Ciencia y Educación
(L-ISSN: 2790-8402 E-ISSN: 2707-3378)
Vol. 7 No. 6
Junio del 2026

DESARROLLO DE UN ALGORITMO HÍBRIDO PS-ABC PARA LA RESOLUCIÓN DEL PROBLEMA DE LA MOCHILA

DEVELOPMENT OF A HYBRID PS-ABC ALGORITHM FOR THE KNAPSACK PROBLEM

Autores: [1]Melvyn Alexandro Puruncajas Orozco, [2]José Andrés Zúñiga Cazorla.
[1]ORCID ID: https://orcid.org/ 0009-0007-7571-0786
[2]ORCID ID: https://orcid.org/0009-0006-5350-2866
[1]E-mail de contacto: melvyn.puruncajas@unach.edu.ec
[2]E-mail de contacto: andres.zuniga@unach.edu.ec
Afiliación: [1]* [2]*Universidad Nacional de Chimborazo, (Ecuador).
Artículo recibido: 30 de Mayo del 2026
Artículo revisado: 2 de Junio del 2026
Artículo aprobado: 2 de Junio del 2026
[1]Ingeniero Industrial, graduado de la Universidad Nacional de Chimborazo, (Ecuador), con 2 años de experiencia laboral en Sistemas de Gestión industrial. Maestrante en Matemática Aplicada con mención en Matemática Computacional de la Universidad Nacional de Chimborazo, (Ecuador).
[2]Ingeniero Mecánico, graduado de la Universidad San Francisco de Quito (Ecuador), con 3 años de experiencia laboral como docente de grado y posgrado. Máster en Inteligencia Artificial, graduado de la Universidad Internacional Valenciana, (España).

Resumen

El Problema de la Mochila es un problema de optimización combinatoria caracterizado por ser complejo en términos computacionales. El objetivo del presente estudio fue el desarrollo de un algoritmo híbrido que combine las ventajas de los algoritmos Colonia Artificial de Abejas y Optimización de Enjambre de Partículas compensando sus limitaciones individuales. El algoritmo Colonia Artificial de Abejas es de lenta convergencia por su baja capacidad de explotación, en contraste, el algoritmo Optimización de Enjambre de Partículas tiende a una prematura convergencia en óptimos locales por su baja exploración. La propuesta presenta ambos enfoques en un algoritmo híbrido entre Optimización de Enjambre de Partículas-Colonia Artificial de Abejas que combina la capacidad de exploración del algoritmo Colonia Artificial de Abejas con la explotación de la Optimización de Enjambre de Partículas buscando equilibrar la diversidad y consistencia en la estimación de soluciones. La eficacia del algoritmo híbrido se evaluó aplicando a instancias de prueba estándar. Los resultados obtenidos demuestran que el algoritmo híbrido alcanza un mejor desempeño en términos de mejor valor y velocidad de procesamiento en comparación con las versiones base de los algoritmos Optimización de Enjambre de Partículas y Colonia Artificial de Abejas, logrando valores más cercanos al óptimo. El algoritmo híbrido propuesto se presenta como una alternativa prometedora para aproximar problemas combinatorios complejos, entregando un balance entre exploración y explotación.

Palabras clave: Problema de la mochila, Colonia artificial de abejas, Optimización de enjambre de partículas, Metaheurística, Algoritmo híbrido, Optimización Combinatoria, PS-ABC.

Abstract

The Knapsack Problem is a combinatorial optimization problem characterized by its computational complexity. The aim of this study was to develop a hybrid algorithm that merges the advantages of the Artificial Bee Colony and Particle Swarm Optimization algorithms, offsetting their individual limitations. The Artificial Bee Colony algorithm struggles with slow convergence due to its weak exploitation capabilities, on the other hand, the Particle Swarm Optimization algorithm tends to converge prematurely at local optima because of its low exploration. The proposal introduces both approaches into a hybrid algorithm combining Particle Swarm Optimization-Artificial Bee Colony, which couples the exploration capability of the Artificial Bee Colony algorithm with the exploitation of the Particle Swarm Optimization

seeking a balance between diversity and consistency when estimating solutions. The efficacy of this hybrid algorithm was evaluated using standard benchmark instances. The obtained results demonstrate that the hybrid algorithm outperforms in terms of solution quality and efficiency when compared to the baseline versions of Particle Swarm Optimization and Artificial Bee Colony algorithms, achieving closer values to the global optimum and lower variability across executions. Therefore, the proposed hybrid algorithm stands as a viable alternative for approaching complex combinatorial problems, delivering a trade-off between exploration and exploitation.

Keywords: Knapsack problem, Artificial bee colony, Particle swarm optimization, Metaheuristic, Hybrid algorithm, Combinatorial optimization, PS-ABC.

