

NUEVO ALGORITMO DE ESTIMACIÓN DE DISTRIBUCIONES CON DIFERENTES MÉTODOS DE APRENDIZAJE APLICADO A PLANIFICACIÓN DE PROYECTOS

Gaafar Sadeq S. Mahdi^{*1}, Pedro Piñero Pérez^{*}, Roberto García Vacacela^{**}, Julio Madera Quintana^{***}, Salah Hassan Al-subhi^{*}, Iliana Pérez Pupo^{*}

^{*} Universidad de Ciencias Informáticas, La Habana, Cuba

^{**} Universidad Católica Santiago de Guayaquil, Guayaquil, Ecuador

^{***} Universidad de Camagüey, Camagüey, Cuba

ABSTRACT

This paper aims at proposing a Constraint-based Learning Estimation of Distribution Algorithm (EDA) to solve the Multi-Mode Resource Constrained Project Scheduling Problem (MMRCPS). This problem is considered a combinatorial optimization problem and consists of a set of tasks to be carried out in an ordered way, limited availability of resources, and an execution mode for each task. Generally, traditional EDAs handle the constraints of the problem using strategies such as penalty, Pareto dominance, or others, without taking into account these constraints during the learning process. These traditional approaches usually cause the generation of infeasible solutions in each iteration. In this paper, an approach based on the constraints handling during the learning process of the probabilistic model is proposed. To validate the performance of the algorithms, the new approach in two scenarios is compared. First, in the solution of a set of classical optimization problems, where different EDAs are compared in terms of efficiency and efficacy. The second group of experiments is performed on a set of instances of the PSPLIB database. In this case, the proposed algorithm is compared with a traditional EDA and with other results reported in the bibliography. The experimental results demonstrate that the proposal outperforms both the results of traditional EDA and other EDA reported in the literature.

KEYWORDS: Constraints handling, Estimation of Distribution Algorithms, Metaheuristic, Scheduling.

RESUMEN

En este trabajo se tiene como objetivo proponer un algoritmo con estimación de distribuciones (EDA) basado en restricciones para resolver el problema de planificación de proyectos con recursos limitados y tareas que se ejecutan de diferentes modos (MMRCPS). Este problema es considerado como un problema de optimización combinatoria y se compone de un conjunto de tareas a ejecutar de forma organizada, una disponibilidad limitada de recursos y un modo de ejecución para cada tarea. Generalmente, los EDA tradicionales manejan las restricciones del problema empleando estrategias como la penalización, la dominancia Pareto u otras, sin tener en cuenta estas restricciones durante el proceso de aprendizaje. Estos enfoques tradicionales provocan la generación de soluciones no factibles en cada iteración. En este trabajo se propone un enfoque basado en el tratamiento de las restricciones durante el proceso de aprendizaje del modelo probabilístico. Para validar el rendimiento de los algoritmos, los autores comparan el nuevo enfoque en dos escenarios. Primero, en la solución de un conjunto de problemas clásicos de optimización, donde se comparan diferentes algoritmos EDA teniendo en cuenta la eficiencia y la eficacia. El segundo grupo de experimentos se realiza sobre un conjunto de instancias de la base de datos PSPLIB. En este caso, se compara el algoritmo propuesto con un EDA tradicional y con otros resultados reportados en la bibliografía y. Los resultados experimentales reflejan que la propuesta supera tanto los resultados de los EDA tradicionales como otros EDA reportados en la literatura.

PALABRAS CLAVES: Tratamiento de restricciones, algoritmos con estimación de distribuciones, metaheurística, planificación

1. INTRODUCCIÓN

Los algoritmos con estimación de distribuciones (EDA Estimation of Distribution Algorithm) fueron introducidos en los años noventa del siglo pasado por los autores Muehlenbein, Mahnig y Ochoa [24]. En general, los EDA constituyen una familia de algoritmos para resolver disímiles problemas de optimización [10] y surgen como alternativa a las dificultades de los algoritmos genéticos (GA Genetic Algorithm)

¹ gaafarsadeq@yahoo.com

[18][43]. Estas dificultades están asociadas a que los GA por su naturaleza no expresan explícitamente las interdependencias entre las variables del problema en cuestión y no usan suficientemente esta información durante el proceso de búsqueda [13]. La principal característica de los EDA es la identificación de las funciones de distribución probabilística que modelan las relaciones de dependencia entre las variables del problema a resolver y la generación de los nuevos individuos a partir de esa distribución.

En este sentido, los EDA pueden ser clasificados de acuerdo a dos criterios fundamentales: la estrategia para el tratamiento de la dependencia entre variables y los métodos de aprendizaje.

