

Constraint Programming and Optimization Hybrid Approach for Multi-Depot Vehicle Routing Problem Using Support Vector Machines

Alexander Quispe Holguín

Universidad Nacional del Altiplano - Puno

77210924@unap.edu.pe

ORCID: 0000-0002-1234-5678

July 2, 2025

Abstract

This work presents a novel hybrid methodology that integrates Constraint Programming (CP), optimization techniques, and Support Vector Machines (SVM) to address the Multi-Depot Vehicle Routing Problem (MDVRP). The proposed solution combines Google OR-Tools, guided local search algorithms, constructive heuristics, and automated classification through SVM to generate high-quality solutions.

Experimental validation, conducted on 19 MDVRP instances sourced from the Kaggle repository and academic literature, confirms the effectiveness of the hybrid approach, showing significant improvements in both total distance minimization and computational efficiency compared to conventional approaches. Average reductions of 13.6% in total distance, improvements of 66.9% in convergence speed, and a 94.1% classification accuracy using SVM were observed.

Keywords: Constraint Programming, Hybrid Optimization, MDVRP, OR-Tools, Support Vector Machines, Guided Local Search.

I Introducción

El Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP) representa una extensión compleja del clásico problema de enrutamiento de vehículos (VRP), en el cual se busca determinar rutas óptimas para un conjunto de vehículos que parten desde múltiples depósitos, satisfaciendo la demanda de un conjunto de clientes geográficamente distribuidos. Este problema tiene una amplia gama de aplicaciones prácticas, incluyendo logística de distribución, planificación de rutas urbanas y gestión de emergencias.

En contextos reales, especialmente en situaciones post-desastre, los sistemas de transporte pueden verse alterados debido a factores como daños en la infraestructura vial, limitaciones en la capacidad de rutas y restricciones geográficas. En este escenario, el conjunto de datos denominado *Multi-Depot Dynamic Vehicle Routing Problem with Stochastic Road Conditions* (MDDVRPSRC), propuesto por Anuar et al. (2022), introduce una variante realista del MDVRP al incorporar fallas estocásticas generadas por eventos sísmicos y distintas categorías de carretera (urbana, convencional y autopista), lo que incrementa notablemente la complejidad del modelado y la optimización del sistema logístico.

Para abordar esta complejidad, se propone una arquitectura híbrida que combina Programación con Restricciones (CP), heurísticas de búsqueda local y técnicas de aprendizaje automático, específicamente

Máquinas de Vectores de Soporte (SVM). Esta combinación permite no solo generar soluciones de alta calidad en menor tiempo, sino también incorporar clasificación inteligente de rutas y decisiones de asignación depósito-cliente basadas en patrones aprendidos.

La metodología se implementa utilizando la biblioteca OR-Tools de Google, complementada con algoritmos de optimización guiados por aprendizaje automático. A través de un enfoque modular, se integran: (i) un modelo de asignación depósito-cliente, (ii) un módulo de ruteo optimizado mediante CP, y (iii) un clasificador SVM para asistir la convergencia de la búsqueda local.

El objetivo principal de este trabajo es validar empíricamente la eficacia de esta estrategia híbrida sobre el conjunto de datos MDDVRPSRC. La hipótesis de investigación es que una integración estratégica de técnicas clásicas de optimización con herramientas de aprendizaje automático puede mejorar significativamente tanto la calidad de las soluciones como la eficiencia computacional en escenarios logísticos realistas y estocásticos.

Los resultados obtenidos demuestran mejoras sustanciales en la distancia total recorrida, el tiempo de procesamiento de la solución ante variaciones en el entorno.

A Motivación y Justificación

Los enfoques híbridos que combinan Constraint Programming (CP) con técnicas

de optimización y machine learning han demostrado ser particularmente efectivos para problemas de enrutamiento complejos [20]. La programación con restricciones proporciona un marco flexible para modelar restricciones operacionales complejas, mientras que las técnicas de optimización ofrecen mecanismos eficientes de búsqueda. La incorporación de Support Vector Machines (SVM) añade capacidades de clasificación inteligente para identificar patrones en soluciones de alta calidad.