Sumário

O Problema da Mochila é um problema de otimização combinatória caracterizado por ser complexo em termos computacionais. O objetivo do presente estudo foi o desenvolvimento de um algoritmo híbrido que combina as vantagens dos algoritmos Colônia Artificial de Abelhas e Otimização por Enxame de Partículas, compensando suas limitações individuais. O algoritmo Colônia Artificial de Abelhas apresenta lenta convergência devido à sua baixa capacidade de explotação. Em contraste, o algoritmo de Otimização por Enxame de Partículas tende a uma convergência prematura em ótimos locais por sua baixa capacidade de exploração. A proposta apresenta ambos os enfoques em um algoritmo híbrido entre Otimização por Enxame de Partículas–Colônia Artificial de Abelhas, que combina a capacidade de exploração do algoritmo Colônia Artificial de Abelhas com a explotação da Otimização por Enxame de Partículas, buscando equilibrar a diversidade e a consistência na estimativa de soluções. A eficácia do algoritmo híbrido foi avaliada aplicando instâncias de teste padrão.
Os resultados obtidos demonstram que o algoritmo híbrido alcança melhor desempenho em termos de qualidade e eficiência em comparação com as versões de base dos algoritmos Otimização por Enxame de Partículas e Colônia Artificial de Abelhas, obtendo valores mais próximos do ótimo e com menor variabilidade entre execuções. Em conclusão, o algoritmo híbrido proposto apresenta-se como uma alternativa promissora para abordar problemas combinatórios complexos, entregando um equilíbrio entre explotação e exploração.

Palavras-chave: Problema da mochila, Colônia de abelhas artificiais, Otimização por enxame de partículas, Metaheurística, Algoritmo híbrido, Otimização combinatória, PS-ABC.

Introducción

El Problema de la Mochila o *Knapsack Problem* (*KP*) es un planteamiento matemático de optimización combinatoria de subconjuntos. Consiste en que dado un contenedor (mochila) con su capacidad máxima para almacenar objetos y teniendo una lista de elementos se debe preparar un subconjunto de objetos, seleccionados de manera óptima, para ser colocados en dicha mochila o rechazados. Esto se traduce como una decisión binaria 0-1 donde cada objeto tiene asignado un valor y un peso (enteros positivos). La premisa consiste en que el subconjunto no exceda la restricción de peso permitido por la mochila, buscando el mayor beneficio posible. Este campo de estudio se mantiene vigente, impulsado por la aparición de variantes cada vez más complejas, de alto impacto en la toma de decisiones y en aplicaciones del mundo real. (Galli et al., 2025).

Al ser el 0-1 *Knapsack Problem* un modelo de optimización combinatoria, típicamente es categorizado como una instancia *NP-Hard*. Esto implica que, debido a su complejidad intrínseca, demanda una mayor cantidad de recursos computacionales para alcanzar la

mejor solución cuando el tamaño de instancia crece en el orden $\boldsymbol{O(2^n)}$. Es así que se consideran los enfoques exactos y heurísticos para abordar el 0-1 *KP*; sin embargo, los métodos exactos resultan inadecuados en instancias grandes (Chauhan et al., 2021). El enfoque heurístico se apoya en estrategias de estimación, enfatizando la búsqueda en el espacio de soluciones aproximadas, permitiendo encontrar resultados cercanos al óptimo. En contraste, los algoritmos metaheurísticos constituyen marcos de búsqueda más generales y suelen encontrar aproximaciones de mayor calidad en una amplia variedad de problemas con alta complejidad.

La adopción de estas técnicas ha mostrado un mejor desempeño en contraste con la aplicación de los métodos exactos debido a su robustez y simplicidad. En esta categoría se observan algoritmos metaheurísticos clásicos como algoritmo genético (GA), Optimización de Enjambre de Partículas (PSO), Optimización de Colonia de Hormigas (ACO), Evolución Diferencial (DE), entre otros. Mientras que la Optimización de Manada de Elefantes (EHO), Algoritmo de Enjambre de Krill (KH) y Optimización de Mariposa Monarca (MBO) son algoritmos metaheurísticos bio-inspirados que han sido aplicados en años recientes para la resolución de diversos problemas de optimización complejos (Feng et al., 2026).

Adoptar los mejores elementos de dos o más algoritmos para generar una arquitectura híbrida ha demostrado mejoras significativas en términos de rendimiento del algoritmo y manejo de problemas complejos, con restricciones y sistemas de alta dimensionalidad. Esta categoría de algoritmos se denomina comúnmente como Algoritmos Híbridos Metaheurísticos (MHAs) complementándose para evitar convergencia en óptimos locales mediante reforzar la explotación como la del algoritmo PSO y la capacidad de exploración demostrada en el algoritmo *Artificial Bee Colony* (Nassef et al., 2024). El algoritmo PSO, en su versión binaria (BPSO) propuesta por Kennedy & Eberhart (1997), sirve como referente base para la resolución del 0-1 *KP*, al adaptar el concepto de trayectoria y velocidad a un espacio. Bajo este enfoque, cada partícula no representa una solución definitiva, sino un vector de probabilidades que estima para cada coordenada adoptar un valor binario en cada iteración. Esta arquitectura permite la explotación rápida hacia óptimos locales, gracias a la función sigmoide que regula la actualización de los bits. La robustez demostrada mediante un *Testbed* clásico confirmando que el PSO binario es capaz de converger rápidamente hacia óptimos locales en problemas discretos, lo que lo convierte en una base idónea para procesos de hibridación con otros metaheurísticos (Kennedy y Eberhart, 1997).