Respecto al tratamiento de las dependencias entre las variables existen tres grupos de EDA. El primer grupo se concentra en la dependencia entre variables tales como: el Algoritmo con Distribución Marginal Univariada (UMDA *Univariate Marginal Distribution Algorithm*) [11][24], el Algoritmo Genético Compacto (cGA *Compact Genetic Algorithm*) [12], y el Aprendizaje Incremental Basado en Poblaciones (PBIL *Population Based Incremental Learning*) [31]. El segundo grupo se centra en los modelos bivariados como: el Algoritmo de Maximización de la Información Mutua para Clasificación (MIMIC *Mutual Information Maximization for Input Clustering*) [16][17] y el Algoritmo con Distribución Marginal Bivariada (BMDA *Bivariate Marginal Distribution Algorithm*) [16][28]. En el tercer grupo se ubican los algoritmos que no imponen restricciones respecto a las dependencias como: el Algoritmo de Optimización Bayesiana (BOA *Bayesian Optimization Algorithm*) [27][3], el Algoritmo Genético Compacto Extendido (ECGA *Extended Compact Genetic Algorithm*) [37], el Algoritmo con Estimación de Distribuciones Basado en el método Max-Min con Escalador de Colinas (MMHCEDA *Max-Min Hill-Climbing Estimation of Distribution Algorithm*) [21], el Algoritmo de Estimación de Distribuciones Basados en Restricciones (CBEDA *Constraint Based Estimation of Distribution Algorithm*) [21], el Algoritmo Evolutivo basado en el Criterio de Información Bayesiana (EBNA_{BIC} *Estimation of Bayesian Networks Algorithm based on Bayesian Information Criterion*) [3][17], y el Algoritmo con Estimación de Redes Gaussianas (EGNA_{BGe} *Estimation of Gaussian Network Algorithm*) [17][22].

De acuerdo al método de aprendizaje, los EDA se clasifican en dos categorías fundamentales: los que se basan en el aprendizaje paramétrico, como el UMDA [11][24], el PBIL [31], el cGA [12], y el Algoritmo con Distribución Factorizada (FDA *Factorized Distribution Algorithm*) [23][33]; y los que combinan el aprendizaje estructural y el paramétrico. Este segundo grupo se subdivide de la siguiente forma: algoritmos de optimización de métricas, algoritmos basados en la detección de dependencias y algoritmos con enfoques híbridos. Ejemplos de algoritmos basados en la familia de optimización de métricas son: BOA [27] [3], ECGA [37] y el EBNA_{BIC} [3][17]. Ejemplos de algoritmos basados en la detección de dependencias son: el Algoritmo Evolutivo basado en Redes Bayesianas (EBNA_{PC} *Estimation of Bayesian Networks Algorithm Based on PC Algorithm*) [17]. Finalmente, ejemplos de algoritmos con enfoques híbridos son: BMDA [16][28], MIMIC [16][17] y el MMHCEDA [21].

Se describen a continuación algunos algoritmos EDA que representan los diferentes enfoques analizados. Varios de ellos serán empleados en la sección de análisis de resultados.

El EDA más simple es el UMDA propuesto en [24], donde en cada iteración, se estima la distribución marginal de cada variable en la solución seleccionada a partir de sus frecuencias relativas. Este algoritmo no gestiona las restricciones de dependencia existente entre las variables en el problema de optimización y centra su aprendizaje en el análisis probabilístico de cada variable de forma independiente.

El MIMIC [17] busca la permutación β que más se aproxima a la distribución de las variables seleccionadas, minimizando la distancia de Kullback-Leiber. Dado que hay $n!$ permutaciones posibles para n variables, los autores sugieren un enfoque ávido que hace uso de la entropía de los datos muestreados. El EBNA [3][17] y sus variantes EBNA_{PC} y EBNA_{BIC} son algunos de los EDA que utilizan el aprendizaje de redes bayesianas, el primero basado en las pruebas de dependencias entre tres variables y el segundo basado en la optimización de métricas. La idea es construir la estructura más adecuada lo más rápido posible (aunque no sea la óptima), a partir de un grafo sin arcos. En cada paso se va añadiendo el arco que mejora el fitness que se esté utilizando hasta llegar a construir la red completa.

El FDA [23] es aplicable a funciones aditivamente descomponibles. Estas funciones se utilizan para simular problemas que pueden descomponerse en subproblemas más pequeños. En este algoritmo se considera que la distribución de probabilidad se factoriza de acuerdo a un modelo de probabilidad simplificado.

Los EDA pueden ser aplicados para resolver problemas de optimización con restricciones. Estos problemas se presentan en varias esferas de la vida cotidiana, como la informática, la construcción, la industria, entre otras, y en muchos de ellos intervienen numerosas variables y factores que los convierten en problemas complejos

del tipo NP- hard [7][40]. Un ejemplo de ello, son los problemas de planificación.

