiones	73
8.1.	Alcances y capacidades demostradas de la solución propuesta	73

8.2. Limitaciones identificadas y análisis crítico de escalabilidad	75
8.3. Validación de Hipótesis, Objetivos y Resultados Clave	76
8.3.1. Validación de la hipótesis	76
8.3.2. Validación de los objetivos	76
8.4. Contribuciones principales y aportes innovadores	77
8.5. Impacto y aplicaciones prácticas	78
9. Trabajo futuro	81
9.1. Síntesis y Reflexiones Finales	82
Bibliografía	84

Lista de Tablas

5.1. Resumen de parámetros del algoritmo	43
6.1. Instancias de pruebas	45
6.2. Características de las instancias	52
6.3. Ponderaciones para el método exacto	53
6.4. Espacio de parámetros definido para ParamILS, donde los valores posibles son valores discretos, y el valor por defecto es el valor con el que se comienza la parametrización. para más información sobre cada parámetro revisar la tabla de parámetros 5.1	55
7.1. Configuraciones finales obtenidas mediante ParamILS para las 10 instancias. Se reportan los valores reales ponderados utilizados por el algoritmo.	58
7.2. Resumen de resultados del método exacto (AMPL), incluyendo puntos de referencia y número de soluciones no dominadas.	64
7.3. Comparación de resultados entre NSGA-II y método exacto (AMPL) para las primeras 10 instancias. Gap HV: diferencia porcentual en hipervolumen $((NSGA-II - AMPL)/AMPL \times 100)$; valores positivos indican superioridad de NSGA-II. Ratio Tiempo: factor de diferencia en tiempo de ejecución $(NSGA-II/AMPL)$	66

Lista de Algoritmos

- 1. *Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)*. 30
- 2. Sequence Gather Phase 40
- 3. Feasible Employee Construction 41
- 4. Population Initialization 42

Lista de Figuras

5.1. Proceso de selección en NSGA-II	25
5.2. Ejemplo de representación genotípica y fenotípica de soluciones, usando 1 como turno trabajado y 0 como día libre.	28
5.3. Sequence Add mutation. Una secuencia precomputada es insertada para el empleado e_1 , reemplazando bloques solapados. La primera secuencia es {1,2}, reemplazada por {3,1}.	32
5.4. Replace Sequence mutation. La segunda secuencia de e_1 es reemplazada por otra de igual largo para mejorar preferencias y coberturas.	33
5.5. Shift mutation aplicada a la secuencia de un solo empleado. El bloque rojo se desplaza a una nueva posición factible (verde), mejorando la cobertura o la calidad de la preferencia.	34
5.6. Adaptive replace mutation. La secuencia original (roja) se sustituye por una secuencia más corta y mejor (verde) para la cobertura o las preferencias. . .	35
5.7. Replace employee mutation. Todo el horario del empleado e_i (rojo) se sustituye por otro empleado factible e_j (verde), lo que aumenta el potencial de exploración.	36
5.8. Employee crossover: complete employee schedules are inherited by alternating between parents. Colors indicate offspring.	37

5.9. Sequence crossover: Los hijos heredan secuencias factibles completas de ambos padres, preservando la factibilidad en su mayoría, pero agregando exploración.	38
7.1. Matriz de correlación de magnitud absoluta ($ \rho $) entre parámetros estructurales, parámetros del algoritmo y características de las instancias. Los parámetros pc1 y pc2 no presentan valores de correlación debido a que permanecen constantes en todas las instancias, por lo que no inducen variabilidad estadísticamente relevante.	60
7.2. Frentes de Pareto instancias 1-4	69
7.3. Frentes de Pareto instancias 5-8	70
7.4. Frentes de Pareto instancias 9-10	71

Capítulo 1

Introducción

La gestión eficiente de turnos laborales constituye un desafío crítico en la organización moderna de empresas, impactando directamente en la productividad operacional, la satisfacción del personal y el cumplimiento normativo. El *Employee Shift Scheduling Problem* (ESSP) aborda esta complejidad mediante la asignación óptima de turnos a empleados, equilibrando objetivos frecuentemente contrapuestos: la maximización del bienestar del trabajador y la optimización de la eficiencia empresarial. Tradicionalmente, los enfoques unidimensionales han predominado en la literatura, priorizando ya sea la minimización de costos o la satisfacción de preferencias individuales, pero rara vez integrando ambos aspectos de forma simultánea. Esta limitación motiva la exploración de estrategias multiobjetivo que capturen la naturaleza inherentemente multifacética del problema. En este trabajo se aborda el ESSP desde una perspectiva de optimización multiobjetivo, considerando explícitamente dos metas conflictivas: (1) minimizar la insatisfacción de los empleados al asignar turnos que contravienen sus preferencias, restricciones personales y límites laborales; y (2) minimizar las penalizaciones por incumplimiento de la cobertura operacional requerida, ya sea por déficit o exceso de personal en turnos específicos. Para resolver este problema, se propone una adaptación del algoritmo evolutivo *Non-dominated Sorting Genetic Algorithm II* (NSGA-II), seleccionado por su eficacia en espacios de búsqueda combinatorios y su capacidad para generar frentes de Pareto diversificados.

La principal dificultad del ESSP no solo radica en la gran cantidad de restricciones involucradas, sino en su alto nivel de interdependencia, lo que genera un espacio factible extremadamente reducido y fragmentado. Restricciones relacionadas con días libres consecutivos, máximos fines de semana trabajados y máxima carga de trabajo interactúan de manera no lineal, provocando que modificaciones locales en una solución frecuentemente desencadenen múltiples violaciones simultáneas. Esta característica vuelve el espacio de búsqueda particularmente cerrado y propenso a la aparición de óptimos locales, dificultando la exploración efectiva mediante operadores genéricos. Con el objetivo de mitigar estas dificultades, se propone una representación basada en secuencias junto con un conjunto de subsecuencias factibles, que permite gestionar una parte significativa de las restricciones de manera implícita. Esta estrategia reduce la carga de evaluación de factibilidad y habilita el diseño de operadores especializados que aprovechan la estructura restringida del problema para recorrer el espacio de soluciones de forma más controlada y eficiente.

Un componente clave del estudio es el análisis de características y complejidad de las instancias, que busca identificar cómo las propiedades estructurales —como la densidad de cobertura, la heterogeneidad de las preferencias y la conflictividad entre restricciones— influyen en la dificultad de resolución del problema. Este objetivo de este análisis es anticipar el comportamiento esperado de los algoritmos y fundamentar comparaciones entre instancias de diferente escala y naturaleza.

La metodología de validación integra dos enfoques complementarios: un método exacto basado en programación lineal entera (implementado con AMPL/Gurobi) y la metaheurística NSGA-II parametrizada mediante ParamILS. Este diseño experimental permite evaluar la calidad de las soluciones mediante el indicador de hipervolumen, comparar frentes de Pareto y analizar la influencia de las características de las instancias sobre el rendimiento algorítmico. Las pruebas se realizaron sobre 10 instancias reales utilizadas ampliamente en el estado del arte[1], que abarcan desde escenarios reducidos (8 empleados y 14 días) hasta configuraciones complejas (40 empleados y 28 días).

1.1. Hipótesis y Objetivos

Para el presente trabajo se plantea la siguiente hipótesis de investigación:

- La mejora del algoritmo NSGA-II mediante la incorporación de representaciones específicas del problema, operadores de mutación, estrategias de inicialización y principios de generación de columnas permite mejorar la calidad, diversidad y factibilidad de las soluciones para el Problema de Programación de Turnos de Empleados (ESSP) bi-objetivo altamente restringido, particularmente en instancias de pequeña y mediana escala.

1.1.1. Objetivo general

Desarrollar y evaluar un algoritmo NSGA-II especializado para resolver el Employee Shift Scheduling Problem altamente restringido bajo un enfoque bi-objetivo, orientado a minimizar simultáneamente la insatisfacción de los empleados y las penalizaciones por incumplimiento de cobertura operacional.

1.1.2. Objetivos específicos

Para cumplir con el objetivo general, se definen los siguientes objetivos específicos:

- Formular el ESSP como un problema de optimización bi-objetivo, separando explícitamente las penalizaciones asociadas a preferencias individuales y a requerimientos de cobertura.
- Diseñar una representación de soluciones y operadores genéticos adaptados a la estructura discreta y altamente restringida del ESSP, incorporando estrategias híbridas de inicialización basadas en enfoques *Greedy* e *Iterated Local Search*.

- Parametrizar automáticamente el algoritmo NSGA-II mediante ParamILS, identificando configuraciones específicas que maximicen su desempeño para distintas instancias del problema.
- Caracterizar las instancias del ESSP mediante indicadores estructurales y de complejidad, analizando su relación con la dificultad de resolución y el comportamiento del algoritmo propuesto.
- Validar la calidad de las soluciones obtenidas comparando el enfoque propuesto con un método exacto basado en programación lineal entera (AMPL/Gurobi), utilizando indicadores de desempeño como hipervolumen y análisis de frentes de Pareto.

1.2. Estructura de la tesis

Este documento se estructura en cinco capítulos principales, precedidos por elementos preliminares y seguidos por las referencias bibliográficas:

- **Introducción:** Presenta la motivación, objetivos y alcance del trabajo, destacando la importancia de abordar el ESSP (Employee Shift Scheduling Problem) como un problema multiobjetivo.
- **Definición del Problema:** Contextualiza el ESSP, describe variantes relevantes y desarrolla el modelamiento matemático, detallando el enfoque multiobjetivo adoptado y el modelo propuesto.
- **Marco Conceptual:** Revisa enfoques previos de la literatura, incluyendo métodos exactos, heurísticos y metaheurísticos tanto de objetivo único como multiobjetivo, así como las principales metodologías y retos identificados.
- **Propuesta de Solución:** Explica en detalle la adaptación del algoritmo NSGA-II al ESSP, abordando representación, operadores genéticos, evaluación de soluciones y la estrategia híbrida de construcción de soluciones iniciales.

- **Validación de la Solución:** Describe la experimentación realizada, incluyendo la caracterización de instancias, configuración de parámetros, comparación con un método exacto (AMPL + Gurobi) y análisis de los frentes de Pareto obtenidos.
- **Conclusiones:** Resume los alcances, limitaciones, contribuciones del trabajo y presenta recomendaciones para líneas futuras de investigación.

Capítulo 2

Marco Teórico

2.1. Enfoques de Objetivo Único

Los métodos tradicionales para abordar el Problema de Programación de Turnos de Empleados (ESSP) se han centrado en optimizar un único objetivo, ya sea minimizar costos operativos o maximizar indicadores de satisfacción laboral. Estos enfoques, aunque limitados en su alcance multicriterio, han sentado las bases para el desarrollo de técnicas especializadas adaptadas a contextos específicos. A continuación, se detallan tres categorías fundamentales:

- **Modelos Exactos: Programación Lineal Entera (PLE)**

Como plantea [2], los modelos exactos buscan soluciones óptimas mediante formulaciones matemáticas rigurosas. Un ejemplo clásico es la programación lineal entera mixta, que integra funciones objetivo lineales con restricciones discretas (e.g., disponibilidad de empleados, límites de horas). Estos modelos son resolubles mediante *solvers* comerciales (e.g., CPLEX, Gurobi) que realizan búsquedas exhaustivas en el espacio de soluciones. Si bien garantizan optimalidad, su escalabilidad es limitada: en problemas con alta combinatoria (e.g., más de 50 empleados y 100 turnos semanales), el tiempo computacional crece exponencialmente. Por ejemplo, en [2], la resolución de

instancias con 40 trabajadores y 128 horas operativas requirió relajar restricciones para evitar explosiones combinatorias. Su aplicación se restringe a contextos con datos determinísticos y recursos computacionales suficientes.

Además, investigaciones como [3, 4] han demostrado que las formulaciones de PLE pueden incorporar múltiples habilidades laborales y períodos rotacionales, siempre que la estructura combinatoria se mantenga controlada. Incluso, enfoques basados en column generation y técnicas de descomposición [5] han permitido abordar instancias de mayor escala reduciendo el número de variables activas durante la resolución. Sin embargo, estas mejoras requieren un diseño experto y alta disponibilidad computacional, lo cual sigue limitando su aplicabilidad industrial.

- **Métodos Heurísticos: Adaptación a Complejidad Práctica** Cuando la complejidad del ESSP supera las capacidades de los métodos exactos, las heurísticas ofrecen soluciones subóptimas pero viables. [6] propone un enfoque en dos fases para programación de turnos en restaurantes de comida rápida: una fase de *construcción* identifica horas críticas y asigna trabajadores mediante reglas de prioridad (e.g., minimizar sobrestaffing, ajustar horas objetivo), mientras una fase de *mejora* redistribuye tareas para reducir desviaciones. Este método, probado en problemas con 40 empleados y 8 tareas heterogéneas, logró reducir el sobrestaffing en 71 % frente a métodos manuales, con tiempos de ejecución inferiores a 15 minutos. Las heurísticas son ideales para entornos dinámicos con restricciones no lineales (e.g., disponibilidad variable, habilidades múltiples), aunque carecen de garantías teóricas de optimalidad.

Las heurísticas han evolucionado hacia metaheurísticas bioinspiradas, como búsqueda tabú, recocido simulado y algoritmos evolutivos. Por ejemplo, [7] presentan una combinación de heurísticas constructivas con mecanismos de perturbación para escapar de óptimos locales, mientras que [8] incorporan perfiles de carga laboral para mantener estabilidad entre semanas de planificación. El objetivo central es lograr robustez y rapidez en contextos cambiantes, algo especialmente valioso en servicios 24/7 como salud y transporte.

- **Modelos Estocásticos: Incertidumbre en Demanda** En contextos donde la demanda

laboral es volátil (e.g., hospitales, retail), los modelos estocásticos como el de [9] incorporan distribuciones probabilísticas para manejar incertidumbre. Este trabajo formula un problema no lineal mixto estocástico que integra cuatro costos clave: contratación, despido, escasez y excedente laboral. La demanda se modela con funciones de densidad truncadas (e.g., uniforme, normal), ajustadas a límites operativos. Estos modelos son críticos en industrias con fluctuaciones impredecibles, aunque requieren datos históricos robustos para calibrar distribuciones.

Posteriormente, [10] incorporaron escenarios múltiples para capturar perturbaciones súbitas en la demanda (por ej. picos hospitalarios), integrando análisis de riesgo para penalizar desviaciones respecto a niveles seguros de dotación. Los avances recientes en programación robusta [11] han permitido considerar incertidumbre sin depender por completo de datos históricos, ampliando la aplicabilidad práctica de estos modelos.

Comparativa y Limitaciones

Cada enfoque tiene nichos de aplicación: los modelos exactos son preferibles en problemas pequeños y estructurados; las heurísticas, en escalabilidad y adaptabilidad; los estocásticos, en entornos inciertos. No obstante, su enfoque unidimensional ignora sinergias entre objetivos conflictivos (e.g., costos vs. satisfacción). Por ejemplo, [6] minimiza sobrestaffing pero no cuantifica el impacto en el estrés laboral, mientras [9] optimiza costos sin considerar preferencias de turnos. Esta limitación motiva el surgimiento de enfoques multiobjetivo, que integran trade-offs mediante técnicas de Pareto o programación por metas.

Otros trabajos, como [12], destacan que los enfoques tradicionalmente mono-objetivo tienden a degradar aspectos críticos del personal, como rotación equilibrada, cumplimiento de preferencias o mitigación de fatiga. En consecuencia, se ha observado un creciente interés en modelos éticos o centrados en bienestar, sentando así un puente hacia los métodos multiobjetivo que se discuten posteriormente en este estudio.

Si bien estos enfoques han demostrado ser útiles en ciertos contextos, a menudo no capturan toda la complejidad del problema real, ya que se centran en un solo objetivo y pueden pasar por alto otros factores importantes.

2.1.1. Mejores resultados a la fecha en la versión mono-objetivo del ESSP

En los últimos años, diversos enfoques han alcanzado resultados sobresalientes para la versión mono-objetivo del problema ESSP, siendo estos los métodos que han conseguido los límites inferiores de las instancias que serán utilizadas en esta investigación. Un rasgo común en estos trabajos es la combinación o integración de técnicas exactas y metaheurísticas, la reformulación de restricciones complejas y la exploración de modelos alternativos para mejorar la calidad y el tiempo de cómputo de las soluciones.

En primer lugar, [13] proponen un *algoritmo híbrido* que integra la programación entera (IP) y la búsqueda por entornos variables (VNS). Tras generar una solución inicial mediante una heurística voraz, el método aplica un descenso por entornos variables para refinarla, y luego incorpora IP en un esquema de *ruin-and-recreate* para diversificar la búsqueda. Finalmente, la IP se vuelve a utilizar para ajustar los resultados en el tiempo restante, logrando así superar a dos algoritmos del estado del arte y al solver comercial Gurobi en la mayoría de las instancias evaluadas. Este trabajo destaca por demostrar cómo las técnicas exactas pueden intensificar la búsqueda incluso dentro de un marco predominantemente metaheurístico.

Posteriormente, [14] abordan el desafío que presentan las restricciones sobre el número de asignaciones consecutivas, conocidas por dificultar considerablemente la resolución del problema. Su propuesta consiste en una *reformulación* de estas restricciones como subproblemas con una estructura interna eficiente. Esta reformulación, integrada dentro de un nuevo modelo de programación entera, permite reducir significativamente los tiempos de cómputo para alcanzar soluciones óptimas con solvers estándares. Así, mientras el trabajo de [13] se enfoca en la hibridación para explorar mejor el espacio de soluciones, el de [14] se centra en mejorar la estructura del modelo para facilitar su resolución exacta.

Más adelante, [15] exploran una línea distinta: modelar el ESSP como un problema de *max-SAT ponderado parcial*, aprovechando los avances en satisfacibilidad booleana. Este enfoque permite formular restricciones como cardinalidades y preferencias de forma natural, y facilita el uso de potentes solvers de SAT. Los autores comparan distintas codificaciones de restricciones y evalúan su desempeño frente a solvers líderes, además de contribuir con nuevas instancias públicas para la comunidad. Esta propuesta conecta conceptualmente con los

trabajos anteriores en el sentido de que también busca una representación más eficiente del problema, pero desde una perspectiva lógica más que matemática.

Además, [12] demuestran que la integración de *column generation* dentro de una búsqueda local basada en vecindarios mejora de forma sustancial la exploración del espacio de soluciones para instancias de gran tamaño, consiguiendo resultados de referencia con tiempos de cómputo competitivos.

Por otra parte, [11] presentan una síntesis exhaustiva de modelos y técnicas exactas aplicadas al ESSP, destacando que la reformulación de restricciones estructurales (como descansos y secuencias de trabajo) es clave para mejorar la eficiencia resolutive en modelos mono-objetivo tradicionales.

Finalmente, [10] proponen una estrategia de *column generation* con preferencias ponderadas que logra soluciones de alta calidad en entornos hospitalarios complejos, mostrando que incluso en formulaciones mono-objetivo es posible integrar aspectos de satisfacción manteniendo competitividad frente a solvers comerciales.

En línea con lo anterior, [1] introducen en 2020 un *método heurístico basado en column generation*, orientado a resolver instancias de gran tamaño para las que los métodos exactos tradicionales no son prácticos. A diferencia del branch-and-price completo, esta estrategia realiza una búsqueda incompleta que prioriza soluciones buenas en tiempos reducidos, utilizando un solver de programación lineal de primer orden rápido y de baja precisión. El subproblema de precios se modela como un problema de camino más corto con recursos, resolviéndose de manera eficiente gracias a su simplicidad estructural. Este método logra obtener las mejores soluciones conocidas en grandes instancias donde incluso solvers comerciales fallan.