El *Artificial Bee Colony* (ABC) es un algoritmo bio-inspirado que imita la dinámica de las abejas. Ha demostrado gran eficacia en la optimización de problemas continuos. Este mismo enfoque se extendió al dominio de problemas discretos, permitiendo abordar directamente el 0-1 *KP* mediante operadores que tratan a las variables binarias. En este modelo, cada fuente de alimento es representada como un vector binario y sobre este actúan las abejas realizando una función análoga a los roles que cumplen las en una colmena de la naturaleza como: abejas empleadas, observadoras y exploradoras. Los resultados experimentales evidenciaron que el ABC binario (BABC) posee precisión y una capacidad de exploración amplia por encima de algoritmos como el PSO y el GA en múltiples instancias del 0-1 *KP*, consolidándose como un

referente para problemas combinatorios, convirtiendo al ABC binario en un candidato idóneo para arquitecturas de hibridación, aportando exploración (Liu y Chen, 2012). La investigación en metaheurística ha mostrado que no existe un algoritmo capaz de superar consistentemente a los demás y se fundamenta en el teorema de *No Free Lunch*, el cual establece que, en toda posible métrica, ningún método de búsqueda, optimización o algoritmo de aprendizaje supervisado es mejor que otro cuando su rendimiento se promedia en todas las posibles instancias de un problema. En otras palabras, un algoritmo tendrá ventaja en cierto grupo de instancias, pero promedia su rendimiento con desventajas en otras. Este hecho brinda la oportunidad para diseñar enfoques híbridos con las mejores características como la exploración y explotación, logrando resultados que superen los métodos metaheurísticos base (Lehre y Lin, 2024).

Los trabajos referenciales de Kennedy y Eberhart (1997) y Liu y Chen (2012) consolidaron las bases teóricas y metodológicas, sin embargo, este campo de estudio aun no alcanza su madurez. En este contexto, la investigación de algoritmos híbridos que generen sinergia de los mejores enfoques se presenta como una alternativa prometedora para ampliar el espectro de problemas binarios y mejorar la calidad de las soluciones (Becerra et al., 2022). Con este enfoque, el presente trabajo tiene por objetivo diseñar un algoritmo híbrido denominado PS-ABC que combina las fortalezas del BPSO y del BABC para ser más eficiente en abordar el 0-1 *KP*, demostrando que la hibridación puede mejorar la calidad de las soluciones encontradas como la velocidad de procesamiento frente a los algoritmos predecesores.

Materiales y Métodos

El estudio se llevó a cabo con enfoque cuantitativo dirigido a recopilar y comparar los datos de rendimiento del algoritmo diseñado PS-ABC frente a sus precursores BPSO y BABC siendo las variables de análisis: el mejor valor obtenido (óptimo) y también la velocidad de procesamiento. Esto con un diseño cuasi experimental dado el control parcial que se realizó a las variables que intervienen (Arana y Guerrero, 2025). La investigación fue de tipo aplicada, ya que se desarrolló el algoritmo PS-ABC como una propuesta que para superar el rendimiento de las arquitecturas en las que se basa, aportando así utilidad inmediata en abordar el problema 0-1 *KP*. Para evaluar esta afirmación, el análisis de datos se realizó con el software SPSS, reconocido por su capacidad para ejecutar análisis estadístico exhaustivo en estudios comparativos promoviendo la validez de resultados, mediante pruebas de normalidad de Shapiro-Wilk, pruebas paramétricas mediante ANOVA y/o pruebas no paramétricas de Kruskal-Wallis (Sánchez et al., 2024).

Tomando en cuenta las recomendaciones de Abdel-Basset et al. (2021), con relación a declarar el entorno tecnológico; los experimentos fueron ejecutados en una laptop personal con procesador AMD Ryzen 7 7730U, 16 GB de memoria RAM a una frecuencia base de 2.0 GHz y sistema operativo Windows 10. En este ambiente computacional se corrieron los algoritmos, implementados en Python 3.11, empleando librerías estándar garantizando reproducibilidad y estabilidad en cada ejecución. Para la comparación entre los algoritmos evaluados, se definió la siguiente configuración base: tamaño de población con 60 partículas, iteraciones máximas en 100 y 50 ejecuciones independientes con semilla aleatoria, evitando así sesgos por afinación inicial y dejando en evidencia las dinámicas

propias de cada método. Los parámetros propios de cada algoritmo como coeficientes de inercia-aceleración en el BPSO, límite de intentos en BABC, radio de búsqueda y número de abejas exploradoras en PSABC, se mantuvieron constantes a lo largo de los experimentos (Dhruv y Dubey, 2023).

Las instancias de prueba empleadas fueron las descritas por Liu et al. (2012), quienes presentan cuatro instancias de referencia: KP20, con 20 ítems y capacidad de mochila igual a 878; KP50, con 50 ítems y capacidad de 1000; KP80, con 80 ítems y capacidad de 1173; y KP100, con 100 ítems y capacidad de 3820. Las instancias cumplen con la misma estructura básica de dos columnas: en la primera fila se define el número de ítems y la capacidad de la mochila y a partir de la segunda fila se presentan el valor y peso de cada ítem (ver figura 1).