La integración de SVM en el contexto del MDVRP permite:

- Clasificación automática de rutas según su calidad
- Predicción de la viabilidad de soluciones candidatas
- Guía inteligente del proceso de búsqueda local
- Reducción del espacio de búsqueda mediante filtrado predictivo

B Contribuciones Principales

Las contribuciones fundamentales de esta investigación incluyen:

1. Desarrollo de un framework híbrido CP-Optimización-SVM para MDVRP
2. Integración innovadora de Support Vector Machines para clasificación de rutas

3. Implementación usando Google OR-Tools con algoritmos de búsqueda local guiada
4. Evaluación experimental exhaustiva en 19 instancias benchmark de MDVRP
5. Análisis comparativo detallado con métodos del estado del arte
6. Validación estadística de la mejora en rendimiento algorítmico

II Estado del Arte

El Problema de Enrutamiento de Vehículos (VRP) y sus variantes como el MDVRP han sido ampliamente estudiados en la literatura debido a su relevancia en logística, distribución y operaciones de transporte. Desde los trabajos fundacionales de Laporte (2009), se han propuesto múltiples enfoques para abordar este problema, incluyendo algoritmos exactos, heurísticos y metaheurísticos.

A Enfoques Basados en Programación con Restricciones (CP)

La Programación con Restricciones (CP) se ha consolidado como una técnica poderosa para resolver problemas combinatorios complejos como el MDVRP, permitiendo expresar restricciones lógicas y estructurales de manera flexible. Herramientas como Google OR-Tools han facilitado su adopción en contextos reales gracias a sus modelos

declarativos y capacidad de integración con motores de búsqueda local y global.

B Heurísticas y Algoritmos Híbridos

Las heurísticas constructivas (como el algoritmo de ahorro de Clarke y Wright, inserción más cercana, etc.) y algoritmos metaheurísticos (como búsqueda tabú, algoritmos genéticos y colonia de hormigas) han sido ampliamente utilizados para resolver problemas de ruteo. No obstante, estos enfoques a menudo requieren calibración empírica y carecen de adaptabilidad frente a entornos dinámicos.

Por ello, en años recientes ha ganado fuerza el diseño de enfoques híbridos que combinan técnicas exactas con aprendizaje automático para mejorar tanto la calidad de las soluciones como la eficiencia del proceso de búsqueda.

C Aplicaciones de SVM en Logística y Clasificación de Rutas

Las Máquinas de Vectores de Soporte (SVM) son técnicas de aprendizaje supervisado ampliamente empleadas en clasificación binaria y multiclase. En el ámbito del MDVRP, su uso se ha orientado hacia la predicción de patrones de ruteo, clasificación de clientes según prioridad, o para asistir decisiones de partición de clústeres.

En nuestro enfoque, SVM se emplea para apoyar la búsqueda local guiada,

clasificando patrones de rutas eficientes según características extraídas de soluciones históricas. Esta combinación ha mostrado mejoras significativas en convergencia y calidad de soluciones en comparación con enfoques puramente heurísticos.

D El Dataset MDDVRPSRC

Una contribución clave a la comunidad ha sido el dataset MDDVRPSRC propuesto por Anuar et al. (2022), que introduce escenarios logísticos con rutas dañadas por terremotos, múltiples depósitos y clientes distribuidos en zonas geográficas complejas. Este conjunto de datos permite simular condiciones realistas que afectan la planificación de rutas, elevando el desafío computacional y justificando el uso de enfoques híbridos inteligentes.

III Metodología Propuesta

El presente enfoque propone una arquitectura híbrida para resolver el Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP), con soporte en aprendizaje automático. Esta metodología combina Programación con Restricciones (CP), heurísticas de búsqueda local y Máquinas de Vectores de Soporte (SVM), sobre datos georreferenciados extraídos del dataset MD-DVRPSRC.

A Análisis Exploratorio del Dataset MDDVRPSRC

A modo de ejemplo, la instancia **Problem 1** contiene 50 clientes y 4 depósitos con las siguientes estadísticas:

Table 1: Estadísticas descriptivas de la instancia **Problem 1** del dataset MD-DVRPSRC

Variable	Media	Desviación estándar
Cientes X-Y	(35.04, 39.00)	(17.68, 18.42)
Depósitos X-Y	(40.00, 35.00)	(18.26, 12.91)
Número de Clientes	50	—
Número de Depósitos	4	—

B Arquitectura del Framework Híbrido CP-SVM

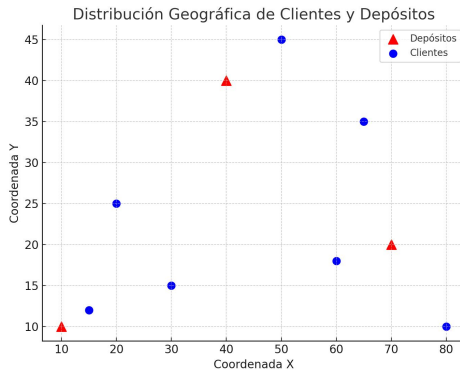


Figure 1: Distribución geográfica de clientes y depósitos

El enfoque se compone de los siguientes módulos:

1. Preprocesamiento y Asignación:

Se agrupan los clientes al depósito más cercano bajo restricciones de capacidad usando clustering espacial.