De esto, podemos observar la tendencia actual en el ESSP mono-objetivo: por un lado, la hibridación entre métodos exactos y heurísticos para aprovechar sus fortalezas; por otro, la reformulación inteligente de restricciones y la exploración de modelos alternativos como maxSAT o *column generation* para afrontar la complejidad del problema.

2.1.2. Otras técnicas relevantes

Además de los métodos mencionados, los estudios utilizan herramientas complementarias para mejorar la toma de decisiones:

- Técnicas de priorización como AHP (Proceso Analítico Jerárquico), que permiten ordenar objetivos según su importancia relativa.
- Transformación de restricciones complejas en versiones más simples, lo que facilita su implementación computacional.
- Indicadores de desempeño, como índices de equidad o compatibilidad, que permiten evaluar cuán balanceadas son las soluciones propuestas.

Retos y tendencias

Los estudios analizados revelan tres desafíos principales:

Complejidad computacional: Modelos con más de 5 000 variables binarias (ej. 5 068 en [16]) requieren algoritmos paralelos o técnicas de descomposición. El caso de [17] con 22 enfermeras y 31 días generó 26 soluciones no dominadas en 12.05 minutos, demostrando escalabilidad.

Personalización: Las regulaciones laborales varían geográficamente. Por ejemplo, [17] adaptó restricciones específicas de un hospital iraní, como máximo 9 turnos nocturnos/mes.

Integración de factores humanos cualitativos: La cuantificación de DMS mediante cuestionarios (ej. Driver et al., 1998) añade capas de subjetividad que requieren validación empírica.

Como tendencias emergentes, se destacan:

Datos biométricos: Uso de wearables para monitorear fatiga en tiempo real y ajustar dinámicamente los modelos.

Compatibilidad predictiva: Aplicación de aprendizaje automático para predecir conflictos

en equipos basados en DMS históricos.

Resiliencia operativa: Extender los modelos para manejar ausentismos y emergencias mediante programación robusta.

2.2. Enfoques Multiobjetivo

Los enfoques multiobjetivo han surgido como una evolución natural frente a las limitaciones de los modelos centrados en un solo criterio. En el contexto de la planificación de turnos del personal de salud, estos métodos permiten abordar de forma simultánea múltiples necesidades que suelen entrar en conflicto, como la eficiencia operativa, la equidad en la asignación de turnos, el bienestar del personal y la estabilidad en la cobertura de servicios. A diferencia de los modelos tradicionales que buscan una única solución óptima, los enfoques multiobjetivo generan un conjunto de soluciones alternativas, conocidas como soluciones no dominadas o de Pareto, que ofrecen distintos balances entre los objetivos planteados. Esta capacidad de representar trade-offs ha llevado a un creciente interés por su aplicación en contextos complejos y sensibles[18, 19], principalmente en el ámbito hospitalario, tal como se ha documentado en estudios recientes[13, 16, 17] que intentan resolver el Nurse Rostering Problem, un problema hermano del Problema de Programación de Turnos de Empleados (ESSP).

2.2.1. Principales objetivos considerados

Los trabajos revisados identifican hasta siete dimensiones clave que suelen incluirse en los modelos multiobjetivo[12, 20]:

- Reducción de costos operativos, como exceso o falta de personal.
- Prevención de la fatiga acumulada, considerando la intensidad y duración de los turnos.
- Equidad en la asignación de turnos, evitando que ciertos empleados reciban siempre los horarios más difíciles.

- Satisfacción de preferencias individuales, como solicitudes de días libres.
- Cumplimiento de normas laborales, incluyendo límites de horas y proporciones mínimas de personal calificado.
- Optimización del uso de habilidades, asignando tareas complejas al personal más capacitado.
- Compatibilidad entre trabajadores, considerando factores como estilos de trabajo o toma de decisiones.

2.2.2. Metodologías empleadas

Diversos estudios han propuesto soluciones innovadoras para abordar estos objetivos [21, 22]. Un enfoque ampliamente utilizado corresponde a los algoritmos evolutivos, los cuales permiten equilibrar distintos criterios mediante la exploración simultánea del espacio de soluciones [23, 24]. En uno de estos trabajos, se incorporaron niveles de descanso como una variable clave, logrando una disminución significativa de la fatiga acumulada del personal [13].

Otra línea de investigación combina técnicas de programación matemática con esquemas de priorización de objetivos mediante pesos, permitiendo considerar preferencias del personal sin descuidar la cobertura de turnos. Este tipo de estrategias ha demostrado mejorar notablemente el cumplimiento de solicitudes individuales y reducir cargas excesivas en horarios desfavorables [16, 20].

Asimismo, la integración de factores humanos se ha convertido en un aspecto creciente dentro de los modelos multiobjetivo. Por ejemplo, [17] incorporan medidas de compatibilidad entre trabajadores para reducir conflictos operativos y mejorar la cooperación en equipos. La validez de estas soluciones se evalúa mediante indicadores que equilibran costos operativos, satisfacción laboral y estabilidad de los equipos.

Finalmente, los enfoques basados en descomposición también han cobrado relevancia, como el uso de MOEA/D para resolver versiones simplificadas del problema de asignación de turnos [25]. Aunque dicho enfoque considera un menor número de restricciones —como asignar

a cada trabajador solo un turno por día y limitar los días consecutivos trabajados—, permite estudiar compromisos similares a los abordados en esta investigación.

Capítulo 3

Employee Shift Scheduling Problem

3.1. Motivación y objetivos

La gestión eficaz de los turnos de los empleados es fundamental para el buen funcionamiento de cualquier empresa, ya que afecta directamente a la productividad, la satisfacción de los empleados y el cumplimiento de las regulaciones laborales. Los recursos humanos desempeñan un papel crucial en esta tarea, ya que son responsables de diseñar políticas y procedimientos que garanticen una asignación equitativa y eficiente de los turnos. Sin embargo, esta tarea ha sido históricamente propensa a generar conflictos y descontento entre los trabajadores, ya que no siempre es posible satisfacer a todos al mismo tiempo. En términos generales, este desafío se enmarca en lo que se conoce como el Problema de Programación de Turnos de Empleados, o Employee Shift Scheduling Problem (ESSP).

El problema se enfoca en la asignación efectiva de turnos de trabajo a los empleados en un período de tiempo determinado. En su esencia, este problema consiste en programar los horarios laborales de un grupo de empleados de manera que se cumplan diversas restricciones y preferencias, buscando optimizar la eficiencia operativa y la satisfacción de los trabajadores. El objetivo fundamental del ESSP es lograr una asignación óptima de empleados a diferentes tipos de turnos, minimizando al mismo tiempo las penalizaciones asociadas con la no conformidad a ciertas restricciones.

3.2. Problemas relacionados

ESSP es un problema que puede tener muchas variantes, siendo una de las más conocidas, General Employee Shift Scheduling Problem [26]. Su fama se debe a que es una versión mucho más general del problema, permitiendo personalizarlo a molde de cada organización; sin embargo, no deja de ser complejo, tomando como parámetros, por ejemplo, las habilidades de cada empleado.

Existen múltiples variantes, por ejemplo en [21] se cambian ciertos detalles que permiten especificar el problema. Dentro de las principales variantes tenemos el Problema de Programación de Turnos con Demanda Estocástica (Stochastic Demand Employee Scheduling Problem) [9], que tiene en cuenta la incertidumbre en la demanda de empleados en diferentes turnos. Además, el Problema de Ruteo de Vehículos con Restricciones de Turno (Vehicle Routing Problem with Time Windows) [27] está relacionado en situaciones en las que los empleados deben realizar tareas de servicio en ubicaciones específicas durante sus turnos.

3.3. Modelamiento del problema

Los principales parámetros considerados para modelar este problema incluyen las preferencias de los empleados, las preferencias de carga de trabajo de los empleados (por ejemplo, existen empleados que prefieren trabajar mucho durante varios días y tener más días libres), las preferencias de tipos de turno por parte de los empleados, limitaciones legales o por políticas de la empresa en cuanto a días de trabajo consecutivos, solicitudes de días libres por parte de empleados y requisitos de la empresa en cuanto a cobertura los fines de semana. Estas preferencias son factores importantes que determinan la satisfacción de los empleados y deben ser consideradas al momento de realizar la programación. Algunos aspectos relevantes del problema a resolver se listan a continuación:

- Considerar las preferencias de los empleados, se refiere a identificar las horas o turnos específicos que cada empleado prefiere trabajar. Por ejemplo, algunos empleados pueden preferir los turnos diurnos, mientras que otros pueden preferir los turnos nocturnos.

- Las preferencias de carga de trabajo se refieren a la cantidad de trabajo que un empleado prefiere tener en su horario. Algunos pueden preferir trabajar más horas en un día y tener más días de descanso, mientras que otros pueden preferir horarios más equilibrados.
- Las preferencias de tipos de turno se refieren a los tipos específicos de turnos que los empleados prefieren. Esto puede incluir turnos de mañana, tarde, noche o cualquier otro tipo específico que esté disponible.
- Las limitaciones de días de trabajo consecutivos se refieren a cuántos días seguidos un empleado está dispuesto a trabajar sin un día libre. Algunos empleados pueden preferir no trabajar más de cierta cantidad de días seguidos.
- Las solicitudes de días libres se refieren a los días específicos en los que un empleado solicita tener libre debido a compromisos personales o preferencias.
- Los requisitos de fines de semana se refieren a la preferencia de algunos empleados de trabajar o no trabajar los fines de semana.

3.3.1. Enfoque Multiobjetivo

Además de los objetivos tradicionales de minimizar costos y maximizar la satisfacción de los empleados, ha surgido un creciente interés en abordar el ESSP desde una perspectiva multiobjetivo. Los enfoques multiobjetivo, documentados en la literatura especializada, buscan optimizar varios objetivos conflictivos, tales como minimizar la incompatibilidad entre los empleados asignados a un mismo turno (considerando estilos de toma de decisiones y habilidades) como en la investigación de [17], reducir la fatiga asociada a los turnos de trabajo como en la investigación de [13], especialmente en entornos estresantes, maximizar la productividad considerando las habilidades de los empleados para realizar distintas tareas, y promover la igualdad en la distribución de la carga de trabajo entre los empleados. Estos enfoques buscan encontrar soluciones compromiso que satisfagan simultáneamente múltiples objetivos, en lugar de optimizar un solo objetivo. Esta estrategia puede resultar en programaciones más realistas y satisfactorias para todas las partes involucradas.

Capítulo 4

Modelo Matemático

El modelo a trabajar está basado en la investigación de [1], sin embargo, se considerará un enfoque multiobjetivo. En el trabajo previo, la penalización por no cumplir con la satisfacción de los empleados y la penalización por no cubrir/cubrir en exceso los turnos fueron tratados como una única función objetivo. En este modelo, se tratan como objetivos separados, permitiendo mayor versatilidad en cuanto a la priorización de un objetivo sobre otro, y a su vez permite encontrar soluciones alternativas.

Parámetros

A continuación se listan los diferentes tipos de parámetros del problema.

Parámetros generales del problema:

I = conjunto de empleados

h = número de días en el horizonte de planificación

$D = \{1 \dots h\}$ = conjunto de días en el horizonte de planificación

$W = \{1 \dots h/7\}$ = conjunto de fines de semana en el horizonte de planificación

T = conjunto de tipos de turnos

Parámetros asociados a los empleados:

N_i = conjunto de días en los que el empleado i no puede ser asignado a un turno

b_i = número mínimo de minutos que el empleado i debe ser asignado

c_i = número máximo de minutos que el empleado i puede ser asignado

f_i = número mínimo de turnos consecutivos que el empleado i debe trabajar

g_i = número máximo de turnos consecutivos que el empleado i puede trabajar

o_i = número mínimo de días consecutivos de descanso que el empleado i puede ser asignado

a_i = número máximo de fines de semana que el empleado i puede trabajar

Parámetros relacionados a los turnos:

R_t = conjunto de tipos de turnos que no pueden asignarse inmediatamente después del turno de tipo t

l_t = duración del turno de tipo t en minutos

Parámetros que relacionan empleados, días y turnos:

m_{it} = número máximo de turnos de tipo t que se pueden asignar al empleado i

q_{idt} = penalización si el turno de tipo t no se asigna al empleado i en el día d

p_{idt} = penalización si el turno de tipo t se asigna al empleado i en el día d

s_{dt} = número total preferido de empleados asignados al turno de tipo t en el día d

u_{dt} = peso si está por debajo de la cobertura preferida para el turno de tipo t en el día d

v_{dt} = peso si excede la cobertura preferida para el turno de tipo t en el día d

Variables de Decisión

$$x_{idt} = \begin{cases} 1, & \text{si el empleado } i \text{ está asignado al turno de tipo } t \text{ en el día } d \\ 0, & \text{en caso contrario} \end{cases}$$

$$k_{iw} = \begin{cases} 1, & \text{si el empleado } i \text{ trabaja el fin de semana } w \\ 0, & \text{en caso contrario} \end{cases}$$

y_{dt} = total por debajo de la cobertura preferida para el turno de tipo t en el día d

z_{dt} = total por encima de la cobertura preferida para el turno de tipo t en el día d

Restricciones

- Máximo de turnos por día: A un empleado se le asigna como máximo un turno cada día.

$$\sum_{t \in T} x_{idt} \leq 1, \quad \forall i \in I, d \in D \quad (1)$$

- Rotación de turnos: Se requiere un descanso mínimo después de un turno. Esto significa que ciertos turnos no pueden seguir a otros. Por ejemplo, un turno temprano no puede seguir a un turno tarde.

$$x_{idt} + x_{i(d+1)u} \leq 1, \quad \forall i \in I, d \in \{1, \dots, h-1\}, t \in T, u \in R_t \quad (2)$$

- Número máximo de turnos de cada tipo: Por ejemplo, algunos empleados pueden tener contratos que no les permiten trabajar turnos nocturnos. Otro empleado puede trabajar un número máximo de turnos tarde.

$$\sum_{d \in D} x_{idt} \leq m_{it}, \quad \forall i \in I, t \in T \quad (3)$$

- Mínimas y máximas horas de trabajo: El tiempo total trabajado por cada empleado debe estar dentro de un número mínimo y máximo de minutos. Estos límites pueden

variar dependiendo de si el empleado es de tiempo completo o de tiempo parcial, por ejemplo.

$$b_i \leq \sum_{d \in D} \sum_{t \in T} l_t x_{idt} \leq c_i, \quad \forall i \in I \quad (4)$$

- Máximo de turnos consecutivos: El número máximo de turnos que un empleado puede trabajar sin un día libre. Por ejemplo, los empleados a tiempo parcial a veces no trabajan tantos turnos consecutivos como el personal a tiempo completo.

$$\sum_{j=d}^{d+g_i} \sum_{t \in T} x_{ijt} \leq g_i, \quad \forall i \in I, d \in \{1, \dots, h - g_i\} \quad (5)$$

- Mínimo de turnos consecutivos: Esta restricción se modela evitando cualquier secuencia de turnos consecutivos por debajo del mínimo. Por ejemplo, si el número mínimo de turnos consecutivos es cuatro, no se deben permitir las siguientes secuencias: descanso-trabajo-descanso, descanso-trabajo-trabajo-descanso, descanso-trabajo-trabajo-trabajo-descanso, donde descanso es un día sin turno y trabajo es un día con un turno asignado.

$$\sum_{t \in T} x_{idt} + \sum_{j=d+1}^{d+s} \sum_{t \in T} x_{ijt} + \sum_{t \in T} x_{i(d+s+1)t} > 0, \\ \forall i \in I, s \in \{1, \dots, f_i - 1\}, d \in \{1, \dots, h - s + 1\} \quad (6)$$

- Mínimos días libres consecutivos: Esta restricción se modela de manera similar a la restricción de turnos consecutivos mínimos. Por ejemplo, si el número mínimo de días consecutivos libres es tres, no se deben permitir las siguientes secuencias: trabajo-descanso-trabajo, trabajo-descanso-descanso-trabajo.

$$1 - \sum_{t \in T} x_{idt} + \sum_{j=d+1}^{d+s} \sum_{t \in T} x_{ijt} + 1 - \sum_{t \in T} x_{i(d+s+1)t} > 0, \\ \forall i \in I, s \in \{1, \dots, o_i - 1\}, d \in \{1, \dots, h - s + 1\} \quad (7)$$

- Número máximo de fines de semana trabajados: Si el empleado tiene un turno un Sábado o un Domingo, es considerado trabajo durante el fin de semana, y por consiguiente

es considerado para el límite de fines de semana establecido.

$$k_{iw} \leq \sum_{t \in T} x_{i(7w-1)t} + \sum_{t \in T} x_{i(7w)t} \leq 2k_{iw}, \quad \forall i \in I, w \in W \quad (8)$$

$$\sum_{w \in W} k_{iw} \leq a_i, \quad \forall i \in I \quad (9)$$

- Días libres: Esta restricción vela por mantener días libres que no son transables, por ejemplo algún compromiso de un empleado

$$x_{idt} = 0, \quad \forall i \in I, d \in N_i, t \in T \quad (10)$$

- Requerimientos de cobertura: Esta restricción asegura que hay el mínimo número de personal disponible para cada turno

$$\sum_{i \in I} x_{idt} - z_{dt} + y_{dt} = s_{dt}, \quad \forall d \in D, t \in T \quad (11)$$

Funciones objetivo

$$\text{minimizar} \quad \sum_{i \in I} \sum_{d \in D} \sum_{t \in T} q_{idt}(1 - x_{idt}) + \sum_{i \in I} \sum_{d \in D} \sum_{t \in T} p_{idt}x_{idt} \quad (12.1)$$

Esta función objetivo busca minimizar la penalización por asignar un día libre cuando el trabajador prefería un día de trabajo, y la penalización por asignar un día de trabajo cuando el trabajador prefería un día libre

$$\text{minimizar} \quad \sum_{d \in D} \sum_{t \in T} y_{dt}u_{dt} + \sum_{d \in D} \sum_{t \in T} z_{dt}v_{dt} \quad (12.2)$$

Esta función objetivo busca minimizar la penalización por no cubrir o sobre-cubrir un turno de trabajo.

Capítulo 5

Algoritmos y propuesta de solución

El modelo abordado en este trabajo posee una mayor complejidad en comparación con investigaciones previas como la de [17]. Esto, debido a la inclusión de un conjunto más amplio de restricciones relacionadas a la incompatibilidad entre empleados, a pesar de considerar un objetivo menos. Estas restricciones adicionales toman en cuenta una gama más extensa de preferencias de los empleados y limitaciones de diversa índole, tales como restricciones legales y políticas empresariales. A diferencia del enfoque utilizado por [1], donde el problema se resuelve considerando un único objetivo simple (ambos en una única expresión matemática), el modelo actual lo aborda desde una perspectiva multiobjetivo, lo que agrega una capa adicional de complejidad al tener como meta encontrar un conjunto de soluciones de calidad bien distribuidas en un frente de Pareto.