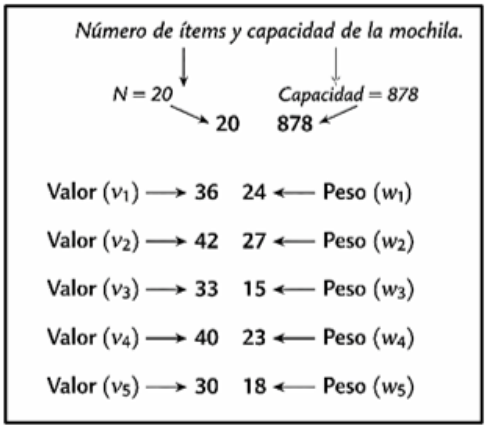


Figura 1. *Ejemplo de estructura de las instancias de prueba.*

Fuente: Elaboración propia

Para este estudio se replicaron los algoritmos *Particle Swarm Optimization*, de Kennedy & Eberhart (1997) en su versión binaria, y *Binary Artificial Bee Colony*, propuesto por Liu & Chen (2012), constituyendo, ambos, la base metodológica sobre la cual se desarrolló la propuesta híbrida y también para la evaluación comparativa posterior. La figura 2 describe la arquitectura del BPSO.

bPSO *(termination criterio: {IterNo, ε_c, ...}, V_{max})*

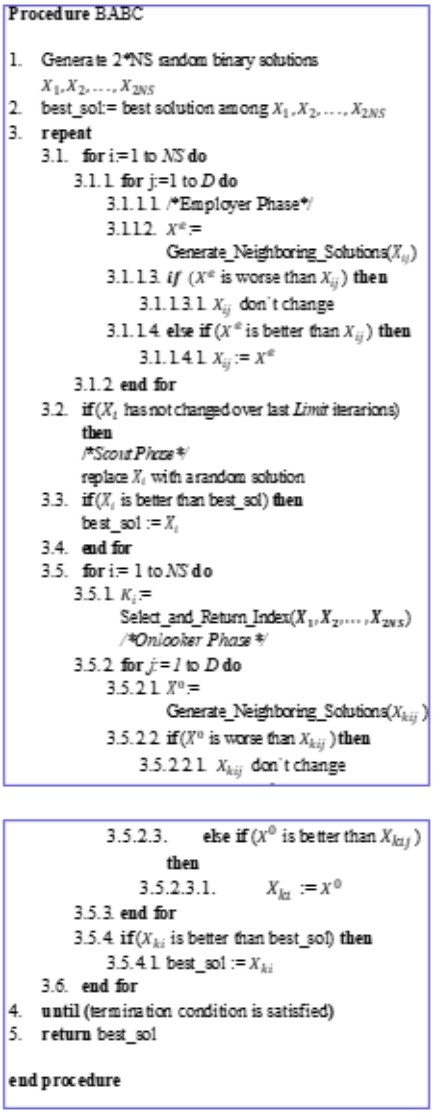
1. For ∀ a ∈ [1, *S*] do:
   1.1. Randomize x_a (1), V_a (1)
   1.2. Let y_a (0) = x_a (1)
   1.3. Let $\hat{y}$ (0) = x_a (1)
2. End For
3. For ∀ t ∈ [1, *IterNo*] do:
   3.1. For ∀ a ∈ [1, *S*] do:
      3.1.1. Compute y_a (*t*) using (1)
      3.1.2. If $(f(y_a(t)) < \min_{1 \le i < a} (f(\hat{y}(t-1)), f(y_i(t)))$ then *gbest* = *a* and $\hat{y}$ (t) = y_a (*t*)
   3.2. End For
   3.3. If any *termination criterion* is met, then **Stop**
   3.4. For ∀ a ∈ [1, *S*] do:
      3.4.1. For ∀ *j* ∈ [1, *N*] do:
         3.4.1.1. Compute $v_{a,j}(t+1)$
         3.4.1.2. If $(|v_{a,j}(t+1)| > v_{max})$ then clamp it to $|v_{a,j}(t+1)| = v_{max}$
         3.4.1.3. Compute $x_{a,j}$ (*t+1*)
      3.4.2. End For
   3.5. End For
4. End For

Figura 2*. Pseudocódigo para el algoritmo BPSO.*

Fuente: Kennedy y Eberhart (1997)

Mientras que la figura 3 describe el algoritmo BABC, proporcionando en ambos casos el proceso de funcionamiento y una visión de sus dinámicas individuales. Para el desarrollo del algoritmo metaheurístico híbrido denominado **PS-ABC**, se integraron las principales características de los algoritmos **Binary Particle Swarm Optimization (BPSO)** y **Binary Artificial Bee Colony (BABC)**, con el propósito de aprovechar simultáneamente las capacidades de exploración global y explotación local en la resolución del problema de la mochila binaria 0-1 (0-1 Knapsack Problem). Del algoritmo BPSO se incorporó el mecanismo de actualización de la velocidad de las partículas, el cual permite orientar la búsqueda hacia regiones prometedoras del

espacio de soluciones mediante la combinación de la experiencia individual y colectiva de la población.