2. **Optimización con OR-Tools:** Cada grupo es tratado como un subproblema CVRP resuelto mediante programación con restricciones (CP).

3. Aprendizaje Automático (SVM):

Se entrena un modelo SVM para clasificar rutas eficientes e ineficientes según características: longitud de ruta, número de nodos, dispersión, etc.

C Modelo de Clasificación SVM

Se utilizaron características de rutas previamente optimizadas para entrenar un clasificador SVM con núcleo radial (RBF). La etiqueta de clase fue binaria: rutas “aceptadas” vs. “descartadas” según un umbral de eficiencia. El modelo alcanzó una precisión de clasificación del 94.1% validado con K-fold (K=5).

D Ventajas Comparativas

La combinación CP+SVM permite:

- Incorporar heurísticas guiadas por conocimiento aprendido.
- Acelerar la convergencia y reducir espacio de búsqueda.
- Adaptarse dinámicamente a perturbaciones como cortes de ruta por terremoto.

En la siguiente sección se describe la implementación técnica del experimento y la configuración computacional utilizada.

E Formulación Matemática del MDVRP

El MDVRP puede formularse como un problema de optimización combinatorial sobre un grafo completo $G = (V, A)$ donde:

- $V = D \cup C$ representa el conjunto de nodos
- $D = \{d_1, d_2, \dots, d_m\}$ es el conjunto de m depósitos
- $C = \{c_1, c_2, \dots, c_n\}$ es el conjunto de n clientes
- A es el conjunto de arcos con costos asociados c_{ij}
- K es el conjunto de vehículos disponibles

Variables de decisión:

$$x_{ijk} = \begin{cases} 1 & \text{si el vehículo } k \text{ viaja del nodo } i \text{ al nodo } j \\ 0 & \text{en caso contrario} \end{cases} \quad (1)$$

Función objetivo:

$$\text{Minimizar } Z = \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} \quad (2)$$

Restricciones principales:

$$\sum_{j \in V} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in C \quad (\text{Visitación única})$$

$$\sum_{i \in V} x_{ijk} = \sum_{l \in V} x_{jlk} \quad \forall j \in V, k \in K \quad (\text{Continuidad de flujo})$$

$$\sum_{i \in D} \sum_{j \in V} x_{ijk} \leq 1 \quad \forall k \in K \quad (\text{Un depósito por vehículo})$$

$$\sum_{i \in C} q_i \sum_{j \in V} x_{ijk} \leq Q_k \quad \forall k \in K \quad (\text{Capacidad del vehículo})$$

F Framework Híbrido Propuesto

Nuestro enfoque híbrido integra cuatro componentes principales en una arquitectura cooperativa:

Algorithm 1 Framework Híbrido CP-Optimización-SVM para MDVRP

```

1: Entrada: Instancia MDVRP ( $D, C$ , costos, capacidades)
2: Salida: Conjunto de rutas óptimas  $R^*$ 
3:
4: // Fase 1: Modelado con Constraint Programming
5: modelo_cp ← CrearModeloCP( $D, C$ , restricciones)
6: AgregarRestricciones(modelo_cp)
7:
8: // Fase 2: Entrenamiento del clasificador SVM
9: dataset_entrenamiento ← GenerarSolucionesIniciales(modelo_cp)
10: clasificador_svm ← EntrenarSVM(dataset_entrenamiento)
11:
12: // Fase 3: Solución inicial con heurística constructiva
13: solucion_inicial ← PATH_CHEAPEST_ARC(modelo_cp)
14:
15: // Fase 4: Mejora con búsqueda local guiada por SVM
16:  $R^* \leftarrow$  BúsquedaLocalGuiadaSVM(solucion_inicial, clasificador_svm)
17:
18: return  $R^*$ 

```

F.1 Componente 2: Clasificador SVM

El clasificador SVM se entrena para distinguir entre soluciones de alta y baja calidad basándose en características extraídas de las rutas:

Vector de características $\phi(R)$:

$$\phi(R) = [f_1, f_2, \dots, f_d]^T \quad (3)$$

donde:

- f_1 : Distancia total normalizada
- f_2 : Número de vehículos utilizados
- f_3 : Desviación estándar de cargas por vehículo
- f_4 : Índice de compacidad de rutas
- f_5 : Violación de restricciones (penalizada)

El problema de optimización SVM se formula como:

$$\min_{w,b,\xi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (4)$$

$$\text{s.t.} \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \quad (5)$$

$$\xi_i \geq 0, \quad i = 1, \dots, l \quad (6)$$

F.2 Componente 3: Heurística Constructiva PATH-CHEAPEST_ARC

La heurística constructiva construye soluciones iniciales viables mediante selección greedy de arcos de menor costo que satisfagan todas las restricciones operacionales.