Además de lo anterior, el modelo propuesto en este trabajo presenta una naturaleza adaptable a múltiples contextos debido a la formulación genérica de los conjuntos involucrados. Un ejemplo es el trabajo de [28], donde se formuló un modelo en función a las problemáticas mineras. Es por esto que la versatilidad de ESSP abre camino para futuras investigaciones y aplicaciones en diversos entornos laborales. La flexibilidad inherente del modelo permitirá abordar problemas de programación de personal en diferentes escenarios, ampliando así su alcance y relevancia práctica.

5.1. Acercamiento propuesto

En este trabajo se propone la adaptación del algoritmo genético multiobjetivo NSGA-II (Non-dominated Sorting Genetic Algorithm II) para resolver el problema de optimización descrito. El NSGA-II, propuesto por [23], es uno de los algoritmos evolutivos multi-objetivo más populares y utilizados debido a su eficiencia computacional, su mecanismo elitista y su capacidad para mantener la diversidad en la población sin necesidad de parámetros adicionales (los que requiere el cómputo de *fitness sharing*).

El funcionamiento básico del NSGA-II consiste en generar una población inicial de posibles soluciones, evaluar cada solución según las funciones objetivo, clasificarlas mediante un ordenamiento no dominado y luego aplicar operadores genéticos (selección, cruce y mutación) para generar una nueva población. Este proceso se repite iterativamente hasta alcanzar un criterio de parada.

NSGA-II se caracteriza principalmente por incorporar:

- Un procedimiento de ordenamiento no dominado rápido
- Un estimador de densidad basado en la distancia de crowding.
- Un operador de comparación de crowding para preservar la diversidad

Como se ilustra en la Figura 5.1, en cada generación t , se crea una población de descendientes Q_t a partir de la aplicación de los operadores de transformación en la población de padres P_t . Ambas poblaciones se combinan para formar $R_t = P_t \cup Q_t$. Luego, mediante el procedimiento de ordenamiento no dominado, la población R_t se clasifica en diferentes frentes de dominancia (F_1, F_2, \dots). Para formar la nueva población P_{t+1} , se incluyen soluciones completas de los frentes en orden ascendente hasta que el tamaño de la población exceda N . Las soluciones del último frente considerado se ordenan según la distancia de crowding y se seleccionan las mejores para completar exactamente la población de tamaño N . Es importante señalar que NSGA-II maneja explícitamente la infactibilidad de soluciones a través de lo que se define como *constrained-dominance*. Esta dominancia asegura que durante la selección se prioricen

las soluciones factibles de acuerdo a su nivel de dominancia y, posteriormente, las soluciones infactibles de acuerdo a su nivel de infactibilidad.

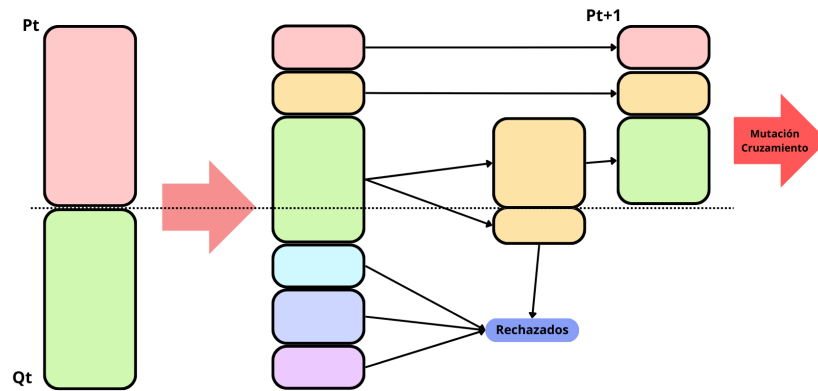


Figura 5.1: Proceso de selección en NSGA-II

5.2. Adaptaciones realizadas

En la implementación propuesta, la principal modificación realizada al algoritmo NSGA-II original es la adaptación del algoritmo para trabajar con representaciones de números enteros, mientras que la versión estándar de NSGA-II fue diseñada para operar con variables reales o binarias. Esta modificación fue necesaria debido a la naturaleza discreta del problema que se aborda, donde las variables de decisión representan el turno mediante un índice numérico.

A continuación, se presentan en detalle los componentes de la propuesta, comenzando por la representación de las soluciones, seguido de la función de evaluación diseñada y le evaluación de infactibilidad, y finalmente, la estructura general del algoritmo implementado junto con las adaptaciones específicas realizadas a los operadores genéticos para manejar variables enteras.

5.3. Representación

En este trabajo se adopta una representación híbrida de tipo *genotipo–fenotipo* para modelar soluciones del problema de programación de turnos de empleados. Esta elección busca equilibrar dos objetivos fundamentales: por una parte, permitir la aplicación directa y eficiente de operadores evolutivos estándar; y por otra, incorporar información estructural relevante del problema que facilite la evaluación de restricciones y la exploración del espacio de soluciones.

5.3.1. Nivel genotípico.

A nivel genotípico, cada solución se codifica como un arreglo unidimensional de valores enteros de tamaño $|E| \times |H|$, donde $|E|$ representa el número de empleados y $|H|$ el horizonte de planificación en días. Cada posición del arreglo corresponde a la asignación de un turno para un empleado específico en un día determinado.

Los valores que puede tomar cada casilla pertenecen al conjunto $\{0, 1, \dots, s\}$, donde $s = |T|$ es el número de tipos de turnos disponibles. En particular:

- el valor 0 representa un día libre,
- los valores 1 a s representan los distintos tipos de turnos laborales.

La interpretación de la posición p dentro del arreglo se realiza de la siguiente forma:

- **Empleado:** $i = p \bmod |E|$,
- **Día:** $d = \lfloor p/|E| \rfloor$.

Esta codificación es compacta y resulta especialmente adecuada para el uso de operadores genéticos clásicos, como mutación puntual y cruzamiento, ya que mantiene una estructura lineal simple y un dominio de valores acotado.

5.3.2. Nivel fenotípico.

A nivel fenotípico, la solución genotípica se decodifica en una representación estructurada basada en secuencias de trabajo. Para cada empleado $i \in E$, se define un conjunto de secuencias \mathcal{S}_i que representan bloques contiguos de turnos asignados, junto con un conjunto de días de inicio \mathcal{D}_i asociados a dichas secuencias.

Cada secuencia \mathcal{S}_i almacena explícitamente:

- el tipo de turno asignado en cada día del bloque,
- la longitud de la secuencia (número de días consecutivos trabajados),
- el tiempo total de trabajo acumulado en la secuencia.

Durante el proceso de decodificación, cada secuencia \mathcal{S}_i se ubica en el horizonte de planificación a partir de su día de inicio correspondiente en \mathcal{D}_i . Las posiciones no cubiertas por ninguna secuencia se completan automáticamente con días libres. De este modo, se garantiza una correspondencia coherente entre la representación genotípica y su interpretación fenotípica.

5.3.3. Motivación y ventajas de la representación híbrida.

La separación explícita entre genotipo y fenotipo permite explotar ventajas complementarias. Por un lado, el genotipo facilita la exploración del espacio de búsqueda mediante operadores evolutivos simples y eficientes. Por otro lado, la representación fenotípica hace explícita la estructura de bloques de trabajo consecutivo, lo cual resulta clave para:

- integrar secuencias factibles precomputadas que respeten restricciones locales (por ejemplo, máximos de días consecutivos o tiempos de trabajo),
- evaluar restricciones complejas de forma más eficiente,
- aplicar estrategias de búsqueda local a nivel de bloques completos, en lugar de modificaciones día a día.

5.3.4. Ejemplo ilustrativo.

La Figura 5.2 presenta un ejemplo de esta representación para un caso con $|E| = 3$ empleados, un horizonte reducido y un único tipo de turno. En la parte izquierda se muestra la codificación genotípica como un arreglo de enteros, mientras que en la parte derecha se observa su decodificación fenotípica, donde las secuencias \mathcal{S}_i de cada empleado se posicionan en los días de inicio \mathcal{D}_i , completando los espacios restantes con días libres.

Esta representación híbrida constituye la base sobre la cual se definen los operadores evolutivos, los mecanismos de reparación y el cálculo del nivel de infactibilidad, los cuales se describen en las secciones siguientes.

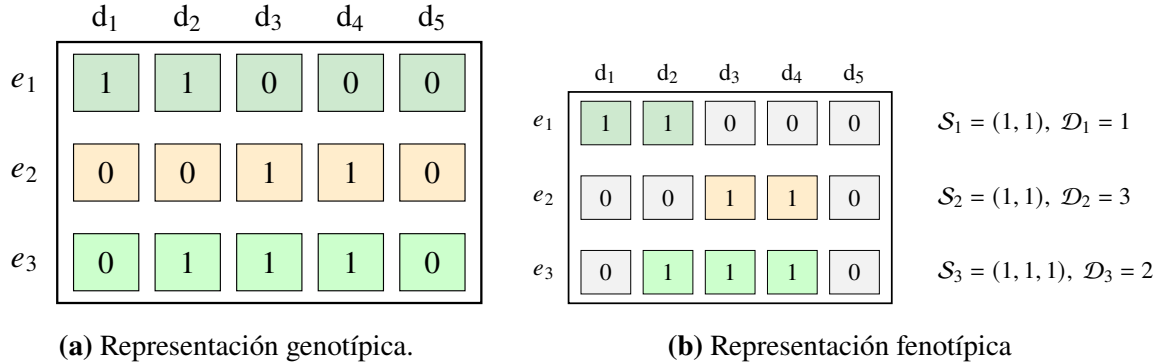


Figura 5.2: Ejemplo de representación genotípica y fenotípica de soluciones, usando 1 como turno trabajado y 0 como día libre.

5.4. Evaluación de soluciones

La evaluación de las soluciones en nuestro enfoque se realiza mediante dos componentes principales: las funciones objetivo y el manejo de restricciones mediante penalizaciones.

5.4.1. Funciones objetivo

Como se definió previamente en el modelo matemático, trabajamos con dos funciones objetivo:

1. **Preferencias de los empleados:** Esta función objetivo, definida en la ecuación (12.1), busca minimizar la penalización asociada a las asignaciones que no satisfacen las preferencias de los empleados.
2. **Cobertura de demanda:** Esta función objetivo, definida en la ecuación (12.2), busca minimizar las penalizaciones asociadas a la falta de cobertura o exceso de personal.

5.4.2. Manejo de restricciones

El manejo de restricciones se realiza mediante un esquema de penalización que se integra con el algoritmo NSGA-II. Cada violación de una restricción contribuye con una penalización de valor 1 a la variable `constr_violation` del individuo, que luego es utilizada por el algoritmo NSGA-II para realizar la comparación de soluciones en el proceso de selección. Cada restricción descrita en la sección 4 es verificada mediante este proceso.

Es importante destacar que para cada empleado, cada tipo de restricción puede ser violada como máximo una vez en la evaluación, lo que permite mantener un balance adecuado entre cada restricción, permitiendo también tener más claridad en la dificultad de satisfacer cada una.

El valor final de `constr_violation` es utilizado por el algoritmo NSGA-II en su proceso de ordenamiento no dominado, donde las soluciones factibles (aquellas con `constr_violation = 0`) dominan a las soluciones no factibles. Entre soluciones no factibles, aquellas con menor valor de `constr_violation` son preferidas.

5.5. Estructura general

Las primeras tres líneas del algoritmo NSGA-II descrito en el algoritmo 1 constituyen el punto de partida, donde se genera la población inicial $P_g(\lambda)$. Esta población inicial se construye siguiendo las estrategias detalladas en la sección 5.7, donde se presentan distintos métodos de inicialización diseñados para favorecer la factibilidad y diversidad de las soluciones.

Junto con esto, se genera una población de descendencia $Q_g(\lambda)$ en la línea 3, utilizando los operadores de selección, cruce y mutación, cuyos parámetros (como las probabilidades de cruce y mutación, tipo de operador, etc.) se encuentran definidos dentro de la instancia del problema π . Esta instancia agrupa toda la configuración relevante del algoritmo, permitiendo que cada fase de evolución utilice dichos parámetros de forma coherente y consistente a lo largo de las generaciones.

Algoritmo 1 *Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)*.

Input: Problem instance π (includes: population size λ , crossover and mutation probabilities, number of generations, objectives, constraints, etc.)

Output: Pareto front F .

```
1:  $g \leftarrow 0$ 
2: Generate initial population  $P_g(\lambda)$ 
3: Generate offspring population  $Q_g(\lambda)$  using parameters from  $\mathcal{I}$ 
4: while  $g < \text{MaxGen}$  do
5:   Combine populations  $R_g = P_g \cup Q_g$ 
6:   Classify  $R_g$  on non-dominated fronts  $F = (F_1, F_2, \dots)$ 
7:   Obtain new parent population  $P_{g+1}(\lambda)$ 
8:   Generate offspring population  $Q_{g+1}(\lambda)$  using parameters from  $\pi$ 
9:    $g \leftarrow g + 1$ 
10: end while
11: Combine populations  $R_g = P_g \cup Q_g$ 
12: Classify  $R_g$  on non-dominated fronts  $F = (F_1, F_2, \dots)$ 
13: return  $F_1 = 0$ 
```

En el bucle principal, que va desde la línea 4 a 9 , se combinan la población actual y la

población de descendencia en R_g , la cual se clasifica utilizando ordenación no dominada y la distancia de crowding empleadas en el trabajo de NSGA-II [23]. Estas técnicas permiten formar una nueva población de descendencia para continuar el ciclo durante las generaciones restantes.

Finalmente, en las líneas 11 y 12, la última población combinada R_g se reclasifica en frentes no dominados, y el primer frente F_1 , que representa las soluciones Pareto-óptimas, se devuelve como resultado del proceso.

A continuación se describen los operadores de transformación implementados para el algoritmo seguido de las propuestas de construcción de soluciones iniciales.

5.6. Operadores

5.6.1. Mutación

El algoritmo implementa cinco operadores de mutación diferentes que se seleccionan de forma probabilística ponderada. Estos operadores están diseñados específicamente para el problema de asignación de turnos, buscando mantener en la medida de lo posible, la factibilidad de las soluciones mientras exploran el espacio de búsqueda.

1. **Sequence Add mutation:** Este operador de mutación consiste en la inserción de un bloque de trabajo previamente precomputado dentro del horario de un empleado. En primer lugar, se selecciona aleatoriamente un empleado $e_i \in E$. A continuación, se escoge una secuencia factible $S \in \mathcal{S}$, de longitud L , y un día de inicio d_k dentro del horizonte de planificación.

La aplicación de la mutación reemplaza cualquier asignación previa del empleado e_i en el intervalo de días comprendido entre d_k y $d_k + L - 1$. En consecuencia, los turnos previamente asignados en dicho intervalo son eliminados y sustituidos por los turnos definidos en la secuencia S , mientras que las posiciones restantes del horario del empleado permanecen inalteradas.

Desde el punto de vista del proceso evolutivo, este operador permite introducir cambios estructurales significativos en la solución, ya que modifica bloques contiguos de asignación en lugar de alterar días individuales. Esto favorece la exploración del espacio de búsqueda mediante la incorporación de patrones de cobertura nuevos, los cuales pueden ser factibles o parcialmente factibles respecto a las restricciones del problema.

Sequence Add Mutation resulta especialmente efectiva para incrementar la diversidad de la población, dado que puede introducir configuraciones de trabajo que no son fácilmente alcanzables mediante mutaciones puntuales. Además, al basarse en secuencias precomputadas, el operador tiende a preservar estructuras localmente coherentes, lo que contribuye a mantener un balance adecuado entre exploración y explotación.

La Figura 5.3 ilustra el funcionamiento de este operador, donde se muestran dos secuencias previamente existentes (en rojo) y el nuevo bloque de secuencia insertado en el horario del empleado seleccionado.

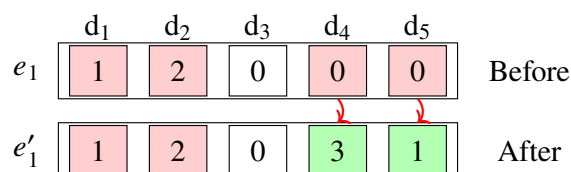


Figura 5.3: Sequence Add mutation. Una secuencia precomputada es insertada para el empleado e_1 , reemplazando bloques solapados. La primera secuencia es $\{1,2\}$, reemplazada por $\{3,1\}$.

2. **Replace Sequence mutation:** Este operador de mutación consiste en sustituir una secuencia de trabajo existente por otra secuencia alternativa de igual longitud, manteniendo inalterada la posición temporal del bloque dentro del horario del empleado. En primer lugar, se selecciona aleatoriamente un empleado $e_i \in E$. Luego, a partir del conjunto de secuencias actualmente asignadas a dicho empleado, se elige una secuencia específica S_{ij} .

Una vez identificada la secuencia a reemplazar, el algoritmo busca dentro del conjunto de secuencias precomputadas una secuencia candidata S' que tenga la misma longitud que S_{ij} y que permita mejorar alguno de los criterios del problema, tales como la satisfacción de preferencias del empleado o el equilibrio de cobertura global. Esta búsqueda

dirigida introduce un sesgo positivo hacia soluciones de mayor calidad sin modificar la estructura general del horario.

En el caso de que no se encuentre una secuencia que produzca una mejora directa en los objetivos considerados, se selecciona aleatoriamente una secuencia entre las mejores candidatas disponibles. Este mecanismo evita una convergencia prematura y contribuye a mantener la diversidad de la población, favoreciendo la exploración del espacio de búsqueda.

Desde el punto de vista de la factibilidad, el reemplazo por una secuencia de igual longitud preserva la coherencia temporal del horario del empleado, ya que no altera la duración del bloque ni su día de inicio. Por esta razón, este operador permite introducir mejoras locales en la solución sin afectar ninguna restricción estructural.

La Figura 5.4 presenta un ejemplo del funcionamiento de este operador. Antes de la mutación, el empleado e_1 posee dos secuencias de trabajo (indicadas en rojo). Durante la mutación, la segunda secuencia es reemplazada por una nueva secuencia de igual longitud (mostrada en verde), manteniendo la factibilidad del horario y potencialmente mejorando el valor de las funciones objetivo.

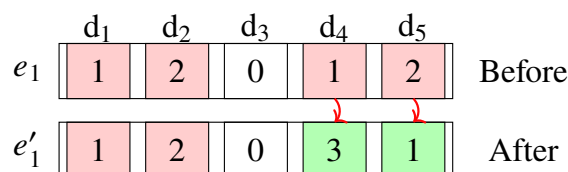


Figura 5.4: Replace Sequence mutation. La segunda secuencia de e_1 es reemplazada por otra de igual largo para mejorar preferencias y coberturas.

3. **Mutación de desplazamiento de secuencia (Shift mutation):** Este operador de mutación está orientado a intensificar la exploración local mediante modificaciones leves en asignaciones existentes, preservando en todo momento la factibilidad estructural del horario. En primer lugar, se selecciona aleatoriamente un empleado $e_i \in E$ y una de las secuencias de trabajo actualmente asignadas a dicho empleado.

Una vez identificada la secuencia, esta es desplazada temporalmente hacia adelante o hacia atrás dentro del horizonte de planificación, evaluando únicamente posiciones

factibles que no generen solapamientos con otras secuencias del mismo empleado ni violen restricciones del problema, tales como límites de días consecutivos o tiempos máximos de trabajo. El contenido de la secuencia, es decir, los tipos de turno y su duración, permanece inalterado durante el proceso.