Procedure BABC

1. Generate 2*NS random binary solutions $X_1, X_2, \ldots, X_{2NS}$
2. best_sol:= best solution among $X_1, X_2, \ldots, X_{2NS}$
3. **repeat**
   - 3.1. **for** i:=1 to *NS* **do**
     - 3.1.1. **for** j:=1 to *D* **do**
       - 3.1.1.1. /*Employer Phase*/
       - 3.1.1.2. X^e:= Generate_Neighboring_Solutions(X_{ij})
       - 3.1.1.3. *if* (X^e is worse than X_{ij}) **then**
         - 3.1.1.3.1. X_{ij} don't change
       - 3.1.1.4. **else if** (X^e is better than X_{ij}) **then**
         - 3.1.1.4.1. $X_{ij} := X^e$
     - 3.1.2. **end for**
   - 3.2. **if**(X_i has not changed over last *Limit* iterarions) **then**
     /*Scout Phase*/
     replace X_i with a random solution
   - 3.3. **if**(X_i is better than best_sol) **then**
     best_sol := X_i
   - 3.4. **end for**
   - 3.5. **for** i:= 1 to *NS* **do**
     - 3.5.1. K_i:= Select_and_Return_Index($X_1, X_2, \ldots, X_{2NS}$)
       /*Onlooker Phase*/
     - 3.5.2. **for** j:=1 to *D* **do**
       - 3.5.2.1. X^0:= Generate_Neighboring_Solutions(X_{kij})
       - 3.5.2.2. **if**(X^0 is worse than X_{kij}) **then**
         - 3.5.2.2.1. X_{kij} don't change
       - 3.5.2.3. **else if** (X^0 is better than X_{kij}) **then**
         - 3.5.2.3.1. $X_{kij} := X^0$
     - 3.5.3. **end for**
     - 3.5.4. **if**(X_{ki} is better than best_sol) **then**
       - 3.5.4.1. best_sol := X_{ki}
   - 3.6. **end for**
4. **until** (termination condition is satisfied)
5. **return** best_sol

end procedure

Figura 3. *Pseudocódigo para el algoritmo BABC.*

Fuente: Elaboración propia

Esta estrategia se fundamenta en una analogía social donde cada partícula contribuye al conocimiento global a partir de su propia experiencia, la mejor solución encontrada individualmente y la mejor solución identificada por el conjunto de partículas. De esta manera, el movimiento de cada partícula está influenciado por tres componentes fundamentales: la experiencia personal, la memoria local y la memoria global, favoreciendo una búsqueda adaptativa y colaborativa de soluciones de alta calidad. Por otra parte, del algoritmo BABC se incorporaron los mecanismos de exploración y explotación inspirados en el comportamiento de una colonia de abejas. En este enfoque, las abejas exploradoras se encargan de localizar nuevas fuentes de alimento, mientras que las abejas obreras intensifican la búsqueda alrededor de aquellas fuentes que presentan mejores condiciones. Esta dinámica permite equilibrar la exploración de nuevas regiones del espacio de búsqueda con la explotación intensiva de soluciones prometedoras.

En el algoritmo híbrido PS-ABC, después de actualizar la posición de las partículas mediante el vector de velocidad y aplicar una función sigmoide para su binarización, se seleccionan las partículas élite que representan las mejores soluciones encontradas. Estas partículas son tratadas como nuevas fuentes de alimento y se someten a una búsqueda local dentro de un radio definido mediante la distancia de Hamming, generando soluciones vecinas a través de la inversión de bits. Cuando una solución vecina supera la calidad de la solución actual, la posición de la partícula es reemplazada por la mejor alternativa encontrada.

Adicionalmente, las partículas que dejan de producir mejoras durante un número determinado de iteraciones son reemplazadas mediante mecanismos de exploración inspirados en las abejas exploradoras, promoviendo así la diversificación de la búsqueda y evitando el estancamiento en óptimos locales. Como resultado, el algoritmo PS-ABC combina eficientemente la capacidad

de convergencia del BPSO con las estrategias de exploración y explotación del BABC, permitiendo obtener un mejor desempeño en la resolución del problema abordado.

Tabla 1. *Pseudocódigo para el algoritmo PS-ABC*

Etapa	Pseudocódigo del algoritmo PS-ABC
1. Función Fitness	**Fitness(x)**valor ← ∑ xi · vipeso ← ∑ xi · wiSi peso ≤ C entonces retornar valorSi no retornar 0Fin Si
2. Inicialización del enjambre	**InicializarEnjambre(D)**Generar *n* partículas binarias aleatorias de dimensión D.Inicializar velocidades en 0.(partículas, velocidades) ← InicializarEnjambre(D)
3. Inicialización de memorias	mejor_local[i] ← partículas[i]fit_local[i] ← Fitness(partículas[i])mejor_global ← argmax(fit_local)fit_global ← max(fit_local)
4. Bucle principal	Para iter ← 1 hasta max_iter hacer
5. Evaluación de partículas	Para i ← 1 hasta n hacerfit_actual ← Fitness(partículas[i])
6. Actualización de mejor local	Si fit_actual > fit_local[i] entonces fit_local[i] ← fit_actual mejor_local[i] ← partículas[i]Fin Si
7. Actualización de mejor global	Si fit_actual > fit_global entonces fit_global ← fit_actual mejor_global ← partículas[i]Fin Si
8. Actualización de velocidad	$r_1, r_2 \leftarrow U(0,1)$ velocidades[i] ← ϕ_0·velocidades[i] + $\phi_1 \cdot r_1$·(mejor_local[i] − partículas[i]) + $\phi_2 \cdot r_2$·(mejor_global − partículas[i])
9. Binarización de partículas	Para j ← 1 hasta D hacerSi U(0,1) < σ(velocidades[i][j]) entonces partículas[i][j] ← 1Si no partículas[i][j] ← 0Fin SiFin Para
10. Selección de partículas élite	Seleccionar sb partículas élite con mayor fit_local
11. Búsqueda local (fase de abejas obreras)	Para cada índice idx en élite hacermejor_radio ← partículas[idx]fit_radio ← Fitness(mejor_radio)
12. Generación de vecinos	Para k ← 1 hasta 10 hacervecino ← copia de partículas[idx]Invertir r bits aleatorios (distancia de Hamming) fit_vecino ← Fitness(vecino)
13. Selección del mejor vecino	Si fit_vecino > fit_radio entonces fit_radio ← fit_vecino mejor_radio ← vecinoFin Si
14. Actualización de partícula élite	Si fit_radio > Fitness(partículas[idx]) entonces partículas[idx] ← mejor_radioFin Si
15. Fin de iteraciones	Fin Para (élite)Fin Para (iteraciones)
16. Salida del algoritmo	Retornar mejor_global, fit_global