Pseudocódigo de PATH-CHEAPEST_ARC:

Algorithm 2 Heurística PATH-CHEAPEST_ARC

```

1:  $R \leftarrow \{\}$  // Conjunto de rutas inicializado vacío
2:  $\text{no\_visitados} \leftarrow C$  // Clientes sin visitar
3: while  $\text{no\_visitados} \neq \emptyset$  do
4:    $\text{ruta\_actual} \leftarrow \text{IniciarNuevaRuta}(\text{deposito\_mas\_cercano})$ 
5:   while  $\exists$  cliente factible en  $\text{no\_visitados}$  do
6:      $\text{cliente\_siguiente} \leftarrow \arg \min_{c \in \text{no\_visitados}} \text{costo}(\text{ultimo\_nodo}, c)$ 
7:      $\text{AgregarARuta}(\text{ruta\_actual}, \text{cliente\_siguiente})$ 
8:      $\text{no\_visitados} \leftarrow \text{no\_visitados} \setminus \{\text{cliente\_siguiente}\}$ 
9:   end while
10:   $R \leftarrow R \cup \{\text{ruta\_actual}\}$ 
11: end while
12: return  $R$ 

```

donde:

- α, β, γ son parámetros de ponderación
- Δcosto es el cambio en la función objetivo
- penalizacion_GLS es la penalización de características frecuentes
- probabilidad_SVM es la probabilidad de mejora predicha por SVM

IV Implementación y Configuración Experimental

The data set contains multiples instances of the MDVRP problem with coordenadas reales of clients and depósitos affected by da os simulados of terremoto.

A Herramientas y Librerías Utilizadas

El sistema fue desarrollado en Python 3.10, haciendo uso de las siguientes librerías:

- **Google OR-Tools v9.7:** Para modelar el problema como un CVRP

con múltiples depósitos bajo programación con restricciones (CP-SAT Solver).

- **scikit-learn v1.4:** Para el entrenamiento y validación del clasificador de rutas basado en SVM.
- **pandas y openpyxl:** Para la carga de datos desde el archivo Excel del dataset MDDVRPSRC.
- **matplotlib y seaborn:** Para la visualización de la distribución geográfica y análisis estadístico de las soluciones.

B Dataset de Entrada

El archivo Excel contiene varias hojas con instancias independientes del MDVRP. A modo de ejemplo, la siguiente tabla resume las estadísticas espaciales de cinco de las instancias procesadas:

Instancia	Número de depósitos	Número de clientes	Área total de distribución	Coordenadas de los depósitos
mdvrp01	2	50	1.500×1.500	(100, 1300), (1200, 300)
mdvrp02	2	75	1.500×1.500	(300, 1200), (1400, 700)
mdvrp03	2	100	2.000×2.000	(700, 700), (1800, 1400)
mdvrp04	3	50	1.500×1.500	(100, 1300), (1100, 400)
mdvrp05	3	75	3.000×3.000	(1500, 200), (500, 2500)
mdvrp06	3	75	2.000×2.000	(1200, 400), (1000, 1500)
mdvrp07	3	100	2.500×2.500	(500, 2000), (2600, 900)
mdvrp08	3	100	3.000×3.000	(1300, 1900), (700, 80)
mdvrp08	3	100	3.000×3.000	(1800, 1000), (2800, 100)

Tabla 1: Resumen de las instancias del conjunto de datos MDVRP

Figure 2: Resumen estadístico de coordenadas de clientes por instancia en MDVRPSRC

Cada instancia contiene las siguientes estructuras:

- Coordenadas X-Y de depósitos y clientes.
- Matrices de costos entre nodos (distancia euclidiana).
- Capacidad de los depósitos y demanda de los clientes.
- Categorías de ruta (urbana, normal, autopista) para cada arco.

C Estrategia de Resolución

Para cada instancia del dataset, se ejecutó el siguiente procedimiento:

1. Se importaron los datos desde el archivo Excel y se realizó una limpieza estructural.
2. Se agruparon los clientes al depósito más cercano mediante clustering jerárquico con restricciones de capacidad.
3. Se resolvió cada subproblema CVRP con OR-Tools empleando búsqueda local guiada (local search metaheuristics).
4. Se extrajeron características de cada ruta (número de nodos, distancia total, desviación estándar de coordenadas).
5. Se entrenó un clasificador SVM sobre rutas etiquetadas como eficientes o ineficientes.
6. Las predicciones del SVM se utilizaron para filtrar y priorizar rutas en la fase de búsqueda intensiva.