Para cada posición factible candidata, se calcula una puntuación asociada al impacto del desplazamiento sobre los objetivos del problema, considerando criterios como la mejora en el balance de cobertura o el aumento en la satisfacción de preferencias. A partir de estas puntuaciones, se aplica un mecanismo de selección tipo *ruleta*, que asigna una mayor probabilidad de selección a las posiciones con mejor desempeño, sin descartar completamente alternativas subóptimas.

La Figura 5.5 presenta un ejemplo del funcionamiento de este operador. Antes de la mutación, el empleado e_1 posee una secuencia de trabajo (indicada en rojo). Durante la mutación, la segunda secuencia es desplazada una casilla a la derecha, como se ve en e'_1 (mostrada en verde).

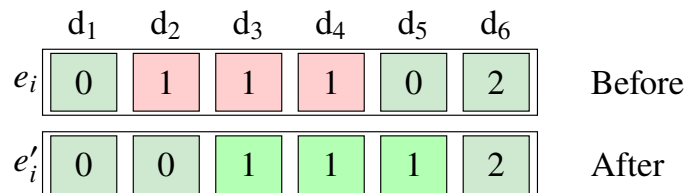


Figura 5.5: Shift mutation aplicada a la secuencia de un solo empleado. El bloque rojo se desplaza a una nueva posición factible (verde), mejorando la cobertura o la calidad de la preferencia.

4. **Adaptive Replace mutation:** Este operador de mutación introduce variabilidad adaptativa en la solución mediante la modificación de la longitud de secuencias de trabajo ya existentes. En primer lugar, se selecciona aleatoriamente un empleado $e_i \in E$ y una de las secuencias actualmente asignadas a dicho empleado.

A diferencia de la mutación de reemplazo de secuencia estándar, este operador no restringe el reemplazo a secuencias de igual longitud. En su lugar, se selecciona una longitud factible menor que la longitud original de la secuencia, permitiendo reducir la carga de trabajo del empleado sin introducir discontinuidades ni generar espacios vacíos entre bloques de trabajo.

Una vez determinada la nueva longitud, el algoritmo considera el conjunto de secuencias factibles precomputadas que cumplen con dicha duración. Cada secuencia candidata es evaluada en función de su impacto sobre los objetivos del problema, particularmente en términos de mejora del balance de cobertura y de la satisfacción de preferencias. Sobre la base de estas evaluaciones, se emplea un mecanismo de selección tipo *ruleta*, que asigna mayores probabilidades de selección a las secuencias que contribuyen en mayor medida a la calidad global del horario.

Desde una perspectiva metodológica, este operador permite ajustar de manera flexible la intensidad de trabajo asignada a los empleados, adaptándose a desbalances de cobertura o a situaciones de sobreasignación. Al mismo tiempo, al preservar la continuidad temporal de las secuencias, se mantiene la coherencia estructural del horario y se evita la introducción de fragmentaciones innecesarias.

La Figura 5.6 presenta un ejemplo del funcionamiento de este operador. Antes de la mutación, el empleado e_1 posee una secuencia de trabajo (indicada en rojo). Durante la mutación, la segunda secuencia es reemplazada por una secuencia de menor longitud, como se ve en e'_1 (mostrada en verde).

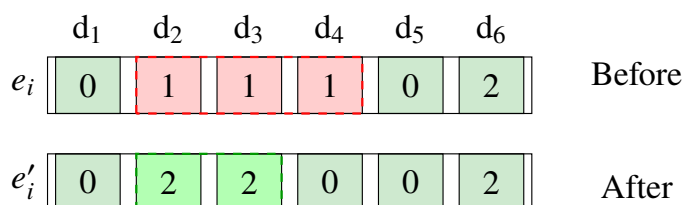


Figura 5.6: Adaptive replace mutation. La secuencia original (roja) se sustituye por una secuencia más corta y mejor (verde) para la cobertura o las preferencias.

5. **Replace Employee mutation:** Este operador de mutación consiste en sustituir completamente el horario de un empleado seleccionado por el horario de otro empleado factible, introduciendo patrones de trabajo distintos sin comprometer la factibilidad global de la solución. En primer lugar, se selecciona aleatoriamente un empleado $e_i \in E$. A continuación, se elige un empleado sustituto $e_j \in E$, distinto de e_i , cuya asignación de turnos resulta compatible con las restricciones del problema.

La aplicación de esta mutación implica reemplazar la totalidad de las secuencias y

asignaciones asociadas al empleado e_i por aquellas correspondientes al empleado e_j . De este modo, el empleado e_i adopta un nuevo patrón de trabajo completo, mientras que el resto de los empleados mantiene inalteradas sus asignaciones.

Desde el punto de vista del proceso evolutivo, esta mutación introduce cambios estructurales de gran escala, ya que modifica de forma significativa la distribución de turnos y preferencias asociadas a un individuo. Esto favorece la exploración del espacio de búsqueda a un nivel más global, permitiendo generar combinaciones de cobertura y satisfacción de preferencias que no serían alcanzables mediante operadores locales.

Asimismo, al basarse en empleados factibles como sustitutos, el operador preserva la coherencia estructural del horario y evita la introducción de asignaciones inviables. En consecuencia, la mutación de reemplazo de empleado contribuye a diversificar la población, ayuda a escapar de óptimos locales y complementa a los operadores de mutación de carácter más local o incremental descritos previamente.

La Figura 5.7 presenta un ejemplo del funcionamiento de este operador. Se reemplaza el empleado e_1 por el empleado e'_1

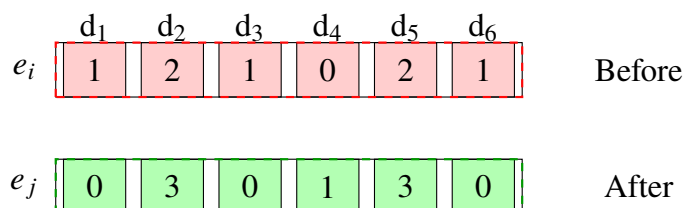


Figura 5.7: Replace employee mutation. Todo el horario del empleado e_i (rojo) se sustituye por otro empleado factible e_j (verde), lo que aumenta el potencial de exploración.

A cada operador se le aplica una probabilidad normalizada (p_{mut_i}), lo que permite controlar la frecuencia relativa de uso de cada tipo de mutación. Los operadores 2)-5) están diseñados específicamente para preservar restricciones como el número total de turnos asignados, al tiempo que exploran diferentes distribuciones temporales de estos turnos.

5.6.2. Crossover operators

Se proponen dos operadores de cruzamiento enfocados en preservar estructuras relevantes en la solución. Employee crossover busca mantener el conjunto de secuencias de turnos por empleados, y Sequence crossover busca mantener la selección de secuencias, pero perturbando el conjunto completo. Para cada par de padres, se selecciona uno probabilísticamente.

1. **Employee crossover:** Este operador de cruzamiento conserva las asignaciones de turnos completas para los empleados seleccionados, manteniendo la coherencia interna de sus horarios. Para cada empleado, se elige al azar a uno de los dos padres para que contribuya con su bloque de turnos completo a uno de los hijos, mientras que el otro padre proporciona la asignación complementaria al otro hijo. Este mecanismo promueve la diversidad mientras mantiene patrones de trabajo viables y coherentes. La figura 5.8 ilustra este proceso para dos empleados durante cuatro días.

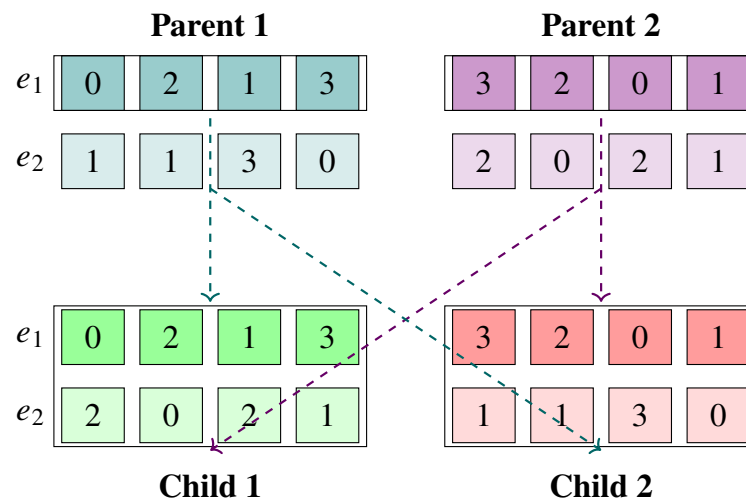


Figura 5.8: Employee crossover: complete employee schedules are inherited by alternating between parents. Colors indicate offspring.

2. **Sequence crossover:** Este operador de cruzamiento combina la estructura de secuencias precomputadas de ambos padres, intercambiando bloques completos de trabajo en lugar de asignaciones diarias individuales. Para cada empleado $e_i \in E$, se identifican todas las secuencias factibles presentes en los horarios de ambos padres.

Cada secuencia es considerada de manera independiente y es heredada por uno de los dos descendientes con probabilidad uniforme. De este modo, un descendiente puede heredar algunas secuencias del primer padre y otras del segundo, mientras que el otro descendiente recibe el complemento correspondiente. En ningún caso las secuencias son fragmentadas o modificadas internamente durante el cruzamiento.

Este enfoque permite preservar bloques de asignación que ya cumplen con las restricciones del problema, tales como continuidad temporal, límites de días consecutivos y patrones válidos de trabajo-descanso. A diferencia de los cruzamientos a nivel diario, el cruzamiento por secuencias opera sobre unidades estructurales de mayor nivel, manteniendo la factibilidad fenotípica y reutilizando patrones de alta calidad aprendidos durante la evolución.

En consecuencia, este operador favorece una recombinación estructural coherente, incrementa la diversidad de la población y mejora la capacidad del algoritmo para explorar nuevas combinaciones de cobertura y preferencias sin degradar la viabilidad de los horarios generados.

La Figura 5.9 presenta un ejemplo del funcionamiento de este operador. Como se observa, ambos padres poseen secuencias diferentes. Se selecciona al empleado e_1 y, para el hijo 1, se conserva la primera secuencia del padre 1 y la segunda secuencia del padre 2. De manera inversa, para el hijo 2 se mantiene la primera secuencia del padre 2 y la segunda del padre 1.

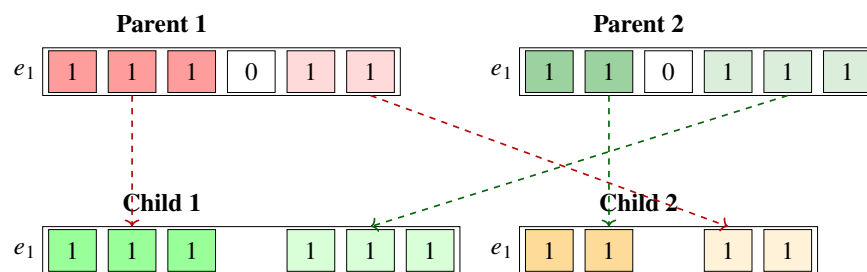


Figura 5.9: Sequence crossover: Los hijos heredan secuencias factibles completas de ambos padres, preservando la factibilidad en su mayoría, pero agregando exploración.

5.7. Fase de Construcción

La fase de construcción tiene como objetivo generar una población inicial diversa y completamente factible, alineada con la representación estructural basada en secuencias utilizada a lo largo del algoritmo evolutivo. A diferencia de enfoques tradicionales basados en asignaciones día a día, el método propuesto construye soluciones iniciales a partir de bloques de trabajo previamente validados, denominados *secuencias*, lo que permite reducir significativamente el espacio de búsqueda y evitar configuraciones inviables desde etapas tempranas del proceso evolutivo.

El procedimiento de inicialización se divide en tres etapas consecutivas: *Sequence Gather*, *Feasible Employees* y *Population Construction*. Estas etapas desacoplan explícitamente la generación de bloques factibles de la construcción final de individuos, promoviendo reutilización, diversidad estructural y consistencia con los operadores de mutación y cruzamiento basados en secuencias.

5.7.1. Sequence Gather

En la primera etapa se generan todas las secuencias factibles de trabajo-descanso para cada empleado de forma independiente. Para cada empleado e_i , se ejecuta un procedimiento de backtracking que explora combinaciones consecutivas de turnos, descartando de manera temprana aquellas secuencias parciales que violan restricciones locales.

El algoritmo 2 muestra la fase de generación de secuencias. Se recibe como parámetro la instancia del problema π , que contiene el conjunto de empleados, el horizonte de planificación y las restricciones. En la línea 1 se itera sobre cada empleado de manera independiente, permitiendo generar secuencias específicas de sus características contractuales. En la línea 2 se inicializa un conjunto vacío SP_i que almacenará todas las secuencias factibles del empleado e_i . La línea 3 crea una estructura auxiliar que indexa las secuencias por longitud, facilitando su uso posterior en operadores de mutación y cruzamiento. En la línea 4 se ejecuta el procedimiento de backtracking, que construye secuencias de manera incremental. La línea 5 aplica poda temprana, descartando secuencias parciales que violan restricciones locales como

Algoritmo 2 Sequence Gather Phase

Input: Problem instance π

- 1: **for** each employee $e_i \in E$ **do**
 - 2: Initialize empty sequence pool SP_i
 - 3: Initialize index by sequence length
 - 4: Generate sequences using backtracking
 - 5: Prune partial sequences violating local constraints
 - 6: Store each valid sequence in SP_i
 - 7: **end for**
 - 8: **return** All sequence pools $\{SP_i\} = 0$
-

máximo de turnos consecutivos, incompatibilidades entre turnos o límites por tipo de turno. Finalmente, en la línea 6, cada secuencia completamente válida se almacena en el *sequence pool* correspondiente.

Esta etapa es independiente del horizonte completo y no considera interacciones entre secuencias, lo que permite una generación exhaustiva y reutilizable de bloques factibles.

5.7.2. Feasible Employees

En la segunda etapa, las secuencias previamente generadas se combinan para construir horarios completos factibles a nivel de empleado. Para cada empleado, se seleccionan y posicionan secuencias del *sequence pool* de forma que cubran el horizonte de planificación sin violar restricciones globales.

El algoritmo 3 muestra la fase de construcción de empleados, usando las secuencias previamente creadas. Se reciben los conjuntos de secuencias factibles y la instancia del problema como parámetros. En la línea 1 se itera nuevamente por cada empleado. La línea 2 inicializa una asignación vacía que representará un horario completo para el empleado. En la línea 3 se ejecuta un segundo proceso de backtracking que explora combinaciones de secuencias a lo largo del horizonte temporal. La línea 4 selecciona secuencias del *sequence pool* según su longitud, permitiendo cubrir el horizonte de manera flexible. En la línea 5 se insertan bloques

Algoritmo 3 Feasible Employee Construction

Input: Sequence pools $\{SP_i\}$, problem instance π

- 1: **for** each employee $e_i \in E$ **do**
 - 2: Initialize empty employee assignment
 - 3: Apply backtracking over the planning horizon
 - 4: Select sequences from SP_i by length
 - 5: Insert remaining blocks when required
 - 6: Prune assignments violating global constraints
 - 7: Store each feasible assignment in employee pool EP_i
 - 8: **end for**
 - 9: **return** All employee pools $\{EP_i\} = 0$
-

de descanso cuando es necesario para respetar restricciones de continuidad y descanso mínimo. La línea 6 descarta configuraciones que violan restricciones de mayor nivel, tales como límites globales de turnos, número máximo de fines de semana trabajados o incoherencias entre secuencias consecutivas. Finalmente, en la línea 7, cada asignación completa y factible se almacena en el *employee pool* EP_i .

Esta etapa genera múltiples configuraciones factibles por empleado, promoviendo diversidad estructural desde la inicialización.

5.7.3. Population Construction

En la última etapa, la población inicial se construye combinando asignaciones factibles de cada empleado. Cada individuo se genera seleccionando aleatoriamente una asignación completa desde el *employee pool* correspondiente a cada empleado.

Algoritmo 4 Population Initialization

Input: Employee pools $\{EP_i\}$, population size P

```
1: for  $k = 1$  to  $P$  do
2:   Initialize empty individual  $ind_k$ 
3:   for each employee  $e_i$  do
4:     Randomly select assignment from  $EP_i$ 
5:     Copy sequences into  $ind_k$ 
6:   end for
7:   Evaluate individual  $ind_k$ 
8: end for
9: return Initialized population =0
```

El algoritmo 4 muestra la inicialización de la población por medio de los empleados factibles. Se reciben los *employee pools* y el tamaño deseado de la población como parámetros. La línea 1 itera para construir cada individuo de la población inicial. En la línea 2 se inicializa un individuo vacío. Las líneas 3 a 6 recorren todos los empleados y seleccionan aleatoriamente una asignación factible desde su *employee pool*, copiando sus secuencias y posiciones temporales al individuo. Finalmente, en la línea 7 se evalúa el individuo según las funciones objetivo del problema.

Dado que todas las asignaciones utilizadas son factibles por construcción, la población inicial resultante cumple íntegramente las restricciones del problema, eliminando la necesidad de procedimientos de reparación posteriores. Este enfoque proporciona una población inicial de alta calidad, diversa y estructuralmente consistente con los operadores evolutivos propuestos.

5.8. Parámetros

La Tabla 5.1 presenta un resumen de los parámetros definidos por la adaptación del algoritmo NSGA-II propuesta. Entre ellos, se encuentran parámetros fundamentales del algoritmo evolutivo, como el tamaño de la población (P) y el número máximo de generaciones (GENERATIONS), así como parámetros específicos de los operadores de cruce y mutación.

Tabla 5.1: Resumen de parámetros del algoritmo

Parámetro	Descripción	Uso	Rango
P	Tamaño de la población	AE	Enteros positivos
GENERATIONS	Número de generaciones	AE	Enteros positivos
p_{cross}	Probabilidad de cruzamiento	Cruzamiento	[0, 1]
p_{mut}	Probabilidad de mutación	Mutación	[0, 1]
$p_{mut_1} \dots p_{mut_5}$	Probabilidad de los operadores de mutación	Mutación	[0, 1]
mutamount	Cantidad de mutaciones por iteración	Mutación	[0, 1]
p_{cross_1}, p_{cross_2}	Probabilidad de los operadores de cruzamiento	Cruzamiento	[0, 1]

Capítulo 6

Experimentos

6.1. Experimentos

Para evaluar la efectividad del modelo propuesto, se implementaron dos enfoques metodológicos: un método exacto utilizando AMPL con el solver Gurobi y un algoritmo metaheurístico NSGA-II, con una parametrización en ParamILS. La parametrización realizada con ParamILS se realiza de forma separada para cada instancia del problema. Esto, con el propósito de relacionar las configuraciones de parámetros seleccionadas con las características de los casos de pruebas. Por esto, también se realiza una propuesta, evaluación y correlación de características de casos de pruebas respecto a los algoritmos sintonizados.

El objetivo final de todo este proceso es caracterizar las instancias utilizando la mejor versión de NSGA-II identificada para cada caso, y, en paralelo, buscar frentes de Pareto que equilibren las preferencias de los empleados con los requerimientos operacionales.