Los problemas de planificación o *Scheduling* son algunos de los problemas de optimización más investigados [3]. Generalmente se componen de un conjunto de tareas, disponibilidad de recursos y algunas restricciones a tener en cuenta durante el proceso de optimización. Estos problemas se consideran aún más complejos si se componen de múltiples sub-problemas interrelacionados entre sí. Tal es el caso de los problemas de planificación de proyectos con recursos limitados y tareas que se ejecutan con múltiples modos (MMRCPSP *Multi-Mode Resource Constrained Project Scheduling Problem*) [9].

Teniendo en cuenta la complejidad de los MMRCPSPs, se han empleado varias metaheurísticas para su solución, entre las que se destacan GA [1][43][44], algoritmos basados en la metaheurística enjambre de partículas (PSO) [34][36], *Simulated Annealing* (SA) [40], algoritmos EDA [3][4][6][38], u otras aproximaciones combinando metaheurísticas con métodos de búsqueda local [2][9][32], siendo los EDA los menos utilizados de toda esta gama de metaheurísticas.

Wang y Fang [41] proponen el HEDA, el cual combina un EDA con búsqueda local. Este algoritmo usa dos modelos probabilísticos para generar una solución al problema de planificación, uno para las tareas y otro para los modos. Para un problema de n tareas y m modos de ejecución, el modelo probabilístico para generar las soluciones será del tamaño $n*n$, mientras que el otro tendrá el tamaño de $n*m$. Se asegura que cada tarea aparezca una sola vez en la lista a través del empleo de un procedimiento llamado *permutation-based Probability Generation Mechanism* (PGM).

Luego Soliman et al. [35] presentan un algoritmo EDA híbrido con técnicas de búsqueda local aleatoria para la solución del problema de planificación del tipo MMRCPSP. La técnica de búsqueda local es basada en los operadores “eliminar” e “insertar” para permitir mejor explotación del espacio de búsqueda. Primeramente, utilizan lista de tareas y modos, luego evalúan todos los individuos por un esquema de generación de tareas llamado *multi-mode serial schedule generation scheme* (MSSGS), y después aplican las técnicas de búsqueda local para evaluar los individuos seleccionados. Sin embargo, el chequeo de cada modo de ejecución de cada tarea en las dos propuestas anteriores de forma iterativa implica un aumento significativo en el costo computacional para evaluar las soluciones obtenidas.

En los trabajos [5] y [6] Mayowa propone el algoritmo RK-EDA, que es un algoritmo EDA diseñado para resolver problemas basados en permutaciones y lo aplican en la solución del problema de asignación de recursos en las tareas de planificación. Este algoritmo se basa al igual que el UMDA en el cálculo de la probabilidad de ocurrencia de los valores en las soluciones prominentes, que es expresado como la media de los valores. Pero a diferencia del UMDA, en este algoritmo los autores fijan un valor de varianza respecto a la media y emplean este valor para controlar el nivel de dispersión o concentración de la búsqueda. Combinan entonces la media calculada con la varianza establecida y generan las nuevas soluciones usando una función de distribución gaussiana.

Otra alternativa fue la combinación del GA con EDA presentado en [4] como el algoritmo PBGA-EDA, donde los nuevos hijos en el algoritmo genético se generan a partir del análisis de las funciones de distribución de las tareas respecto a los modos.

Por último, Tian y colaboradores en [38] proponen el algoritmo Robust hGMEDA como una extensión del hGMEDA [39] y se basa en la hibridación entre EDA y GA, para la solución del MMRCPSP con incertidumbre en el tiempo de duración entre tareas y lo validan con bases de datos de PSPLIB. Los resultados obtenidos por estos algoritmos serán comparados con la propuesta de este trabajo en la sección de análisis de resultados

A partir del análisis se identifican como tendencias en el trabajo con los EDA las siguientes:

1. La hibridación de los EDA con otros métodos de optimización compensando las capacidades de exploración y explotación durante la búsqueda.
2. A pesar de los avances logrados en la solución del MMRCPSP, aún persisten un conjunto de dificultades asociadas al tratamiento eficiente de las restricciones, lo que provoca la generación innecesaria de soluciones no factibles, afectando la eficiencia durante la búsqueda.

Se mantiene el tratamiento de restricciones con métodos establecidos en la bibliografía entre los que destacan estrategias como la penalización y el enfoque Pareto, pero no se identifican estrategias que combinen el tratamiento de restricciones en el modelo probabilístico. Este último elemento constituye una de las motivaciones del presente trabajo.