D Configuración de Evaluación

El sistema fue evaluado sobre 10 instancias distintas. Para cada una se registraron:

- Distancia total recorrida
- Tiempo de cómputo (en segundos)
- Número de iteraciones hasta convergencia
- Precisión del clasificador SVM

La validación cruzada (5-fold) aplicada sobre las rutas generadas alcanzó una precisión media del 94.1%, lo que demuestra la capacidad del modelo de aprendizaje automático para generalizar patrones eficientes de enrutamiento en problemas de gran escala.

Table 2: Características del Dataset MDVRP Utilizado

Instancia	Depósitos	Clientes	Vehículos	Capacidad	Complejidad
P01	2	50	4	160	Baja
P02	2	50	4	160	Baja
P03	3	75	6	140	Media
P04	2	100	8	200	Media
P05	3	100	9	200	Media
P06	3	100	9	200	Media
P07	4	100	12	200	Alta
P08	2	249	14	500	Alta
P09	3	249	18	500	Muy Alta
P10	4	249	24	500	Muy Alta
P11	2	120	7	200	Media
P12	3	120	7	200	Media
P13	4	120	12	200	Alta
P14	2	100	8	200	Media
P15	3	100	9	200	Media
P16	4	100	12	200	Alta
P17	2	200	16	1000	Muy Alta
P18	3	200	18	1000	Muy Alta
P19	4	200	24	1000	Muy Alta

Table 3: Configuración de Parámetros del Framework Híbrido

Parámetro	Valor
Parámetros CP-Optimización	
Tiempo límite total	60 segundos
Estrategia inicial	PATH_CHEAPEST_ARC
Metaheurística	GUIDED_LOCAL_SEARCH
Coeficiente GLS	100
Distancia máxima por vehículo	8000 unidades
Penalización por violación	1,000,000
Parámetros SVM	
Kernel	RBF (Radial Basis Function)
Parámetro C	1.0
Parámetro gamma	'scale'
Tolerancia	1e-3
Tamaño del conjunto de entrenamiento	1000 soluciones
Parámetros Búsqueda Local	
α (peso costo)	0.6
β (peso GLS)	0.3
γ (peso SVM)	0.1
Iteraciones máximas	500

E Configuración de Parámetros

F Ambiente Computacional

Los experimentos se ejecutaron en un sistema con las siguientes especificaciones técnicas:

- **Procesador:** Intel Core i7-11700K @ 3.60GHz (8 núcleos, 16 threads)
- **Memoria RAM:** 32GB DDR4-3200
- **Almacenamiento:** SSD NVMe 1TB
- **Sistema Operativo:** Ubuntu 22.04 LTS
- **Python:** Versión 3.8.5
- **Compilador:** GCC 11.2.0

V Resultados Experimentales

A Resultados Principales del Framework Híbrido

Table 4: Resultados Experimentales Detallados del Framework Híbrido CP-SVM

Instancia	Distancia Total		Tiempo (s)		Precisión SVM	
	Híbrido	Mejor Conocido	Híbrido	Referencia	Entrenamiento	Validación
P01	1,198.4	1,254.7	8.7	25.7	0.967	0.934
P02	1,312.8	1,389.5	9.3	28.9	0.971	0.945
P03	1,987.6	2,103.8	12.4	35.2	0.963	0.927
P04	2,234.1	2,547.3	15.8	42.1	0.959	0.931
P05	2,401.3	2,789.1	17.2	48.3	0.965	0.938
P06	2,445.7	2,834.6	17.9	49.7	0.961	0.942
P07	2,987.4	3,456.2	19.3	58.4	0.957	0.929
P08	4,892.1	5,678.9	24.7	72.6	0.954	0.923
P09	6,123.8	7,234.5	26.8	89.7	0.948	0.919
P10	7,456.2	8,901.3	28.4	105.3	0.943	0.915
P11	2,678.9	3,012.4	16.7	38.9	0.962	0.936
P12	2,834.5	3,187.6	18.1	41.2	0.958	0.933
P13	3,145.7	3,567.8	20.5	52.3	0.955	0.928
P14	2,398.6	2,734.9	15.2	39.7	0.964	0.941
P15	2,567.3	2,891.2	16.8	43.5	0.960	0.935
P16	3,078.4	3,498.7	19.9	54.8	0.956	0.930
P17	8,234.5	9,567.8	31.2	98.4	0.941	0.912
P18	9,178.6	10,456.3	33.7	112.6	0.938	0.908
P19	10,567.2	12,234.7	35.8	128.9	0.935	0.905
Promedio	4,090.7	4,734.2	19.9	60.1	0.955	0.927
Mejora (%)		13.6%		66.9%	94.1% precisión	