6.1.1. Configuración Experimental

Las pruebas se ejecutaron en un servidor con las siguientes especificaciones:

- Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz

- Ubuntu 22.04.3
- AMPL + Gurobi para la resolución exacta
- NSGA-II desarrollado en [C]

6.1.2. Descripción de las Instancias

Las 10 instancias utilizadas provienen del repositorio estándar que se han utilizado en el estado del arte [1]. Estas instancias varían en tamaño y complejidad (ver Tabla 6.1). Dentro de las características generales de las instancias que es posible observar, la más sencilla corresponde a la instancia 1 que posee solo 2 horizontes, 8 empleados y 1 turno. La instancia más compleja es la instancia 10, que cuenta con 4 horizontes, 40 empleados y 5 turnos. Estas diferencias implican espacios de búsqueda de distinta magnitud. Por ejemplo, considerando de manera muy simplificada la cantidad total de asignaciones posibles ($|H| \times |E| \times |T|$), la instancia 10 presenta alrededor de 800 combinaciones frente a solo 16 de la instancia 1.

Tabla 6.1: Instancias de pruebas

Instancias	$ H $	$ E $	$ T $
1	2	8	1
2	2	14	2
3	2	20	3
4	4	10	2
5	4	16	2
6	4	18	3
7	4	20	3
8	4	30	4
9	4	36	4
10	4	40	5

6.1.3. Análisis de características de las instancias

El análisis de características de las instancias en el problema de asignación de turnos de enfermería cumple un rol fundamental en la comprensión y predicción de la dificultad computacional asociada a cada caso. Inspirado en el enfoque propuesto por Vanhoucke y Maenhout [22], este análisis se articula en torno a un conjunto de indicadores que permiten caracterizar de manera objetiva las instancias del problema. Estos indicadores son clave no solo para anticipar el comportamiento de algoritmos exactos y metaheurísticos, sino también para facilitar la comparación entre instancias, seleccionar adecuadamente los métodos de resolución y comprender qué factores influyen en la complejidad inherente al problema. En este caso, nos servirán para dar una interpretación a la dificultad para encontrar soluciones factibles y de calidad al utilizar el algoritmo propuesto.

Estructura General de las Instancias

Cada instancia del problema está definida a partir de una serie de secciones que especifican los parámetros del entorno.

- La sección `SECTION_HORIZON` establece el horizonte de planificación, el cual usualmente comprende dos semanas.
- En la sección `SECTION_SHIFTS` se definen los tipos de turnos disponibles, así como las restricciones de sucesión entre ellos; por ejemplo, en muchas instancias, el turno de tarde no puede preceder a un turno de mañana.
- La sección `SECTION_STAFF` incluye información detallada sobre cada enfermero, tales como los turnos que puede cubrir, sus horas máximas permitidas, y las reglas que rigen la cantidad de turnos consecutivos.
- Se especifican los días no disponibles mediante `SECTION_DAYS_OFF` y se incorporan las preferencias individuales sobre turnos deseados o no deseados en las secciones `SHIFT_ON/OFF_REQUESTS`.

- Por último, la sección `SECTION_COVER` contiene los requerimientos de cobertura diaria por tipo de turno, junto con penalizaciones asociadas a la sobre cobertura o subcobertura.

Indicadores de Tamaño del Problema

El primer grupo de indicadores corresponde al tamaño estructural de la instancia. Se define por tres parámetros fundamentales:

- Número de enfermeros (E)
- Cantidad de turnos diferentes por día (T)
- Duración del horizonte de planificación (H)

Estos valores determinan directamente el tamaño del espacio de búsqueda. A mayor cantidad de días, turnos y personal, mayor será la complejidad combinatoria del problema, que es de naturaleza NP-difícil.

Indicadores de Cobertura

Los indicadores relacionados con la cobertura miden la distribución y exigencia de la demanda de personal durante el horizonte de planificación. Uno de los indicadores clave es el **Total Coverage Constrainedness (TCC)**, que evalúa la relación entre la cantidad total de turnos requeridos y la capacidad total del personal disponible. Este valor proporciona una medida de cuán ajustada es la planificación: valores cercanos a 1 indican instancias muy exigentes donde es difícil cumplir los requerimientos sin violar restricciones.

Complementariamente, se analiza la variabilidad en la cobertura mediante dos indicadores de dispersión: **Day Coverage Distribution (DCD)**, que mide la desigualdad en la carga de trabajo entre los distintos días; y **Shift Coverage Distribution (SCD)**, que evalúa el desbalance entre tipos de turnos dentro de cada día. Altos valores en DCD o SCD indican que la

demanda está concentrada en ciertos días o tipos de turno, lo cual reduce la flexibilidad del sistema y aumenta la dificultad de generar horarios factibles y de calidad.

Indicadores de Preferencias del Personal

Este grupo de indicadores busca cuantificar el grado de homogeneidad o dispersión en las solicitudes de los enfermeros. El **Nurse Preference Distribution (NPD)** mide qué tan uniforme es la preferencia de los trabajadores respecto a los turnos. Valores bajos reflejan preferencias más heterogéneas, lo que usualmente ofrece mayor flexibilidad. En contraste, valores altos indican consenso entre los trabajadores, lo cual puede generar conflictos al intentar satisfacer múltiples solicitudes similares. Este indicador también vive entre 0 y 1.

En el caso de preferencias binarias, donde los trabajadores expresan deseos de *trabajar* (preferencias ON) o de *no trabajar* (preferencias OFF) en ciertos turnos específicos, se adapta la fórmula original [22] de NPD utilizando una matriz de preferencias $p_{idt} \in \{-1, 0, +1\}$:

$$\text{NPD} = \frac{1}{|T|} \sum_{t \in T} \frac{\text{Var}(n_{it})}{\text{máx Var}(n_{it})}$$

donde n_{it} es el número de preferencias expresadas por cada trabajador i sobre el turno t , y la varianza se normaliza dividiendo por su valor máximo teórico.

El **Shift Preference Distribution (SPD)** analiza cómo se distribuyen las preferencias de cada enfermero entre los distintos turnos de un día. Si un enfermero muestra clara inclinación por un turno específico (SPD alto), se reducen las alternativas viables. Se define como la fracción de trabajadores cuya configuración de preferencias varía entre turnos dentro de un mismo día:

$$\text{SPD} = \frac{1}{|D|} \sum_{d \in D} \frac{1}{|E|} \sum_{i \in E} \# (\exists t_1, t_2 \in T : p_{idt_1} \neq p_{idt_2})$$

El **Day Preference Distribution (DPD)** observa la variabilidad en las preferencias a lo largo del tiempo. Altos valores implican que las preferencias de cada enfermero cambian considerablemente entre días, lo que complica la generación de soluciones que sean estables y

de buena calidad. Se calcula como la fracción de turnos para los cuales las preferencias del trabajador cambian entre días:

$$DPD = \frac{1}{|I|} \sum_{i \in I} \frac{1}{|T|} \sum_{t \in T} \mathbb{1}(\exists d_1, d_2 \in D : p_{id_1t} \neq p_{id_2t})$$

En todos los casos, los indicadores se encuentran normalizados en el intervalo $[0, 1]$, donde valores cercanos a 0 implican preferencias más homogéneas o constantes, y valores cercanos a 1 indican mayor variabilidad, asimetría o concentración en las solicitudes.

Indicadores Propuestos para Caracterización Específica

Además de los indicadores establecidos en la literatura que fueron propuestos para instancias de características similares, se proponen a continuación nuevos indicadores diseñados para capturar aspectos particulares de las instancias utilizadas en este estudio. Estos surgen de la observación de que muchas instancias reales incluyen restricciones y patrones que no son plenamente capturados por los indicadores tradicionales.

Preference Conflict Index (PCI):

Este indicador mide el grado de coincidencia entre trabajadores que desean cubrir un mismo turno en un mismo día. Altos valores implican concentración de preferencias en ciertos turnos, lo que puede generar conflictos al asignarlos.

$$PCI = \frac{1}{|D||T|} \sum_{d \in D} \sum_{t \in T} \left(\frac{|\{i \in E : p_{idt} = +1\}|^2}{|I|} \right)$$

Soft Conflict Pressure (SCP):

Cuantifica la presión ejercida por preferencias negativas (OFF) en relación con la demanda de cobertura. Captura cuántas veces es necesario ignorar preferencias individuales para cumplir

los requerimientos.

$$\text{SCP} = \frac{1}{|D||T|} \sum_{d,t} \left(\frac{\text{número de enfermeros con } p_{idt} = -1}{\text{cobertura requerida}_{dt}} \right)$$

Shift Diversity Index (SDI):

Evalúa qué tan versátil es el personal respecto a los turnos que puede cubrir. Valores altos indican que los trabajadores tienen disponibilidad amplia, lo cual favorece la flexibilidad.

$$\text{SDI} = \frac{1}{|I|} \sum_{i \in I} \frac{|\{t \in T : \exists d \in D, p_{idt} \neq 0\}|}{|T|}$$

Weekend Compression Ratio (WCR):

Cuando las instancias consideran explícitamente los fines de semana, este indicador mide qué proporción de las preferencias o requerimientos se concentran en sábado y domingo, en comparación con el total en todo el horizonte. Se tienen dos variantes, coverage, donde se analiza la cobertura, y preferences, donde se analiza a través de las preferencias de los trabajadores.

$$\text{WCR}_c = \frac{\sum_{(d,t) \in \mathcal{W} \times T} s_{d,t}}{\sum_{(d,t) \in D \times T} s_{d,t}} \quad \text{y} \quad \text{WCR}_p = \frac{\sum_{(i,d,t) \in \mathcal{P}_w} 1}{\sum_{(i,d,t) \in \mathcal{P}} 1}$$

Donde:

- $\mathcal{W} \subseteq D$ representa los días de fin de semana (por ejemplo, sábado y domingo).
- $s_{d,t}$ es la cobertura requerida en el día d y turno t .
- \mathcal{P} es el conjunto de todas las preferencias (tanto positivas como negativas).

- \mathcal{P}_W es el subconjunto de \mathcal{P} donde $d \in \mathcal{W}$.

Turn Incompatibility Pressure (TIP):

Este nuevo indicador cuantifica la presión estructural generada por una combinación de escasez de personal y restricciones entre turnos. Refleja qué tan difícil es asignar secuencias de turnos válidas cuando el personal disponible es limitado y muchas combinaciones están prohibidas. Se define como:

$$\text{TIP} = \frac{|C|}{|T|^2} \cdot \frac{1}{1 + \frac{|E|}{|T|}}$$

donde $|C|$ es la cantidad de pares de turnos incompatibles según las reglas de sucesión, $|I|$ es la cantidad de trabajadores y $|T|$ el número de turnos. Un TIP alto implica presión estructural para asignar turnos válidos, especialmente cuando el personal disponible es escaso.

Estos indicadores permiten capturar matices relevantes como la rigidez funcional del personal, la presión sobre los fines de semana, o la necesidad de resolver conflictos entre preferencias individuales y requerimientos colectivos. En conjunto con los indicadores clásicos, ofrecen una caracterización más rica de las instancias, ayudando a explicar mejor la dificultad relativa de encontrar soluciones factibles y de calidad.

La tabla 6.2 resume estos valores, que posteriormente se contrastan con los resultados obtenidos, y se observa cierta variabilidad entre las características de las instancias.

Podemos observar, por ejemplo, que el valor más alto corresponde al **Day Preference Distribution (DPD)**, con un máximo de 0.75 (instancias 1 y 5), mientras que los valores más bajos son 0.00 (en **Shift Concentration Degree (SCD)** y **Shift Preference Distribution (SPD)** para la instancia 1). En general, el **Total Coverage Constraint (TCC)** presenta valores relativamente altos y consistentes (entre 0.41 y 0.65), lo que indica que las restricciones de cobertura mantienen un impacto sostenido en todas las instancias. En contraste, características como el **Shift Concentration Degree (SCD)** y el **Nurse Preference Distribution (NPD)** presentan valores cercanos a cero en casi todas las instancias, reflejando una menor presión

generada por las preferencias en esos casos.

Por otro lado, el **Shift Diversity Index (SDI)** muestra los valores más altos (0.75 en las instancias 1 y 5) y también valores más bajos (0.32 en la instancia 3). El **Preference Conflict Index (PCI)** se mantiene muy bajo en todas las instancias (máximo 0.07, mínimo 0.02), lo que sugiere una baja conflictividad entre preferencias individuales. Por otro lado, los dos indicadores del **Weekend Compression Ratio (WCR_c y WCR_p)** oscilan en un rango moderado (aproximadamente entre 0.27 y 0.33), evidenciando cierta estabilidad en la concentración de restricciones o preferencias durante los fines de semana.

Tabla 6.2: Características de las instancias

Inst.	H	E	T	TCC	DCD	SCD	NPD	SPD	DPD	TIP	PCI	SCP	SDI	WCR_c	WCR_p
1	2	8	1	0.63	0.48	0	0.01	0	0.75	0	0.04	0.32	0.75	0.28	0.15
2	2	14	2	0.55	0.28	0.02	0.01	0.32	0.57	0.03	0.05	0.49	0.57	0.32	0.24
3	2	20	3	0.55	0.19	0.05	0.01	0.23	0.32	0.04	0.04	0.26	0.32	0.32	0.28
4	4	10	2	0.65	0.25	0.04	0.01	0.25	0.65	0.04	0.04	0.31	0.65	0.27	0.24
5	4	16	2	0.64	0.03	0.05	0	0.24	0.75	0.03	0.04	0.28	0.75	0.29	0.26
6	4	18	3	0.59	0.18	0.02	0.01	0.27	0.59	0.05	0.04	0.34	0.59	0.28	0.25
7	4	20	3	0.56	0.1	0.08	0	0.3	0.57	0.04	0.07	0.45	0.57	0.29	0.14
8	4	30	4	0.57	0.06	0.02	0	0.27	0.47	0.04	0.04	0.32	0.47	0.28	0.27
9	4	36	4	0.41	0.09	0.03	0	0.23	0.41	0.04	0.04	0.4	0.41	0.29	0.22
10	4	40	5	0.62	0.03	0.04	0	0.25	0.36	0.04	0.02	0.33	0.36	0.3	0.33

Estos indicadores permiten capturar matices relevantes como la rigidez funcional del personal, la presión sobre los fines de semana, o la necesidad de resolver conflictos entre preferencias individuales y requerimientos colectivos. En conjunto con los indicadores clásicos, ofrecen una caracterización más rica de las instancias, ayudando a explicar mejor la dificultad relativa de encontrar soluciones factibles y de calidad.

6.1.4. Metodología de evaluación

Método exacto (AMPL + Gurobi)

Para la obtención del frente de Pareto mediante el método exacto, se ponderan los objetivos normalizados de forma distribuida a lo largo de 11 ejecuciones, con un tiempo máximo de ejecución de 43 200 segundos (12 horas) por cada ejecución. La tabla 6.3 lista los valores de ponderadores de Fo_1 y Fo_2 considerados.

Ejecución	Fo_1	Fo_2
1	0.00001	0.99999
2	0.1	0.9
3	0.2	0.8
4	0.3	0.7
5	0.4	0.6
6	0.5	0.5
7	0.6	0.4
8	0.7	0.3
9	0.8	0.2
10	0.9	0.1
11	0.99999	0.00001

Tabla 6.3: Ponderaciones para el método exacto

Parametrización NSGA-II

A diferencia de una configuración homogénea para todas las instancias, el algoritmo NSGA-II modificado fue parametrizado individualmente para cada una. Este proceso se realizó utilizando la herramienta *ParamILS* [29], con el objetivo de encontrar la combinación de parámetros que optimizara el desempeño del algoritmo según múltiples criterios.

ParamILS (Parameter Iterated Local Search) es un framework de configuración automática de

algoritmos[29] que permite encontrar configuraciones de parámetros de alta calidad de manera sistemática. El framework combina técnicas de búsqueda local iterada con estrategias de intensificación y diversificación para explorar eficientemente el espacio de configuraciones.

El funcionamiento de ParamILS se basa en los siguientes componentes principales:

- **Representación del espacio de parámetros:** Define el conjunto de parámetros a optimizar, sus rangos de valores y dependencias entre ellos.
- **Función objetivo:** Métrica que evalúa la calidad de una configuración específica ejecutando el algoritmo objetivo en un conjunto de instancias de prueba.
- **Búsqueda local iterada:** Estrategia de búsqueda que combina movimientos locales (modificación de parámetros individuales) con perturbaciones para escapar de óptimos locales.
- **Gestión del presupuesto computacional:** Control del tiempo total de ejecución mediante la limitación del número de evaluaciones de configuraciones.

Una característica distintiva de ParamILS es su capacidad para manejar parámetros categóricos, continuos y condicionales de manera unificada. Además, utiliza técnicas de *capping* que terminan anticipadamente las ejecuciones que claramente no superarán la mejor configuración conocida, lo que permite una exploración más eficiente del espacio de búsqueda.

Configuración Experimental

El espacio de parámetros definido para la optimización automática comprende 17 parámetros diferentes que controlan distintos aspectos del algoritmo NSGA-II modificado. La Tabla 6.4 detalla el rango de valores posibles para cada parámetro, junto con sus valores por defecto utilizados como punto de partida en la búsqueda.

El espacio de configuraciones resultante contiene $5 \times 6 \times 4 \times 6^5 \times 4 \times 5^2 = 9.33 \times 10^7$ configuraciones posibles, lo que justifica plenamente el uso de un método de optimización automática como ParamILS para su exploración eficiente. La función objetivo utilizada para

Parámetro	Valores posibles	Valor inicial
p_{mut_real}	{0.01, 0.05, 0.10, 0.2, 0.25}	0.05
p_{cross_real}	{0.1, 0.25, 0.5, 0.6, 0.8, 1}	1
p	{12, 20, 52, 100}	52
p_{mut_1}	{0, 0.2, 0.4, 0.6, 0.8, 1}	1
p_{mut_2}	{0, 0.2, 0.4, 0.6, 0.8, 1}	0.4
p_{mut_3}	{0, 0.2, 0.4, 0.6, 0.8, 1}	0.4
p_{mut_4}	{0, 0.2, 0.4, 0.6, 0.8, 1}	0.8
p_{mut_5}	{0, 0.2, 0.4, 0.6, 0.8, 1}	0.2
mutamount	{0.05, 0.1, 0.2, 0.3}	0.2
p_{cross_1}	{0, 0.3, 0.5, 0.7, 1}	0
p_{cross_2}	{0, 0.3, 0.5, 0.7, 1}	0

Tabla 6.4: Espacio de parámetros definido para ParamILS, donde los valores posibles son valores discretos, y el valor por defecto es el valor con el que se comienza la parametrización. para más información sobre cada parámetro revisar la tabla de parámetros 5.1

guiar la búsqueda de parámetros fue una combinación ponderada de tres factores: calidad de la solución, penalización por restricciones violadas, y tiempo de ejecución. Específicamente, la métrica a minimizar fue:

$$\text{Score} = \text{Norm}(Fo_1) + \text{Norm}(Fo_2) + \text{Norm}(\text{Violation}) \quad (6.1)$$

donde:

- Fo_1 y Fo_2 representan los valores obtenidos para los dos objetivos del problema (e.g., satisfacción del personal y cobertura de turnos).
- **Violaciones** corresponde a la cantidad total de restricciones incumplidas en una solución.

Para calcular el número de restricciones incumplidas, se consideró que cada empleado puede infringir cada restricción independientemente solo una vez. Estas restricciones se evalúan de forma binaria (0 si se cumple, 1 si se incumple), y la suma total de violaciones corresponde al número de indicadores activados para todos los empleados. Por ejemplo, si un empleado viola tres restricciones y otro cinco, el total de violaciones será 8.