Fuente: Elaboración propia

Resultados y Discusión

Con el propósito de poder verificar y comparar el rendimiento del híbrido PS-ABC propuesto, se presentan los resultados obtenidos en la ejecución sistemática de los algoritmos correspondientes a BPSO, BABC y la propuesta híbrida implementados en Python. Para garantizar la correcta recolección de datos, cada experimento se realizó con 50 ejecuciones individuales. El análisis se llevó a cabo sobre las instancias de referencia KP20, KP50, KP80 y KP100, lo que permitió registrar el comportamiento de cada algoritmo en entornos de distinta complejidad y escala.

Tabla 1. *Comparación estadística de valores obtenidos en los tres algoritmos.*

	BPSO			BABC			PS-ABC		
Instancias	Media	Desv. Están.	Mejor valor	Media	Desv. Están.	Mejor valor	Media	Desv. Están.	Mejor valor
KP20	1040.1	2.452	1042	1028.40	3.505	1037	1041.3	1.753	1042
KP50	3074.62	11.614	3098	2939.34	26.92	2999	3083.2	8.325	3099
KP80	5089.38	45.825	5172	4430.12	109.929	4666	5136.98	30.887	5183
KP100	15035.98	59.722	15150	15083.04	38.04	15141	15100.72	39.272	15164

Fuente: Elaboración propia

La tabla 2 recopila las métricas de valor como: el mejor valor encontrado, la media y la desviación estándar. Mientras que la tabla 3 recopila las métricas de tiempo como: el mejor tiempo logrado, media y desviación estándar. Esta estrategia asegura evidenciar tendencias consistentes y comparables entre algoritmos constituyendo la base para discutir las ventajas del enfoque híbrido frente a sus metaheurísticas predecesoras. La Tabla 3 presenta la comparación de los tiempos de ejecución de los algoritmos BPSO, BABC y PS-ABC para diferentes instancias del problema de la mochila binaria. Los resultados muestran que el algoritmo híbrido PS-ABC obtuvo consistentemente los menores tiempos de ejecución en todas las instancias evaluadas, registrando tiempos promedio de 0,5412 s para KP20, 1,9898 s para KP50, 2,7083 s para KP80 y 3,2686 s para KP100. En contraste, BABC presentó los tiempos más elevados, alcanzando una media de 66,5 s en la instancia KP100, lo que evidencia una menor eficiencia computacional a medida que aumenta el tamaño del problema. Por su parte, BPSO mostró un desempeño intermedio, con tiempos considerablemente menores que BABC, pero superiores a los obtenidos por PS-ABC. Asimismo, las desviaciones estándar observadas en el algoritmo híbrido fueron las más bajas en la mayoría de los casos, indicando una mayor estabilidad y consistencia entre ejecuciones.

Tabla 2. *Comparación estadística de tiempos de ejecución de los tres algoritmos.*

	BPSO			**BABC**			**PS-ABC**		
Instancias	Media	Desv. Están.	Mejor tiempo (s)	Media	Desv. Están.	Mejor tiempo (s)	Media	Desv. Están.	Mejor tiempo (s)
KP20	0.7595	0.01698	0.744	1.5909	0.01435	1.569	0.5412	0.0053	0.535
KP50	4.121	0.11648	4.001	15.7403	0.15046	15.591	1.9898	0.07876	1.908
KP80	6.4898	0.11173	6.372	16.8225	0.11496	16.554	2.7083	0.05654	2.653
KP100	8.1005	0.25706	7.901	66.500	4.2418	37.271	3.2686	0.10958	3.118

Fuente: Elaboración propia.

La tabla 4 presenta los criterios para interpretar los p-value obtenidos durante el desarrollo del análisis estadístico. Las diferencias estadísticamente significativas en el rendimiento de los algoritmos, correspondientes a los valores alcanzados por algoritmo se presentan en la tabla 5, mientras que los tiempos de ejecución se resumen en la tabla 6. Esto, mediante el uso de estadísticos inferenciales respaldados por pruebas de normalidad y verificación de supuestos.

Tabla 3. *Guía de interpretación del p-value.*

Tipo de prueba	H_0	H_1	**Criterio de decisión**
(*de normalidad*) Shapiro-Wilk	Datos provienen de una distribución normal	Datos No provienen de una distribución normal	Si p-value >0.05: Se acepta H0. Si p-value ≤ 0.05 se rechaza H0
(*de inferencia*) ANOVA/Kruskal-Wallis	No existe diferencias significativas entre los algoritmos	Al menos un grupo tiene diferencias significativas	

Fuente: Elaboración propia

El análisis comparativo de los algoritmos BPSO, BABC y el híbrido PS-ABC evidencian el mejor desempeño en términos de calidad de solución y tiempo de procesamiento. Mediante las instancias evaluadas, se confirma que el híbrido alcanzo mejores valores con menor variabilidad, confirmando que la integración de las capacidades de explotación del BPSO y exploración del BABC fueron ideales.