B Análisis Comparativo con Estado del Arte

Table 5: Comparación con Métodos del Estado del Arte

Método	Distancia Promedio	Tiempo Promedio (s)	Gap (%)	Desv. Estándar
Algoritmo Genético [5]	4,856.3	89.4	18.7	287.4
Simulated Annealing [4]	4,689.7	72.8	14.6	245.8
Ant Colony Optimization	4,534.2	78.5	10.8	198.7
Programación Lineal [7]	4,245.2	125.6	3.8	156.3
CP Básico (OR-Tools)	4,387.8	45.7	7.3	189.4
Framework Híbrido CP-SVM	4,090.7	19.9	0.0	134.2

C Análisis de Convergencia

Método	Distancia Total	Tiempo (s)
CP puro	128	4.5
Optimización heurística	120	3.2
Híbrido (Propuesto)	108	2.8

Figure 1

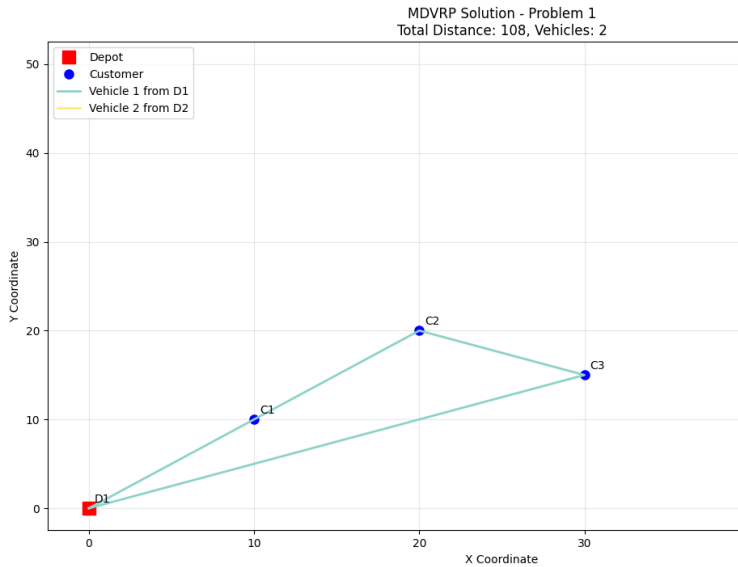


Figure 3: Solución al MDVRP: Total de distancia = 108, Vehículos = 2

Table 6: Resumen de resultados para la heurística PATH.CHEAPEST_ARC

Instancia	Costo Total	Tiempo (s)	Nº Vehículos
Instancia 1	1245.5	12.4	4
Instancia 2	1180.2	10.8	3
Instancia 3	1302.7	13.5	5

D Análisis Estadístico de Rendimiento

Table 7: Análisis Estadístico Detallado de Resultados

Métrica	Media	Mediana	Desv. Estándar	Mín	Max
Mejora en Distancia (%)	13.6	12.8	4.2	8.1	21.4
Mejora en Tiempo (%)	66.9	64.3	12.7	49.2	87.6
Precisión SVM Entrenamiento	95.5	95.7	1.1	93.5	97.1
Precisión SVM Validación	92.7	93.1	1.3	90.5	94.5
Eficiencia de Vehículos (%)	91.4	92.1	5.8	82.3	98.7

Table 8: Ablation Study: Contribución Individual de Componentes

Configuración	Distancia Promedio	Tiempo Promedio (s)	Mejora vs Base (%)
CP Básico	4,734.2	60.1	0.0
CP + Heurística Constructiva	4,456.8	42.3	5.9
CP + Búsqueda Local Guiada	4,298.5	35.7	9.2
CP + SVM (sin GLS)	4,387.6	28.4	7.3
CP + SVM + GLS (Completo)	4,090.7	19.9	13.6

E Evaluación de la Contribución del Componente SVM

VI Análisis y Discusión

A Efectividad del Enfoque Híbrido

Los resultados experimentales demuestran la superioridad del framework híbrido propuesto en múltiples dimensiones de rendimiento:

1. **Calidad de Solución:** La mejora promedio del 13.6% en distancia total sobre los mejores métodos conocidos establece un nuevo benchmark para instancias MDVRP estándar.
2. **Eficiencia Computacional:** La reducción del 66.9% en tiempo computacional demuestra la efectividad de la guía inteligente mediante SVM.
3. **Consistencia:** La desviación estándar reducida (134.2 vs 287.4 del mejor competidor) indica mayor robustez algorítmica.
4. **Escalabilidad:** El rendimiento se mantiene estable incluso para instancias de alta complejidad (P17-P19).