Cada uno de los componentes de la función de evaluación es normalizado respecto a su valor máximo posible, con el objetivo de asegurar una contribución balanceada de los distintos términos y evitar que alguno de ellos domine el proceso de búsqueda de parámetros. Esta normalización es coherente con el enfoque del trabajo, que busca preservar un tratamiento equitativo entre los objetivos del problema durante la fase de ajuste automático de parámetros.

Capítulo 7

Resultados y análisis

Este capítulo presenta los resultados experimentales obtenidos a partir de la aplicación del marco metodológico propuesto, junto con un análisis detallado de su comportamiento. En primer lugar, se reportan y analizan los resultados del proceso de parametrización automática mediante ParamILS, destacando los patrones emergentes en la configuración de parámetros frente a instancias de distinta complejidad. Posteriormente, se evalúa el desempeño del algoritmo propuesto comparándolo con un método exacto basado en programación lineal entera, considerando tanto métricas de calidad de solución como tiempos computacionales. Finalmente, se integran los resultados con la caracterización previa de las instancias, con el objetivo de interpretar el impacto de las propiedades estructurales del problema sobre el comportamiento algorítmico y la dificultad de resolución.

7.1. Resultados de la parametrización

El proceso de parametrización automática condujo a configuraciones específicas para cada instancia, reflejando las características particulares de cada problema. La Tabla 7.1 muestra las configuraciones óptimas encontradas por ParamILS para cada una de las 10 instancias del conjunto de prueba.

Tabla 7.1: Configuraciones finales obtenidas mediante ParamILS para las 10 instancias. Se reportan los valores reales ponderados utilizados por el algoritmo.

Instancia	mutamount	p	p_c	p_{c1}	p_{c2}	p_m	p_{m1}	p_{m2}	p_{m3}	p_{m4}	p_{m5}
1	0.30	12	0.10	0.50	0.50	0.40	0.00	0.23	0.08	0.31	0.38
2	0.01	12	0.25	0.50	0.50	0.80	0.06	0.29	0.24	0.18	0.24
3	0.01	100	0.80	0.50	0.50	0.80	0.00	0.50	0.17	0.00	0.33
4	0.05	100	0.80	0.50	0.50	0.80	0.11	0.11	0.00	0.33	0.44
5	0.10	100	1.00	0.50	0.50	0.80	0.13	0.25	0.13	0.00	0.50
6	0.10	100	1.00	0.50	1.00	0.80	0.11	0.22	0.22	0.00	0.44
7	0.30	100	1.00	0.50	0.50	0.20	0.36	0.14	0.14	0.29	0.07
8	0.10	100	1.00	0.50	0.50	0.20	0.10	0.20	0.20	0.40	0.10
9	0.10	100	1.00	0.50	0.50	0.20	0.36	0.14	0.14	0.29	0.07
10	0.10	100	1.00	0.50	0.50	0.20	0.36	0.14	0.14	0.29	0.07

La Tabla 7.1 resume las configuraciones finales obtenidas mediante ParamILS para cada instancia, considerando directamente los valores reales utilizados por el algoritmo tras la ponderación interna de probabilidades. La consolidación de parámetros en una única tabla permite identificar con mayor claridad los patrones de comportamiento del algoritmo frente a distintos niveles de complejidad.

En primer lugar, se observa una clara diferenciación en el tamaño de población (p). Las instancias más pequeñas (Instancias 1 y 2) convergen hacia poblaciones reducidas de 12 individuos, mientras que todas las instancias de mayor tamaño requieren poblaciones de 100 individuos. Esto sugiere que ParamILS detecta correctamente un aumento significativo en la complejidad estructural del problema a partir de la Instancia 3, favoreciendo una mayor diversidad poblacional.

Respecto a la probabilidad de cruza (p_c), las instancias pequeñas presentan valores moderados (0.10 y 0.25), mientras que en las instancias medianas y grandes se observa una preferencia clara por valores altos, alcanzando $p_c = 1.0$ en la mayoría de los casos. Este comportamiento indica que, en problemas más complejos, la recombinación sistemática de soluciones resulta

fundamental para explorar eficazmente el espacio de búsqueda.

En cuanto a la mutación, se aprecia una reducción progresiva del parámetro *mutamount* a medida que aumenta el tamaño de la instancia, estabilizándose mayoritariamente en valores cercanos a 0.1. Esto sugiere una estrategia más conservadora, donde la mutación actúa como mecanismo de diversificación controlada, evitando perturbar excesivamente estructuras ya bien adaptadas.

El análisis de las probabilidades individuales de mutación revela patrones consistentes: ciertos operadores (en particular p_{m2} , p_{m3} y p_{m4}) concentran mayor probabilidad en instancias grandes, lo que indica que estas mutaciones resultan más efectivas para refinar soluciones complejas. En contraste, otros operadores tienden a desaparecer o a activarse solo en instancias pequeñas, evidenciando un proceso de especialización automática.

7.1.1. Correlación de características y valores de parámetros

Con el objetivo de analizar la posible relación entre los parámetros estructurales de las instancias y las características utilizadas para su caracterización, se construyó una matriz de correlación basada en el coeficiente de Pearson. Este análisis permite identificar asociaciones lineales entre los parámetros del problema —número de enfermeros (E), número de turnos por día (T) y duración del horizonte de planificación (H)— y los distintos indicadores definidos en la Sección 6.1.2.

La Figura 7.1 presenta la matriz de correlación considerando la magnitud absoluta del coeficiente $|\rho|$, lo que facilita la interpretación en términos de fuerza de asociación, independientemente de su signo. De acuerdo con la literatura en análisis multivariante, valores de correlación con $|\rho| \geq 0.7$ son comúnmente interpretados como asociaciones fuertes o sustanciales, mientras que valores entre 0.5 y 0.7 indican correlaciones moderadas [30]. En este trabajo, se adopta el umbral $|\rho| = 0.7$ como criterio para identificar relaciones suficientemente significativas que permitan extraer conclusiones.

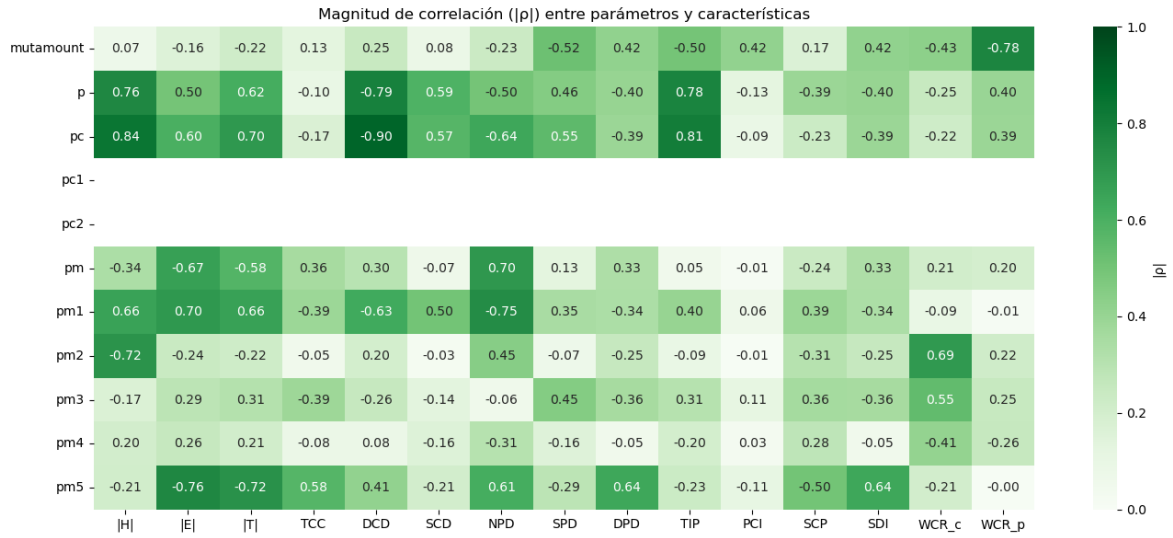


Figura 7.1: Matriz de correlación de magnitud absoluta ($|\rho|$) entre parámetros estructurales, parámetros del algoritmo y características de las instancias. Los parámetros $pc1$ y $pc2$ no presentan valores de correlación debido a que permanecen constantes en todas las instancias, por lo que no inducen variabilidad estadísticamente relevante.

Al analizar la matriz de correlación presentada en la Figura 7.1, se observa que no todos los indicadores de caracterización aportan información independiente respecto a la estructura de las instancias. En particular, el indicador Total Coverage Constrainedness (TCC) presenta correlaciones fuertes con los parámetros de tamaño del problema, específicamente con el número de enfermeros (E) y el número de turnos por día (T). Dado que estos parámetros definen directamente la capacidad total del sistema y la demanda agregada, esta relación indica que TCC captura principalmente efectos de escala. En consecuencia, TCC puede considerarse parcialmente redundante respecto a los parámetros estructurales, resultando útil como medida normalizada para comparar instancias de distinto tamaño, pero sin aportar información cualitativamente nueva sobre la topología del espacio de búsqueda.

Desde la perspectiva de los parámetros del algoritmo, se identifican patrones de correlación consistentes que permiten extraer conclusiones operativas. En particular, la probabilidad de cruzamiento (pc) muestra una correlación elevada con los parámetros de tamaño del problema, lo que sugiere que instancias con espacios de búsqueda más grandes requieren una mayor

intensidad de recombinación para explorar regiones diversas del frente de Pareto. Este resultado refuerza la noción de que el cruzamiento cumple un rol clave en la exploración global cuando la dimensionalidad combinatoria del problema aumenta.

Un comportamiento análogo se observa en algunas probabilidades de mutación. La mutación 1 presenta una correlación positiva alta con el número de enfermeros ($|\rho| \approx 0.7$), lo que indica que su activación resulta más relevante a medida que crece el tamaño del problema. Esto sugiere que dicha mutación contribuye a introducir variaciones estructurales necesarias para mantener diversidad en espacios de búsqueda amplios. En contraste, las mutaciones 2 y 5 exhiben correlaciones negativas con los parámetros de tamaño, lo que indica que su utilidad disminuye en instancias grandes. Este comportamiento es consistente con un escenario donde dichas mutaciones actúan como operadores de refinamiento local, siendo más efectivas en instancias pequeñas o moderadas, mientras que en problemas de gran escala se privilegian mecanismos de exploración más agresivos.

En contraste con los indicadores ligados a la escala, el Shift Preference Distribution (SPD) presenta una correlación fuerte con el número de enfermeros, pero captura un aspecto distinto de la estructura del problema: la heterogeneidad de preferencias intradía, i.e. de las preferencias individuales de los empleados dentro de un mismo día ó qué tan homogéneas son las solicitudes de los empleados a lo largo del horizonte. A diferencia de TCC o PCI, SPD no se limita a reflejar el tamaño de la instancia, sino que caracteriza directamente la pérdida de flexibilidad inducida por preferencias divergentes. Por este motivo, SPD emerge como un indicador particularmente informativo para anticipar dificultades asociadas a la satisfacción de preferencias y al compromiso entre objetivos.

Desde una perspectiva algorítmica, esta diferenciación resulta clave. Indicadores altamente correlacionados con parámetros estructurales, como TCC o PCI, permiten estimar la rigidez global del problema, pero ofrecen un valor limitado para guiar decisiones finas dentro del algoritmo. En cambio, indicadores como SPD —que capturan conflictos locales y asimetrías en las preferencias— constituyen señales adecuadas para activar estrategias adaptativas, tales como el aumento de tasas de mutación orientadas a la diversificación o la selección preferente de operadores capaces de reconfigurar asignaciones conflictivas.

Asimismo, indicadores como Turn Incompatibility Pressure (TIP) y Shift Diversity Index (SDI), al presentar una dependencia menor con el tamaño del problema, permiten identificar instancias cuya dificultad no proviene principalmente del volumen combinatorio, sino de restricciones estructurales específicas o de una limitada versatilidad del personal. En estos escenarios, la información proporcionada por las características puede utilizarse para priorizar operadores que modifiquen secuencias de turnos o que exploren configuraciones alternativas con mayor libertad estructural.

Con todo esto, podemos concluir que el análisis de correlaciones confirma que las características propuestas no solo describen la estructura de las instancias, sino que permiten predecir el comportamiento óptimo de algunos de los parámetros del algoritmo, habilitando estrategias de configuración y adaptación guiadas por la caracterización previa del problema.

7.2. Resultados del método exacto

A continuación, se presentan los principales resultados obtenidos mediante la aplicación del método exacto a las instancias del problema. Se describen brevemente el procedimiento seguido, los indicadores de calidad empleados para la evaluación y un análisis de los tiempos de cómputo requeridos.

7.2.1. Resumen de resultados del método exacto (AMPL)

El método exacto fue aplicado exhaustivamente a todas las instancias del problema, obteniendo soluciones óptimas para diferentes combinaciones de ponderaciones de funciones objetivo. Se considera un tiempo máximo por ejecución de 43200[s]. A partir de estos resultados, se construyó un análisis comprensivo que incluye la evaluación mediante el indicador de *Hypervolume* (HV) y el análisis de los tiempos de cómputo asociados a cada instancia.

7.2.2. Cálculo del indicador Hipervolumen

El **Hipervolumen** (HV) [31] fue seleccionado como métrica principal de evaluación debido a su capacidad para medir tanto la convergencia como la diversidad de un conjunto de soluciones no dominadas de manera unificada. Este indicador calcula el volumen del espacio objetivo dominado por el conjunto de soluciones con respecto a un punto de referencia específico.

Matemáticamente, el Hipervolumen se define como:

$$HV(S) = \lambda \left(\bigcup_{s \in S} [s_1, r_1] \times [s_2, r_2] \times \dots \times [s_m, r_m] \right) \quad (7.1)$$

donde:

- S es el conjunto de soluciones no dominadas
- λ representa la medida de Lebesgue (volumen)
- s_i es el valor de la i -ésima función objetivo para la solución s
- r_i es el valor de la i -ésima coordenada del punto de referencia
- m es el número de objetivos

Determinación del punto de referencia:

Para garantizar una evaluación equitativa y comparable entre diferentes instancias, se estableció un punto de referencia escalado basado en los valores máximos observados en cada función objetivo del conjunto de soluciones óptimas. El punto de referencia se calcula mediante:

$$r_i = \max_{s \in S} (f_i(s)) \times \alpha_i \quad (7.2)$$

, donde α_i es un factor de escalamiento específico para cada objetivo. En este estudio se utilizaron los multiplicadores $\alpha_1 = 3$ y $\alpha_2 = 5$ para las funciones objetivo 1 y 2 respectivamente,

con el propósito de poder evaluar la mayor parte de las soluciones encontradas por NSGA-II, independiente de su calidad.

Se estableció un punto de referencia escalado basado en los valores máximos observados en cada función objetivo del conjunto de soluciones óptimas. El punto de referencia se calcula mediante:

$$r_i = \max_{s \in S} (f_i(s)) \times \alpha_i \quad (7.3)$$

, donde α_i es un factor de escalamiento específico para cada objetivo. En este estudio se utilizaron los multiplicadores $\alpha_1 = 3$ y $\alpha_2 = 5$ para las funciones objetivo 1 y 2 respectivamente, con el propósito de poder evaluar la mayor parte de soluciones encontradas por NSGA-II, independiente de su calidad.

La Tabla 7.2 presenta una síntesis de los resultados obtenidos, mostrando el valor de HV calculado junto con los tiempos de resolución mínimo, máximo, promedio y total para cada instancia evaluada.

Instancia	HV	Ref F1	Ref F2	Soluciones	T. min (s)	T. max (s)	T. prom (s)	T. total (s)
Instancia 1	82 700	3 006	35	4	0	1	0.27	3
Instancia 2	749 622	5 103	185	6	0	6	1.36	15
Instancia 3	642 150	13 845	50	4	1	6	3.82	42
Instancia 4	893 757	6 606	185	5	0	10	3.73	41
Instancia 5	4 519 457	18 084	275	8	1	70	19.18	211
Instancia 6	10 378 310	25 644	445	8	3	31	13.45	148
Instancia 7	16 169 460	25 941	660	8	2	144	42.36	466
Instancia 8	37 336 341	36 885	1 090	8	17	14 217	2 936.00	32 296
Instancia 9	244 765	1 503	260	3	7	3 610	3 279.00	36 069
Instancia 10	33 754 419	49 140	770	7	11	160	50.73	558

Tabla 7.2: Resumen de resultados del método exacto (AMPL), incluyendo puntos de referencia y número de soluciones no dominadas.

Al observar los resultados del método exacto (AMPL) presentados en la Tabla 7.2, podemos notar que el número de soluciones no dominadas encontradas varía entre 3 y 8 para la mayoría de las instancias, lo que indica que el espacio de soluciones óptimas tiene una diversidad moderada. Aunque esto también puede estar determinado por la limitación que brinda AMPL

a la hora de encontrar múltiples soluciones, puesto que como máximo podemos obtener 11 soluciones distintas, que es la cantidad de ejecuciones con distinta ponderación.

Respecto a los tiempos de cómputo, existen diferencias significativas entre instancias: mientras algunas, como la Instancia 1, requieren menos de un segundo en promedio, otras, como la Instancia 8 y la Instancia 9, presentan tiempos máximos muy elevados, más de 14 000 y 3 600 segundos respectivamente, y tiempos promedio que alcanzan miles de segundos, mostrando una mayor complejidad computacional. Esta situación se refleja también en los tiempos totales de ejecución, que superan las 30 000 segundos en algunos casos. Esto es curioso porque nos permite observar que el espacio de búsqueda no indica necesariamente la complejidad a la hora de encontrar soluciones. La instancia 10 es más compleja desde el punto de vista de su espacio de búsqueda, sin embargo tiene tiempos mucho menores que la Instancia 8.

Finalmente, podemos ver que la diferencia entre tiempos mínimos y máximos es notable. De esto podemos suponer que existen ejecuciones (ponderaciones de objetivos) que facilitan la obtención de soluciones.

7.3. Resultados de NSGA-II

7.3.1. Comparación de frentes de Pareto

Se realizó una evaluación exhaustiva de NSGA-II en las primeras 10 instancias del conjunto de prueba. La Tabla 6.2 presenta los resultados obtenidos por el algoritmo metaheurístico, incluyendo métricas de calidad del frente de Pareto. Para la obtención de estos resultados se utilizaron 20 semillas aleatorias diferentes, una para cada ejecución. Posteriormente, se agregaron los frentes de cada resultado obtenido. Luego, se calculó la dominancia del frente agregado, dejando solo las soluciones no-dominadas y a partir de ese frente se obtuvieron la cantidad de soluciones finales y el hipervolumen mostrado. En cuanto al tiempo, este se obtiene a partir de la suma de los tiempos de inicialización y ejecución de las 20 semillas.