Tabla 4. *Comparación de pruebas de normalidad y de inferencia de valores obtenidos*

	BPSO			BABC			PS-ABC		
Instancias	Prueba de normalidad	Prueba de inferencia	Resultado	Prueba de normalidad	Prueba de inferencia	Resultado	Prueba de normalidad	Prueba de inferencia	Resultado
KP20	0.000	0.000	93.17	0.000	0.000	27.32	0.000	0.000	106.01
KP50	0.95	0.042	2939.34	0.056	0.000	3074.62	0.554	0.042	3083.2
KP80	0.781	0.000	85.35	0.335	0.000	25.50	0.048	0.000	115.65
KP100	0.206	0.000	15035.98	0.105	0.145	15083.04	0.065	0.145	15100.72

Fuente: Elaboración propia.

Este hallazgo coincide con lo señalado por Nassef et al. (2024), quienes destacan a la hibridación como una metodología para superar las limitaciones individuales de los algoritmos.

Tabla 5*. Comparación de pruebas de normalidad y de inferencia de tiempos obtenidos.*

	BPSO			BABC			PS-ABC		
Instancias	Prueba de normalidad	Prueba de inferencia	Resultado	Prueba de normalidad	Prueba de inferencia	Resultado	Prueba de normalidad	Prueba de inferencia	Resultado
KP20	0.000	0.000	75.50	0.000	0.000	125.5	0.000	0.000	25.5
KP50	0.000	0.000	75.50	0.000	0.000	125.5	0.000	0.000	25.5
KP80	0.000	0.000	75.50	0.941	0.000	125.5	0.000	0.000	25.5
KP100	0.000	0.000	75.50	0.000	0.000	125.5	0.000	0.000	25.5

Fuente: Elaboración propia.

Los resultados muestras una reducción estadísticamente significativa del PS-ABC frente a sus algoritmos predecesores, especialmente cuando crece la complejidad de las instancias. Por un lado, BABC presento mayores tiempos de ejecución, BPSO presento una velocidad intermedia, el híbrido logró un balance entre la calidad de resultados y velocidad de procesamiento. Este hecho refuerza el planeamiento de Zheng et al. (2022), quienes sostienen que los algoritmos híbridos pueden mejorar los modelos en términos de reducir el consumo de tiempo computacional y mejorar la eficiencia en los resultados. El análisis inferencial sustentado con pruebas de normalidad y ANOVA/Kruskal-Wallis, confirmó diferencias estadísticamente significativas dejando en evidencia que el algoritmo con mejor rendimiento fue híbrido el PS-ABC, consolidándose como una alternativa para abordar el problema de la mochila (0-1 KP). Sin embargo, se reconoce que las instancias de prueba sobre las cuales se realizó los experimentos son limitadas en número y escala. Y conforme a Lehre y Lin (2024), aunque los resultados son prometedores, la aplicación más generalizada del híbrido PS-ABC dependerá de estudios que fortalezcan la evidencia lograda y consoliden su desempeño en contexto de mayor complejidad.

Conclusiones

El algoritmo híbrido PS-ABC para abordar el problema de la mochila (0-1 KP) demostró una mejora significativa en la obtención de soluciones de mejor calidad frente a los algoritmos predecesores, validando su pertinencia problemas de optimización binaria. La integración de los enfoques de explotación BPSO y exploración BABC permitió alcanzar un equilibrio en la capacidad de búsqueda sin incrementar el costo computacional.La eficiencia del híbrido PS-ABC también mostró la reducción en el tiempo de procesamiento, reforzando su capacidad como herramienta de optimización en escenarios prácticos. En consecuencia, el algoritmo híbrido presenta la oportunidad de abordar el problema de la mochila con otras variantes y escenarios de mayor escala. Su viabilidad a futuro requiere la afinación adecuada de sus parámetros iniciales y se validen los resultados en contextos más diversos.

Referencias Bibliográficas

Abdel, M., Mohamed, R., Sallam, K., Chakrabortty, R., & Ryan, M. (2021). BSMA: A novel metaheuristic algorithm for multi-dimensional knapsack problems: Method and comprehensive analysis. *Computers & Industrial Engineering, 159*. https://doi.org/10.1016/j.cie.2021.107469

Arana, M., & Guerrero, G. (2025). Paradigma, enfoque y método: Trilogía de pertinencia investigativa. *Ciencia y Educación, 6*(12.1), 446–458. https://doi.org/10.5281/zenodo.17940550

Becerra, M., Lemus, J., Cisternas, F., Crawford, B., Soto, R., Astorga, G., et al. (2022). Continuous metaheuristics for binary optimization problems: An updated systematic literature review. *Mathematics, 11*(1), 129. https://doi.org/10.3390/math11010129

Chauhan, P., Pant, M., & Deep, K. (2021). Gompertz PSO variants for knapsack and multi-knapsack problems. *Applied Mathematics: A Journal of Chinese Universities, 36*(4). https://doi.org/10.1007/s11766-021-4583-y

Dhruv, A., & Dubey, A. (2023). Managing software provenance to enhance reproducibility in computational research. *Computing in Science & Engineering, 25*(3), 60–65. https://doi.org/10.1109/MCSE.2023.3314288