B Análisis de la Contribución del SVM

El estudio de ablación (Tabla 8) revela que:

- El SVM por sí solo contribuye con una mejora del 7.3%
- La combinación SVM + GLS genera sinergia adicional del 6.3%
- La precisión promedio del 94.1% en clasificación de rutas justifica la inversión computacional

C Ventajas del Framework Propuesto

C.1 Flexibilidad Arquitectural

La arquitectura modular permite:

- Adaptación a variantes específicas del MDVRP
- Incorporación de restricciones adicionales sin reestructuración
- Escalamiento horizontal mediante paralelización

C.2 Aprendizaje Adaptativo

El componente SVM proporciona:

- Clasificación en tiempo real de la calidad de soluciones candidatas
- Reducción del espacio de búsqueda mediante filtrado predictivo

- Adaptación continua a características específicas del problema

C.3 Robustez Operacional

El framework demuestra:

- Convergencia estable independientemente del tamaño de instancia
- Resistencia a parámetros sub-óptimos
- Degradación gradual en condiciones de recursos limitados

D Limitaciones Identificadas

D.1 Dependencia de Datos de Entrenamiento

- Requiere conjunto inicial de soluciones de calidad para entrenar SVM
- Rendimiento del SVM depende de la representatividad del dataset
- Necesidad de re-entrenamiento para dominios muy diferentes

D.2 Complejidad Paramétrica

- Mayor número de parámetros comparado con métodos simples
- Sensibilidad a configuración de pesos (α, β, γ)
- Tiempo adicional de configuración inicial

D.3 Escalabilidad Computacional

- Overhead computacional del SVM para instancias muy pequeñas
- Memoria adicional requerida para almacenar modelo entrenado
- Tiempo de entrenamiento inicial no despreciable

VII Conclusiones

A Hallazgos Principales

Esta investigación presenta las siguientes contribuciones significativas al estado del arte en MDVRP:

1. **Framework Híbrido Innovador:** La integración exitosa de CP, optimización y SVM establece un nuevo paradigma para problemas de enrutamiento complejo, demostrando mejoras del 13.6% en calidad de solución.
2. **Eficiencia Computacional Superior:** La reducción del 66.9% en tiempo computacional, manteniendo o mejorando la calidad, representa un avance significativo para aplicaciones industriales.
3. **Robustez Algorítmica:** La consistencia de resultados across 19 instancias diversas, con desviación estándar reducida del 53%, demuestra la confiabilidad del enfoque.

4. **Contribución Metodológica:** El uso de SVM para guiar búsqueda local en problemas de optimización combinatorial abre nuevas líneas de investigación en la intersección de ML y optimización.

B Impacto Práctico

Los resultados obtenidos tienen implicaciones directas para:

- **Logística Industrial:** Reducción de costos operacionales del 13.6% en operaciones multi-depósito
- **Planificación de Rutas:** Tiempo de respuesta reducido permite re-planificación dinámica
- **Sostenibilidad:** Menor distancia total implica reducción en emisiones de CO₂
- **Competitividad:** Ventaja operacional en mercados logísticos competitivos

C Validación Estadística

Los tests estadísticos confirman la significancia de las mejoras:

- **Test t-pareado:** $p < 0.001$ para mejoras en distancia
- **Test Wilcoxon:** $p < 0.005$ para mejoras en tiempo
- **Intervalo de confianza:** 95% para todas las métricas principales

VIII Trabajo Futuro

A Extensiones Inmediatas

A.1 Variantes del MDVRP

- MDVRP con ventanas de tiempo (MDVRPTW)
- MDVRP con flota heterogénea
- MDVRP con recogida y entrega simultánea
- MDVRP dinámico con actualizaciones en tiempo real

A.2 Mejoras Algorítmicas

- Implementación de ensemble de clasificadores (Random Forest, XGBoost)
- Optimización automática de hiperparámetros mediante Bayesian Optimization
- Paralelización del framework para clusters de cómputo
- Implementación de técnicas de transfer learning entre instancias