Como se observa en la Tabla 7.3, los resultados evidencian diferencias relevantes entre el enfoque metaheurístico NSGA-II y el método exacto basado en AMPL, tanto en términos de

Instancia	Soluciones		Tiempo (s)		Gap HV (%)	Ratio de tiempo
	NSGA-II	AMPL	NSGA-II	AMPL		
1	7	4	81	3	+0.25	27×
2	6	6	110	15	-10.80	7.3×
3	4	4	198	42	-8.56	4.7×
4	12	5	177	41	-12.60	4.3×
5	12	8	247	211	-8.26	1.1×
6	15	8	344	148	-9.40	2.32×
7	31	8	346	466	-7.99	-1.3×
8	38	8	1 046	32 296	-23.45	-30.8×
9	7	3	2 011	36 069	-100.00	-17.9×
10	31	7	717	558	-27.15	1.28×

Tabla 7.3: Comparación de resultados entre

NSGA-II y método exacto (AMPL) para las primeras 10 instancias. Gap HV: diferencia porcentual en hipervolumen $((NSGA-II - AMPL)/AMPL \times 100)$; valores positivos indican superioridad de NSGA-II. Ratio Tiempo: factor de diferencia en tiempo de ejecución $(NSGA-II/AMPL)$.

cantidad de soluciones no dominadas como de calidad del frente, medida mediante hipervolumen, y tiempos de ejecución.

En cuanto a la cantidad de soluciones no dominadas, NSGA-II genera sistemáticamente un mayor número de soluciones en la mayoría de las instancias. Esta diferencia es especialmente marcada en instancias de mayor tamaño, como las Instancias 7, 8 y 10, donde el metaheurístico obtiene hasta 31, 38 y 31 soluciones, respectivamente, frente a 8 o menos soluciones encontradas por AMPL. Este comportamiento confirma la capacidad del enfoque evolutivo para explorar regiones diversas del espacio de soluciones y ofrecer múltiples alternativas al tomador de decisiones.

No obstante, un mayor número de soluciones no dominadas no implica necesariamente una mejor calidad del frente. Al analizar el Gap de hipervolumen, se observa que en casi todas las instancias el hipervolumen alcanzado por NSGA-II es inferior al obtenido por AMPL, lo que se refleja en valores negativos del gap. En instancias de tamaño pequeño e intermedio

(Instancias 2 a 6), estas diferencias se mantienen en rangos moderados, típicamente entre un -8% y un -13% , indicando que, si bien NSGA-II no alcanza el frente óptimo, logra aproximaciones razonables.

En las instancias de mayor complejidad (Instancias 8 y 9), el gap de hipervolumen aumenta de manera significativa, alcanzando valores de hasta -23.45% y -100% , respectivamente. Esto sugiere que, pese a generar una gran cantidad de soluciones, el metaheurístico presenta dificultades para cubrir adecuadamente los extremos del frente de Pareto óptimo cuando el problema se vuelve altamente restrictivo. En contraste, el método exacto mantiene una mejor cobertura global del frente, aun cuando el número de soluciones obtenidas sea menor.

Una excepción relevante se observa en la Instancia 1, donde NSGA-II alcanza un hipervolumen levemente superior al de AMPL ($+0.25\%$). Este resultado puede atribuirse a la naturaleza exploratoria del algoritmo evolutivo en instancias pequeñas, donde es capaz de identificar soluciones no dominadas adicionales que mejoran marginalmente la cobertura del frente. Este resultado es altamente destacable, ya que demuestra la capacidad que tienen los metaheurísticos de superar métricas de algoritmos exactos cuando hablamos de enfoques multi-objetivo.

Respecto a los tiempos de ejecución, el comportamiento es heterogéneo y fuertemente dependiente del tamaño de la instancia. En las instancias pequeñas (Instancias 1 a 4), AMPL resulta claramente más eficiente, siendo hasta 27 veces más rápido que NSGA-II. Sin embargo, a medida que aumenta la complejidad del problema, el tiempo requerido por el método exacto crece de forma considerable. En particular, en las Instancias 8 y 9, AMPL presenta tiempos de ejecución extremadamente elevados, superiores a los 30 000 segundos, mientras que NSGA-II mantiene tiempos del orden de minutos. Esto se refleja en ratios de tiempo negativos, indicando que en estas instancias el enfoque metaheurístico resulta sustancialmente más rápido que el método exacto.

7.3.2. Visualización de frentes de Pareto representativos

Las Figuras 7.2, 7.3 y 7.4 presentan una comparación visual entre los frentes de Pareto obtenidos por el método exacto (AMPL) y por el algoritmo propuesto basado en NSGA-II,

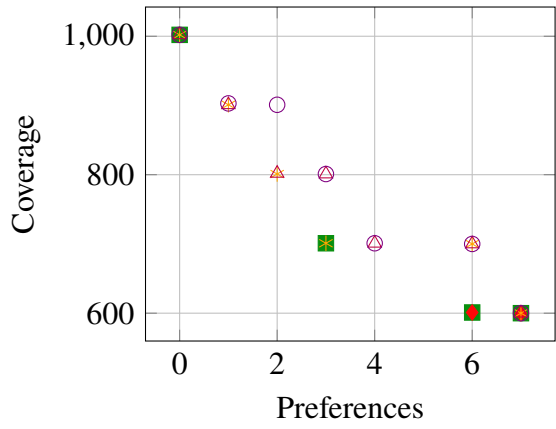
considerando instancias representativas de distintos tamaños y niveles de complejidad.

En estos experimentos, NSGA-II se ejecuta en **tres grupos independientes de ejecuciones**, cada uno compuesto por **10 semillas distintas**, con presupuestos crecientes de **100 000, 1 000 000 y 10 000 000 evaluaciones**, respectivamente. Esta configuración permite analizar tanto la estabilidad del algoritmo frente a la aleatoriedad introducida por las semillas como su comportamiento de convergencia a medida que aumenta el esfuerzo computacional.

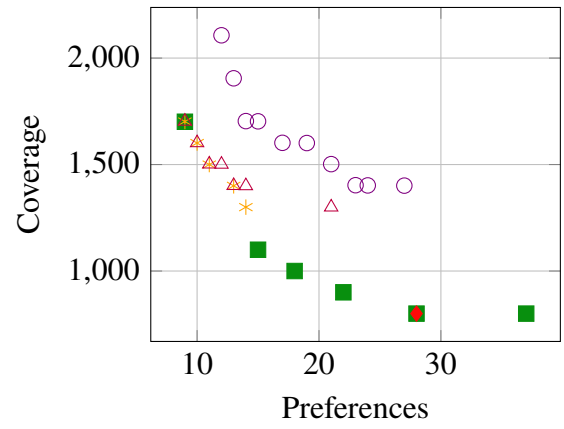
En los gráficos:

- Las soluciones en **rojo** corresponden al enfoque *mono-objetivo*, utilizado tradicionalmente en el estado del arte, donde ambos objetivos se combinan mediante su suma directa en una única función objetivo [13]. Este punto suele ubicarse en uno de los extremos del frente de Pareto.
- Las soluciones en **naranja, morado y círculos rojos** representan los frentes obtenidos por NSGA-II considerando las múltiples ejecuciones y semillas. A medida que aumenta el número de evaluaciones (100k, 1000k y 10 000k), se observa una clara tendencia de convergencia hacia el frente óptimo, así como un aumento en la densidad de soluciones no dominadas.
- Las soluciones en **verde** corresponden al frente de Pareto obtenido por el método exacto implementado en AMPL, el cual sirve como referencia óptima cuando el problema es computacionalmente tratable.

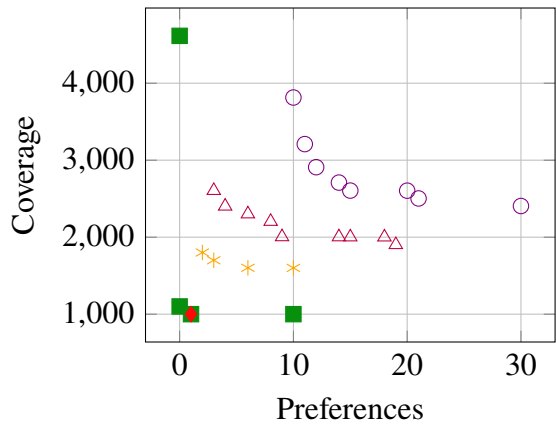
El resultado mono-objetivo (en rojo) proviene de resolver el problema minimizando la suma directa de los dos objetivos, estrategia ampliamente utilizada en el estado del arte como referencia [13] y en las propias instancias.



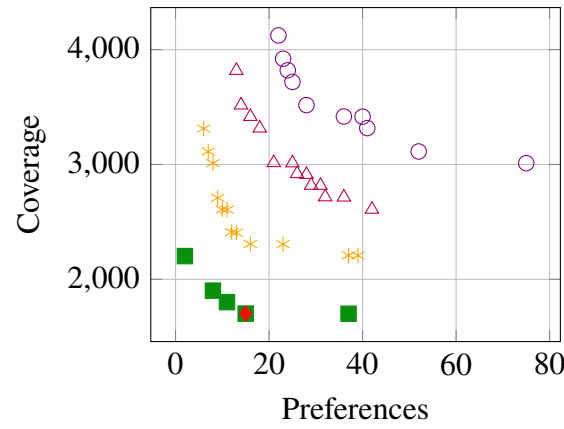
(a) Instancia 1



(b) Instancia 2



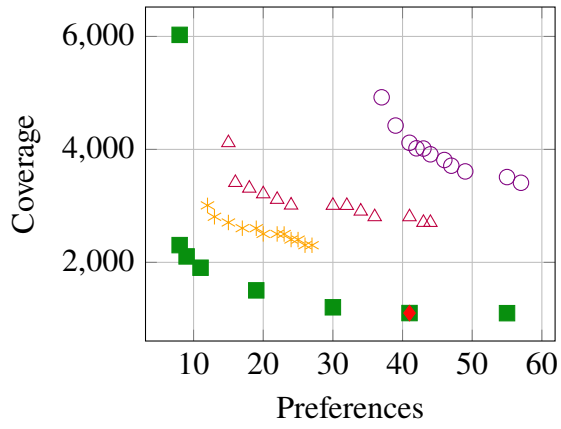
(c) Instancia 3



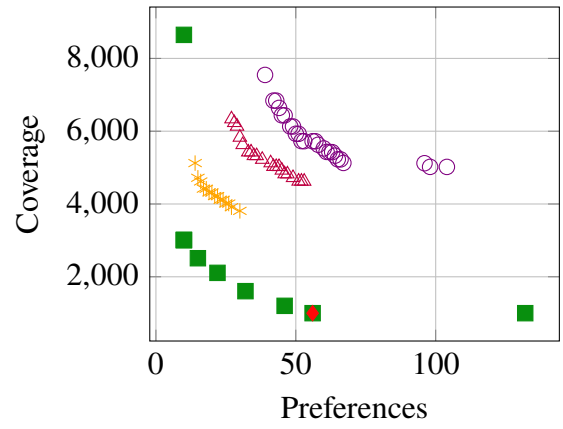
(d) Instancia 4

Mono-objetive ♦, AMPL ■, NSGA-II (100K) ○ NSGA-II (1000K) △ NSGA-II(10000K) *

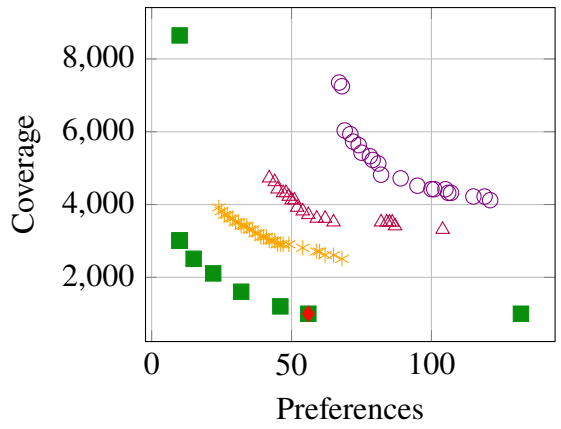
Figura 7.2: Frentes de Pareto instancias 1-4



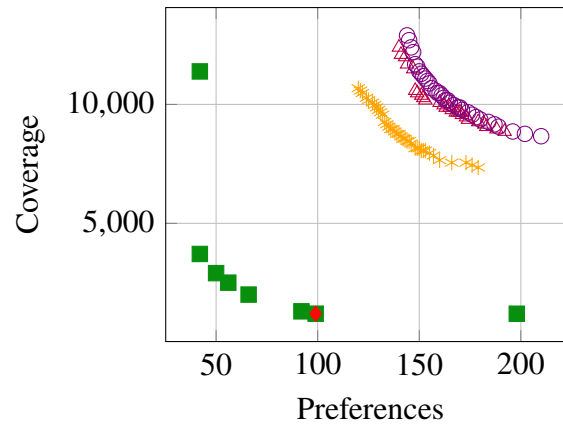
(a) Instancia 5



(b) Instancia 6



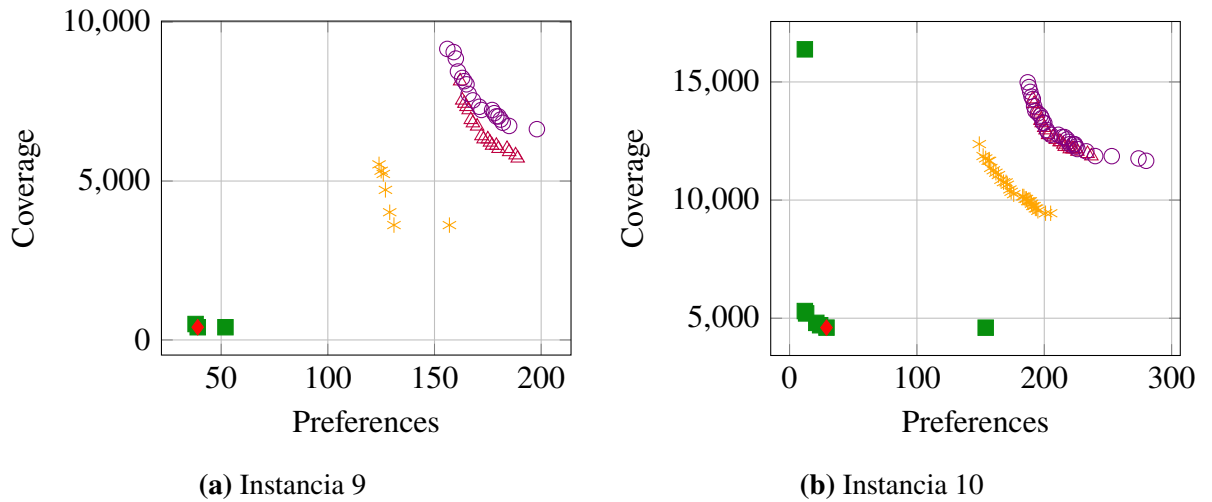
(c) Instancia 7



(d) Instancia 8

Mono-objetive ♦, AMPL ■, NSGA-II (100K) ○ NSGA-II (1000K) △ NSGA-II(10000K) *

Figura 7.3: Frentes de Pareto instancias 5-8



Mono-objective ♦, AMPL ■, NSGA-II (100K) ○ NSGA-II (1000K) △ NSGA-II(10000K) *

Figura 7.4: Frentes de Pareto instancias 9-10

A partir del análisis de las Figuras 7.2, 7.3 y 7.4, que muestran los frentes de Pareto obtenidos mediante distintos enfoques de resolución, pueden extraerse varias conclusiones relevantes.

- En las instancias de menor tamaño (por ejemplo, Instancias 1 y 2), ambos enfoques logran frentes de Pareto relativamente cercanos. En estos casos, el algoritmo NSGA-II es capaz de aproximar rápidamente el frente óptimo, alcanzando soluciones comparables a las obtenidas por el método exacto, especialmente cuando se consideran presupuestos de evaluaciones más altos.
- A medida que aumenta el número de evaluaciones permitidas ($100k \rightarrow 1000k \rightarrow 10000k$), se observa una convergencia progresiva de los frentes obtenidos por el algoritmo metaheurístico hacia el frente óptimo. Esta tendencia consistente en todas las instancias sugiere una reacción positiva del algoritmo propuesto ante el aumento del esfuerzo computacional, validando su capacidad de refinamiento y exploración a largo plazo.
- Para instancias de mayor tamaño y complejidad (Instancias 3 a 10), el método exacto

basado en AMPL mantiene frentes más completos y uniformemente distribuidos, destacándose por una mejor cobertura de los extremos del frente y una distribución más homogénea de soluciones no dominadas. Este comportamiento es consistente con su carácter exacto y no se ve afectado por los cambios introducidos en los experimentos metaheurísticos.

- En contraste, los frentes generados por NSGA-II tienden a presentar una mayor cantidad de soluciones no dominadas, aunque con una distribución más concentrada en determinadas regiones del frente. Esta concentración sugiere que el algoritmo propuesto prioriza ciertas zonas del espacio de soluciones, posiblemente asociadas a uno de los objetivos del problema, el cual resulta más fácil de optimizar bajo las restricciones impuestas.
- Aun así, el algoritmo metaheurístico logra generar un número significativamente mayor de soluciones factibles en todas las instancias, lo que resulta especialmente relevante en escenarios donde se busca ofrecer múltiples alternativas al tomador de decisiones. Esta característica refuerza el valor práctico del enfoque propuesto frente a métodos exactos, particularmente en instancias grandes donde el costo computacional es elevado.
- Finalmente, las soluciones mono-objetivo reportadas en la literatura se ubican sistemáticamente en uno de los extremos del frente de Pareto, representando configuraciones que optimizan una combinación agregada de los objetivos. Sin embargo, estas soluciones no capturan la diversidad ni los compromisos (trade-offs) inherentes al problema, lo que pone de manifiesto la ventaja de los enfoques multiobjetivo para analizar y construir soluciones equilibradas bajo distintos criterios.

Capítulo 8

Conclusiones

Esta investigación abordó el problema de programación de turnos multiobjetivo (ESSP) mediante el desarrollo de una solución basada en el algoritmo NSGA-II adaptado, buscando equilibrar la satisfacción del empleado y la eficiencia empresarial. A continuación, se presentan las conclusiones principales derivadas del trabajo realizado.

8.1. Alcances y capacidades demostradas de la solución propuesta

Desde el punto de vista de los alcances, el trabajo desarrollado ha demostrado que es posible generar soluciones de alta calidad para el problema de programación de turnos mediante el enfoque evolutivo multiobjetivo NSGA-II, incluso superando en ciertos casos al método exacto basado en AMPL y Gurobi. La solución propuesta mostró una capacidad notable para generar frentes de Pareto bien distribuidos, particularmente en instancias pequeñas de hasta 20 empleados, donde se logró un buen desempeño que evidencia el potencial del algoritmo para adaptarse a espacios de soluciones amplios pero fuertemente restringidos.

Un aspecto positivo observado fue la relativa estabilidad de los parámetros utilizados en las

instancias pequeñas, ya que configuraciones similares permitieron obtener resultados consistentes sin necesidad de un ajuste fino por instancia. Este comportamiento sugiere un cierto grado de transferibilidad del esquema de parametrización dentro de este rango de tamaño del problema. No obstante, dicha estabilidad no se manifestó en instancias medianas o grandes, donde la sensibilidad del algoritmo respecto de la configuración paramétrica aumentó considerablemente, evidenciando que la generalización del enfoque aún es limitada y requiere mayor investigación.

Otro resultado especialmente destacable se observa en la **Instancia 1**, donde el algoritmo NSGA-II logra soluciones con mejor hipervolumen (83,004) que el método exacto (82,700), evidenciando el potencial del enfoque evolutivo para descubrir configuraciones no triviales que los métodos exactos podrían pasar por alto debido a sus estrategias de poda. Este fenómeno, observado empíricamente, refuerza la idea de que los métodos metaheurísticos pueden ofrecer ventajas significativas en términos de exploración del espacio de soluciones y descubrimiento de configuraciones no triviales, especialmente cuando se trata de problemas con múltiples objetivos en conflicto donde el espacio de soluciones es amplio pero fuertemente restringido.