Feng, Y., Hu, T., Chen, X., & Wang, G. (2026). Recent advances in knapsack problem: A comprehensive review of models, algorithms, and applications. *Neurocomputing, 666*. https://doi.org/10.1016/j.neucom.2025.132135

Galli, L., Martello, S., & Toth, P. (2025). The quadratic knapsack problem. *European Journal of Operational Research, 326*(1). https://doi.org/10.1016/j.ejor.2024.12.032

Kennedy, J., & Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 5, 4104–4108. https://doi.org/10.1109/ICSMC.1997.637339

Lehre, P., & Lin, S. (2024). No free lunch theorem and black-box complexity analysis for adversarial optimisation. *Advances in Neural Information Processing Systems, 37*, 121570–121597. https://doi.org/10.52202/079017-3864

Liu, W., & Chen, H. (2012). BABC: A binary version of artificial bee colony algorithm for discrete optimization. *International Journal of Advances in Computing Technology, 4*(14), 307–314. https://doi.org/10.4156/IJACT.VOL4.ISSUE14.35

Liu, W., Niu, B., & Chen, H. (2012). Binary artificial bee colony algorithm for solving 0-1 knapsack problem. *International Journal on Advances in Information Sciences and Service Sciences, 4*(22), 464–470. https://doi.org/10.4156/AISS.VOL4.ISSUE22.57

Nassef, A., Abdelkareem, M., Maghrabie, H., & Baroutaji, A. (2024). Hybrid metaheuristic algorithms: A recent comprehensive review with bibliometric analysis. *International Journal of Electrical and Computer Engineering, 14*(6). https://doi.org/10.11591/ijece.v14i6.pp7022-7035

Sánchez, W., Ramos, J., Montoya, W., & García, F. (2024). Herramientas estadísticas avanzadas para el análisis de datos en investigaciones cuantitativas: Una revisión sistemática. *Ciencia y Educación, 5*(12), 71–89. https://doi.org/10.5281/zenodo.14579231

Yaghini, N., & Seyed, M. (2024). An artificial bee colony algorithm for the multidimensional knapsack problem: Using design of experiments for parameter tuning. *International Journal of Applied Metaheuristic Computing, 15*(1). https://doi.org/10.4018/IJAMC.350225

Zheng, Y., Lv, X., Qian, L., & Liu, X. (2022). An optimal BP neural network track prediction method based on a GA–ACO hybrid algorithm. *Journal of Marine Science and Engineering, 10*(10), 1399. https://doi.org/10.3390/jmse10101399

Declaraciones éticas y editoriales del artículo
Contribución de los autores (Taxonomía CRediT) Melvyn Alexandro Puruncajas Orozco: conceptualización de la investigación, diseño metodológico, desarrollo del proceso investigativo, análisis formal de los datos, redacción del borrador original del manuscrito, revisión crítica del contenido científico y supervisión general del estudio. José Andrés Zúñiga Cazorla: curación y organización de los datos, participación en la recolección de información, validación de los resultados obtenidos y elaboración de representaciones gráficas y visualización de los datos.
Declaración de conflicto de intereses Los autores declaran que no existe conflicto de intereses en relación con la investigación presentada, la autoría del manuscrito ni la publicación del presente artículo.
Declaración de financiamiento La presente investigación no recibió financiamiento específico de agencias públicas, comerciales o de organizaciones sin fines de lucro. En caso de existir financiamiento institucional o externo, este deberá ser declarado explícitamente por los autores en esta sección.
Declaración del editor El editor responsable certifica que el proceso editorial del presente artículo se desarrolló conforme a los principios de integridad científica, transparencia y buenas prácticas editoriales. El manuscrito fue sometido a un proceso de evaluación mediante revisión por pares doble ciego, garantizando la confidencialidad de la identidad de los autores y revisores durante todo el proceso de dictamen académico. Asimismo, el editor declara que el artículo cumple con los criterios científicos, metodológicos y éticos establecidos por la revista.
Declaración de los revisores Los revisores externos que participaron en la evaluación del presente manuscrito declaran haber realizado el proceso de revisión de manera objetiva, independiente y confidencial. Asimismo, manifiestan que no mantienen conflictos de interés con los autores ni con la investigación evaluada, y que sus observaciones y recomendaciones se fundamentan exclusivamente en criterios científicos, metodológicos y académicos.
Declaración ética de la investigación Los autores declaran que la investigación se desarrolló respetando los principios éticos de la investigación científica, garantizando la confidencialidad de los datos y el respeto a los participantes del estudio. En los casos en que la investigación involucre seres humanos, los procedimientos deben ajustarse a los principios éticos establecidos en la Declaración de Helsinki y a las normativas institucionales correspondientes.
Declaración sobre el uso de inteligencia artificial Los autores declaran que el uso de herramientas de inteligencia artificial, en caso de haberse utilizado durante el proceso de investigación o redacción del manuscrito, se realizó únicamente como apoyo técnico para mejorar la claridad del lenguaje o el análisis de información, manteniendo siempre la responsabilidad intelectual sobre el contenido del artículo. Las herramientas de inteligencia artificial no fueron utilizadas como autoras del manuscrito ni sustituyen la responsabilidad académica de los investigadores.
Disponibilidad de datos Los datos que respaldan los resultados de esta investigación estarán disponibles previa solicitud razonable al autor de correspondencia, respetando las normas éticas y de confidencialidad establecidas por la investigación.