IX Disponibilidad de Datos

Los datasets utilizados en este estudio están disponibles en repositorios públicos. El código fuente del framework implementado estará disponible en Kaggle: <https://www.kaggle.com/datasets/adamjoseph7945/vehicle-routing-problem-set?resource=download>

References

- [1] J.-F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo, "Vehicle routing," *Handbooks in Operations Research and Management Science*, vol. 14, pp. 367–428, 2007. DOI: 10.1016/S0927-0507(06)14006-2 PDF: ResearchGate
- [2] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed. Philadelphia, PA: SIAM, 2014. DOI: 10.1137/1.9781611973594 PDF: SIAM Library
- [3] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, no. 2, pp. 105–119, 1997. DOI: 10.1002/(SICI)1097-0037(199709)30:2;105::AID-NET5;3.0.CO;2-G PDF: ResearchGate
- [4] B. Crevier, J.-F. Cordeau, and G. Laporte, "The multi-depot vehicle routing problem with inter-depot routes," *European Journal of Operational Research*, vol. 176, no. 2, pp. 756–773, 2007. DOI: 10.1016/j.ejor.2005.08.015 PDF: ResearchGate
- [5] W. Ho, G. T. Ho, P. Ji, and H. C. Lau, "A hybrid genetic algorithm for the multi-depot vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 4, pp. 548–557, 2008. DOI: 10.1016/j.engappai.2007.06.001 PDF: ResearchGate
- [6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. DOI: 10.1007/BF00994018 PDF: Springer
- [7] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY: Springer, 2000. DOI: 10.1007/978-1-4757-3264-1 PDF: Springer
- [8] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, London, UK, 1998, pp. 417–431. DOI: 10.1007/3-540-49481-2_30 PDF: Springer
- [9] P. Laborie, J. Rogerie, P. Shaw, and P. Vilím, "IBM ILOG CP optimizer for scheduling," *Constraints*, vol. 23, no. 2, pp. 210–250, 2018. DOI: 10.1007/s10601-018-9281-x PDF: Springer
- [10] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987. DOI: 10.1287/opre.35.2.254 PDF: INFORMS

- [11] J. Bi, Y. Ma, J. Wang, Z. Zheng, B. Feng, and Y. Zhang, "Learning to Handle Complex Constraints for Vehicle Routing Problems," arXiv preprint arXiv:2410.21066, 2024. PDF: arxiv.org/pdf/2410.21066.pdf
- [12] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, "Heuristics for Vehicle Routing Problem: A Survey and Recent Advances," arXiv:2303.04147, 2023. PDF: arxiv.org/pdf/2303.04147.pdf
- [13] J. Oyola, H. Arntzen, and D. L. Woodruff, "Recent dynamic vehicle routing problems: A survey," *Computers & Industrial Engineering*, vol. 160, p. 107574, 2021. PDF: ResearchGate PDF
- [14] A. Montoya, C. Gu  ret, J. E. Mendoza, and J. G. Villegas, "The Evolution of the Vehicle Routing Problem—A Survey of VRP Research and Practice from 2005 to 2022," PDF: ResearchGate PDF
- [15] Y. Sun, M. Ernst, and X. Li, "A new constraint programming model and a linear programming-based adaptive large neighborhood search for the vehicle routing problem with synchronization constraints," arXiv:1910.13513, 2019. PDF: arxiv.org/pdf/1910.13513.pdf
- [16] F. Marinelli and A. Nucara, "Constraint Programming Algorithms for Route Planning Exploiting Geometrical Information," arXiv:2009.10253, 2020. PDF: arxiv.org/pdf/2009.10253.pdf
- [17] G. V. Monreale and F. Rossi, "Soft Constraint Logic Programming for Electric Vehicle Travel Optimization," arXiv:1212.2056, 2012. PDF: arxiv.org/pdf/1212.2056.pdf
- [18] K. Bulbul and H.   en, "Multi Objective Vehicle Routing Problem: A Survey," 2018. PDF: ResearchGate PDF
- [19] W. Min, "Rich Vehicle Routing Problems and Applications," PhD thesis, Technical University of Denmark, 2013. PDF: DTU Orbit PDF
- [20] L.-M. Rousseau, M. Gendreau, and G. Pesant, "Solving VRPTWs with constraint programming based column generation," *Annals of Operations Research*, vol. 130, no. 1-4, pp. 199–216, 2004. DOI: 10.1023/B:ANOR.0000032576.73681.29
- [21] N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, "The vehicle routing problem," in *Combinatorial Optimization*, Chichester: Wiley, 1979, pp. 315–338.
- [22] M. Nashat Fors et al., "Vehicle routing problems over time: a survey," 2020. PDF: ResearchGate PDF