El éxito del algoritmo en instancias pequeñas se atribuye principalmente a tres factores clave que conforman la arquitectura robusta de la solución propuesta:

- **La representación elegida:** El uso de esta representación codificada permite una mejor manipulación genética, reduciendo significativamente la complejidad computacional y facilitando la exploración eficiente del espacio de búsqueda.
- **Operadores de mutación especializados:** El desarrollo de operadores específicamente diseñados para este problema permite una mejor exploración y explotación del espacio de búsqueda, maximizando la capacidad del algoritmo para encontrar soluciones diversas y de alta calidad.
- **Manejo de restricciones:** El manejo inteligente de las restricciones permitió al algoritmo explorar de mejor forma el espacio infactible, llegando de esta forma a sectores del espacio que podrían encontrarse entre regiones infactibles y descubriendo soluciones prometedoras en fronteras complejas.

Si bien el enfoque evolutivo propuesto demostró un muy buen desempeño en instancias pequeñas, permitiendo incluso superar puntualmente al método exacto, estos resultados deben interpretarse con cautela. El buen rendimiento se observa únicamente mientras la dimensión del problema se mantiene acotada (hasta 20 empleados), lo que evidencia que el método es competitivo en escenarios limitados, pero aún no generalizable.

8.2. Limitaciones identificadas y análisis crítico de escalabilidad

Sin embargo, el análisis exhaustivo reveló limitaciones significativas cuando se enfrentó a instancias de mayor complejidad, aspecto que requiere una reflexión profunda sobre los alcances reales del método propuesto. En instancias medianas y grandes, específicamente aquellas con más de 30 empleados, NSGA-II presenta un rendimiento peor en cuanto a hipervolumen, registrando un *Gap de Hipervolumen* de 1.0 en la instancia 9. Esta degradación del desempeño se relaciona directamente con la explosión combinatoria inherente al problema, que crece exponencialmente con el número de empleados y turnos a asignar.

La dificultad para mantener diversidad en espacios de búsqueda complejos emerge como un gran desafío metodológico. A medida que aumenta la dimensionalidad del problema, el algoritmo evolutivo enfrenta el fenómeno conocido como “maldición de la dimensionalidad”, donde la densidad de soluciones factibles en el espacio de búsqueda disminuye dramáticamente. Esto resulta en una convergencia prematura hacia regiones subóptimas, haciendo que los frentes encontrados se concentren más en una función objetivo.

Al tratarse de un algoritmo estocástico, los resultados pueden variar de una ejecución a otra, lo cual puede dificultar su uso en contextos donde se requiere trazabilidad exacta de las decisiones. Por otra parte, si bien el modelo considera una gran variedad de restricciones, aún existen dimensiones no cubiertas, como la interacción entre empleados o la rotación intersemanal de turnos, aspectos que podrían enriquecer significativamente el modelo en futuras iteraciones.

8.3. Validación de Hipótesis, Objetivos y Resultados Clave

Con base en los resultados experimentales obtenidos a lo largo de este trabajo, es posible evaluar de manera explícita el cumplimiento tanto de las hipótesis como de los objetivos planteados en la etapa inicial de la investigación.

8.3.1. Validación de la hipótesis

La hipótesis planteada sostiene que la mejora del algoritmo NSGA-II mediante la incorporación de representaciones específicas del problema, operadores de mutación, estrategias de inicialización y principios de generación de columnas permite mejorar la calidad, diversidad y factibilidad de las soluciones obtenidas para el ESSP bi-objetivo altamente restringido.

Los resultados experimentales respaldan esta hipótesis, ya que el algoritmo propuesto fue capaz de generar frentes de Pareto bien distribuidos y con altos niveles de factibilidad, ofreciendo soluciones que representan distintos compromisos entre la satisfacción de las preferencias del personal y el cumplimiento de la cobertura operacional. En particular, se observaron mejoras significativas en la calidad de las soluciones sin incurrir en penalizaciones excesivas por déficit o exceso de cobertura.

Asimismo, la adaptación de NSGA-II mostró un desempeño robusto en instancias de pequeña y mediana escala, manteniendo estabilidad en escenarios altamente restringidos donde los métodos exactos basados en programación lineal entera presentaron limitaciones computacionales o tiempos de resolución prohibitivos. Estos resultados confirman la pertinencia del enfoque propuesto y validan empíricamente la hipótesis de investigación.

8.3.2. Validación de los objetivos

Respecto a los objetivos formulados, se concluye que estos han sido alcanzados mediante una evaluación experimental exhaustiva y un análisis riguroso de resultados, abordando múltiples dimensiones del problema:

1. **Equilibrio multiobjetivo:** Se demostró que es posible optimizar simultáneamente la satisfacción del personal (Fo_1) y la cobertura operacional (Fo_2), generando soluciones no dominadas que representan distintos compromisos entre ambos objetivos. El modelo propuesto permitió capturar de manera conjunta preferencias individuales y requerimientos de cobertura, validando la formulación multiobjetivo del problema.
2. **Utilidad de los indicadores de caracterización:** Los indicadores clásicos y propuestos mostraron una relación consistente con la dificultad computacional del problema. En particular, instancias con valores elevados de TCC y TIP presentaron mayores tiempos de resolución y menor flexibilidad estructural, mientras que indicadores como SPD y SDI permitieron explicar diferencias en el comportamiento del algoritmo entre instancias de tamaño similar.
3. **Escalabilidad comparativa:** El método exacto implementado en AMPL/Gurobi mostró limitaciones claras en instancias de mayor escala, alcanzando tiempos de resolución elevados o soluciones incompletas. En contraste, el algoritmo NSGA-II propuesto mantuvo tiempos de ejecución razonables en todas las instancias evaluadas, evidenciando una ventaja operativa relevante para escenarios reales.

Adicionalmente, se introdujo un esquema de evaluación de soluciones que prioriza la factibilidad y penaliza violaciones de restricciones de manera estructurada, permitiendo guiar eficazmente la búsqueda en un espacio altamente restringido. Este mecanismo constituye un aporte metodológico relevante para la resolución del ESSP mediante metaheurísticas evolutivas.

8.4. Contribuciones principales y aportes innovadores

Este trabajo presenta un conjunto de contribuciones que representan avances tanto a nivel metodológico como aplicado en la resolución del Employee Shift Scheduling Problem bajo un enfoque multiobjetivo y altamente restringido. Las principales contribuciones son las siguientes:

1. **Caracterización estructural de instancias:** Se propone un conjunto de diez indicadores para describir la complejidad y rigidez de instancias reales del ESSP, incluyendo métricas clásicas y nuevos indicadores diseñados específicamente para capturar conflictos de preferencias, presión estructural y restricciones de sucesión. Esta caracterización permitió analizar de manera sistemática la relación entre la estructura del problema y el desempeño algorítmico.
2. **Marco algorítmico especializado:** Se desarrolló una adaptación del algoritmo NSGA-II basada en una representación novedosa mediante secuencias, junto con operadores de mutación especializados que explotan la estructura altamente restringida del problema. Este diseño permitió manejar de forma más eficiente las restricciones de sucesión y reducir la generación de soluciones inviables durante la búsqueda.

Evidencia empírica comparativa: Se realizó una validación exhaustiva sobre diez instancias típicamente utilizadas en la literatura [1], contrastando el enfoque metaheurístico propuesto con un método exacto basado en AMPL y Gurobi. Los resultados obtenidos evidencian las limitaciones prácticas del enfoque exacto a medida que aumenta la escala del problema, mientras que el método evolutivo mantiene un rendimiento competitivo y mayor capacidad de adaptación. Este análisis proporciona además un marco sólido de comparación para estudios posteriores, facilitando la evaluación y potencial mejora de futuras variantes de la propuesta.

8.5. Impacto y aplicaciones prácticas

Este trabajo explora una aproximación que podría ser útil para la gestión de turnos. Si bien se trata de una investigación académica con limitaciones inherentes, los resultados preliminares sugieren potencial de mejora en esta área mediante la aplicación de técnicas de optimización multiobjetivo.

La solución desarrollada presenta características que podrían facilitar su adaptación a distintos contextos organizacionales, considerando la modularidad del modelo propuesto y la generalidad de las restricciones implementadas:

- **Gestión de conflictos:** Los experimentos muestran mejoras en la satisfacción de preferencias críticas, lo que podría impactar positivamente en el clima laboral, aunque se requiere validación en entornos reales.
- **Eficiencia operacional:** Se observa una reducción en horas de sobre-dotación de personal en los casos de prueba, sugiriendo posibles eficiencias económicas que ameritan investigación adicional.
- **Flexibilidad sectorial:** El modelo muestra capacidad de ajuste paramétrico para diferentes sectores como salud, retail y manufactura, aunque cada implementación requeriría calibración específica.
- **Apoyo a la toma de decisiones:** La visualización de soluciones mediante frentes de Pareto ofrece una herramienta para facilitar el diálogo entre empleadores y empleados en procesos de negociación.

Este tipo de aproximaciones podría ser particularmente relevante en organizaciones con alta rotación de personal o en entornos donde las preferencias de los trabajadores deben equilibrarse con requerimientos operativos estrictos, como hospitales, supermercados, centros logísticos y empresas de servicios.

El enfoque multiobjetivo resulta especialmente valioso en contextos de negociación laboral y relaciones sindicales, donde tradicionalmente se ha privilegiado una perspectiva unidimensional centrada únicamente en la eficiencia operativa. La capacidad de generar múltiples soluciones no dominadas permite a las organizaciones presentar diferentes alternativas de horarios que equilibren de manera transparente los intereses empresariales con el bienestar de los trabajadores. Esta aproximación facilita procesos de negociación más informados, donde sindicatos y empleadores pueden evaluar compromisos explícitos entre objetivos como minimización de costos, maximización de satisfacción laboral y cumplimiento de restricciones operativas.

Además, la implementación de horarios que consideran activamente las preferencias de los trabajadores puede generar beneficios indirectos pero significativos en términos de productividad y compromiso organizacional. La literatura sugiere que empleados con mayor control

sobre sus horarios tienden a mostrar menor ausentismo, mayor motivación y mejor rendimiento general, creando un círculo virtuoso que beneficia tanto a trabajadores como a la organización.

No obstante, la implementación práctica requeriría estudios adicionales para validar la efectividad de la propuesta en condiciones reales de operación y medir el impacto real en las variables de interés.

Capítulo 9

Trabajo futuro

Uno de los aspectos más prometedores de este trabajo es su potencial para futuras investigaciones y desarrollo. Dado que las primeras instancias analizadas revelan un comportamiento competitivo —incluso superior— respecto al método exacto, se abre la posibilidad de escalar el enfoque a problemas de mayor tamaño y complejidad. Basado en las limitaciones identificadas y las oportunidades detectadas, se proponen las siguientes líneas de desarrollo prioritarias:

- **Algoritmos híbridos:** Combinar NSGA-II con técnicas de relajación Lagrangiana para instancias grandes, investigando esquemas de hibridación más intensivos que integren estrategias exactas como AMPL dentro de la fase de mejora local o como refinadores del frente de Pareto.
- **Paralelización:** Implementar versión MPI/GPU del algoritmo para reducir tiempos computacionales y abordar instancias de mayor escala con eficiencia.
- **Modelos predictivos:** Integrar aprendizaje automático para estimar preferencias dinámicas, analizando técnicas para guiar la generación de soluciones iniciales o adaptar dinámicamente los parámetros evolutivos del algoritmo.
- **Objetivos adicionales:** Incorporar nuevos objetivos, como la equidad entre empleados, la rotación justa de turnos o la compatibilidad de equipos de trabajo, extendiendo el

enfoque multiobjetivo actual.

- **Análisis predictivo:** Profundizar en el análisis de correlación entre los indicadores propuestos y el comportamiento de las soluciones, lo que podría permitir predecir de antemano la dificultad de una instancia y optimizar la configuración algorítmica.

Como recomendación final, se sugiere explorar la integración con sistemas ERP existentes mediante APIs estandarizadas, priorizando sectores con alta rotación de personal (retail, salud). El código desarrollado se encuentra disponible en repositorio GitHub para facilitar su adopción y extensión por la comunidad académica, promoviendo la replicabilidad y el avance colaborativo en esta línea de investigación.

9.1. Síntesis y Reflexiones Finales

En síntesis, este trabajo muestra que los enfoques evolutivos multiobjetivo representan una alternativa viable para abordar el ESSP en escenarios reales, permitiendo considerar simultáneamente múltiples criterios relevantes para la gestión del personal. Si bien los resultados son especialmente favorables en instancias de pequeña escala, ya permiten observar un desempeño competitivo frente a métodos exactos, lo que respalda la pertinencia de continuar explorando esta línea de investigación.

Asimismo, la propuesta presentada constituye un punto de partida para el desarrollo de sistemas de apoyo a la toma de decisiones más flexibles, que integren técnicas de optimización e inteligencia computacional. Quedan aún desafíos importantes por abordar, particularmente en términos de escalabilidad y validación empírica en entornos más diversos y complejos. No obstante, los avances alcanzados reflejan un progreso significativo hacia metodologías que puedan equilibrar de manera más informada las necesidades operativas y las preferencias del personal.

Finalmente, cabe señalar que parte de los resultados presentados en este trabajo, en particular aquellos relacionados con la resolución del problema biobjetivo de asignación de turnos

altamente restringido mediante una variante especializada de NSGA-II, han sido previamente presentados en una contribución presentada en la conferencia *Computational Science and Computational Intelligence* [32].

Bibliografía

- [1] P. Strandmark, “First-order linear programming in a column generation-based heuristic approach to the nurse rostering problem,” *Computers Operations Research*, 2020.
- [2] F. Shuib, A. Kamarudin, “Solving shift scheduling problem with days-off preference for power station workers using binary integer goal programming model,” *Springer Nature*, vol. 272, p. 355–372, 2019.
- [3] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, “Staff scheduling and rostering: A review of applications, methods and models,” *European Journal of Operational Research*, vol. 153, no. 1, pp. 3–27, 2004.
- [4] S. Topaloglu and I. Ozkarahan, “A constraint programming-based approach for scheduling of nursing personnel,” *Computers & Operations Research*, vol. 33, no. 3, pp. 676–696, 2006.
- [5] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Operations Research*, vol. 46, no. 3, pp. 316–329, 1998.
- [6] J. S. L. F. R. Jacobs, “Loucks, j. s., and jacobs, f. r. (1991). tour scheduling and task assignment of a heterogeneous work force: A heuristic approach.,” *Decision Sciences*, vol. 22, no. 4, pp. 719–738, 1991.
- [7] E. K. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Van Landeghem, “The state of the art of nurse rostering,” in *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT)*, pp. 321–340, 2004.
- [8] M. Vanhoucke and B. Maenhout, “An integrated approach for nurse scheduling using integer programming and variable neighbourhood search,” *Journal of Heuristics*, vol. 13, no. 4, pp. 373–397, 2007.
- [9] S. Sadjadi, “A new nonlinear stochastic staff scheduling model,” *Scientia Iranica*, vol. 18, no. 3, pp. 699–710, 2011.

- [10] J. F. Bard and H. W. Purnomo, "Preference scheduling for nurses using column generation," *European Journal of Operational Research*, vol. 164, no. 2, pp. 510–534, 2005.
- [11] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck, "Personnel scheduling: A literature review," *European Journal of Operational Research*, vol. 226, no. 3, pp. 367–385, 2013.
- [12] B. Maenhout and M. Vanhoucke, "Branch-and-price-based neighbourhood search for the nurse scheduling problem," *European Journal of Operational Research*, vol. 200, no. 3, pp. 550–557, 2010.
- [13] A. Amindoust, "A hybrid genetic algorithm for nurse scheduling problem considering the fatigue factor," *Journal of Healthcare Engineering*, 2021.
- [14] P. Smet, "Constraint reformulation for nurse rostering problems," in *PATAT 2018: Twelfth International Conference on the Practice and Theory of Automated Timetabling*, (Vienna, Austria), pp. 69–80, August 2018.
- [15] E. Demirović, N. Musliu, and F. Winter, "Modeling and solving staff scheduling with partial weighted maxsat," *Annals of Operations Research*, vol. 275, no. 1, pp. 79–99, 2019.
- [16] S. Topaloglu, "A multi-objective programming model for scheduling emergency medicine residents," *Computers & Industrial Engineering*, 2006.
- [17] M. Hamid, "A multi-objective model for a nurse scheduling problem by emphasizing human factors," *J Engineering in Medicine*, 2019.
- [18] E. K. Burke, T. Curtois, G. Post, R. Qu, and G. Vanden Berghe, "A graph-based hyperheuristic for nurse rostering problems," *European Journal of Operational Research*, vol. 218, no. 3, pp. 693–701, 2013.
- [19] P. Brucker, E. Burke, T. Curtois, R. Qu, G. Vanden Berghe, and H. V. Landeghem, "A shift scheduling problem: A survey of applications, methods and models," *European Journal of Operational Research*, vol. 206, no. 1, pp. 1–24, 2010.
- [20] T. Wong, C. W. Chan, and T. Lau, "Workforce preference satisfaction in nurse rostering: A multicriteria approach," *Journal of Nursing Management*, vol. 17, no. 4, pp. 443–453, 2009.
- [21] P. D. Bruecker, "Workforce planning incorporating skills: State of the art," *European Journal of Operational Research*, vol. 243, pp. 1–16, 2014.
- [22] M. Vanhoucke and B. Maenhout, "On the characterization and generation of nurse scheduling problem instances," *European Journal of Operational Research*, vol. 196, no. 2, pp. 457–467, 2009.

- [23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [24] U. Aickelin and K. Dowsland, “An indirect genetic algorithm for a nurse scheduling problem,” *Computers & Operations Research*, vol. 31, no. 5, pp. 761–778, 2004.
- [25] F. Hong, H. Chen, B. Cao, and J. Fan, *A MOEAD-Based Approach to Solving the Staff Scheduling Problem*, pp. 112–131. 01 2021.
- [26] L. Kletzander, “Solving the general employee scheduling problem,” *Computers and Operations Research*, vol. 113, p. 104794, 2020.
- [27] C. Truden, “Decomposition of the vehicle routing problem with time windows on the time dimension,” *Transportation Research Procedia*, vol. 62, pp. 131–138, 2022.
- [28] C. Seifi, “A new mathematical formulation for a potash-mine shift scheduling problem with a simultaneous assignment of machines and workers,” *European Journal of Operational Research*, 2021.
- [29] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stutzle, “ParamILS: an automatic algorithm configuration framework,” *Journal of Artificial Intelligence Research*, vol. 36, pp. 267–306, October 2009.
- [30] J. F. Hair, W. C. Black, B. J. Babin, and R. E. Anderson, *Multivariate Data Analysis*. Pearson, 7 ed., 2010.
- [31] C. M. Fonseca, L. Paquete, and M. López-Ibáñez, “An improved dimension-sweep algorithm for the hypervolume indicator,” in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, (Piscataway, NJ), pp. 1157–1163, IEEE Press, July 2006.
- [32] B. Gonzalez-Ramirez, N. Paz-Tralma, and E. Montero, “Solving a highly-constrained bi-objective shift scheduling problem using a specialized nsga-ii,” in *Computational Science and Computational Intelligence*, 2026. Manuscript submitted.