Existen diferentes escenarios dentro del MMRCPSP caracterizados por la fuerte presencia de restricciones. Un ejemplo de esto es la construcción de cronogramas óptimos o cuasi óptimos para la ejecución de

proyectos. Este es un problema presente en numerosos entornos de la práctica social entre los que se destacan la gestión de proyectos de inversión, el desarrollo de proyectos de tecnologías de información, entre otros. Generalmente estos son escenarios con un elevado impacto económico y social [30], con presencia de restricciones y múltiples modos de ejecución. Constituye una importante motivación para los autores de este trabajo la aplicación de los algoritmos propuestos en la presente investigación en dichos escenarios.

En este trabajo, se propone el Algoritmo con Estimación de Distribución basado en Restricciones (CLEDA *Constraint-based Learning Estimation of Distribution Algorithm*) que está inspirado en el algoritmo FDA [23]. Este algoritmo utiliza el aprendizaje de los modelos probabilísticos que describen las restricciones de dependencia entre las variables e introduce transformaciones a los modelos aprendidos considerando otras restricciones del problema de optimización a resolver.

El trabajo está estructurado en cinco secciones. En la segunda sección se describe el problema de optimización mientras que la propuesta del nuevo algoritmo y el alcance del mismo se detallan en la tercera sección. En la cuarta sección se presentan los resultados empíricos de la comparación del algoritmo propuesto con otros algoritmos, en la resolución de problemas clásicos de optimización y en la resolución del problema de *Scheduling* del tipo MMRCPS. En la quinta sección se presentan las conclusiones del trabajo.

2. PROBLEMA DE OPTIMIZACIÓN

El problema de planificación de proyectos con recursos limitados (RCPS) es uno de los más importantes de su tipo en el contexto de *Scheduling*. Su objetivo es determinar la fecha de inicio de cada tarea de tal forma que se minimice la duración del proyecto y no se violen las restricciones de recursos ni de precedencia entre las tareas [43].

El problema de planificación de proyectos con recursos limitados y tareas que se ejecutan de múltiples modos (MMRCPS) es una extensión del RCPS convencional [9]. En este problema además de determinar el tiempo de inicio de las tareas, hay que seleccionar el modo de procesamiento de cada tarea de tal manera se minimice la duración del proyecto sujeta a las restricciones antes mencionadas. Este problema puede describirse como sigue:

Un proyecto consiste de un conjunto de n tareas. Cada tarea $j = \{1, \dots, n\}$. A partir de $j=2$, cada tarea tiene un conjunto de tareas antecesoras H_j y no puede comenzar hasta que haya terminado cada tarea antecesora h , $h \in H_j$. Además, una tarea j debe ser ejecutada en un modo de procesamiento $m \in \{1, \dots, M_j\}$, donde M_j representa el número de los posibles modos de j . Para ello, existen dos categorías de recursos: renovables $k \in [1, K]$ y no renovables $k^r \in [1, K^r]$, los primeros son disponibles en cada período de tiempo t y pueden ser utilizados más de una vez durante el desarrollo del proyecto, mientras que los segundos, como lo indica su nombre, no pueden ser reutilizados y su disponibilidad en el horizonte T del proyecto es K^r .

El objetivo del MMRCPS es minimizar la duración del proyecto (*Makespan*) mediante la selección exacta del modo de ejecución de cada tarea formulado como sigue:

$$\text{minimizar } Z = \sum_{m \in M} \sum_{t=EF_j}^{LF_j} tx_{jmt} \quad \forall m \in M_j \quad (2.1)$$

Donde EF_j y LF_j representan el tiempo de finalización más temprana y más tardío de la tarea j respectivamente. La variable $x_{jmt} = 1$ si la tarea j es ejecutada con el modo m y en el período de tiempo t , de lo contrario toma el valor 0. Esta ecuación está sujeta al conjunto de restricciones asociadas con la precedencia entre las tareas y la disponibilidad de los recursos tanto renovables como no renovables descritos en la sección 3.

3. MATERIALES Y MÉTODOS

En esta sección se propone el algoritmo CLEDA y se realiza un breve análisis que muestra las diferencias entre este algoritmo y un EDA en su forma clásica, ver Algoritmo 1.

Algoritmo 1. Pseudo-código de un algoritmo EDA en su forma clásica

1. $k \leftarrow 0$. Generar R individuos de forma aleatoria
2. DO {
3. *Evaluar los individuos en la función objetivo $f(x)$*
4. *Seleccionar un conjunto S de N individuos, donde $N \leq R$*
5. $p^S(x, k)$ = *Estimar la distribución de probabilidad a partir de S*
6. *Generar R nuevos individuos a partir de $p(x, k+1) \approx p^S(x, k)$*
7. $k \leftarrow k + 1$
8. } UNTIL *Hasta que se cumpla la condición de parada*

Como características fundamentales del CLEDA se pueden señalar las siguientes:

- Estimación del modelo probabilístico, inspirado en el algoritmo FDA [23], que describe las relaciones de dependencia entre variables de los individuos seleccionados.
- El modelo probabilístico construido es transformado a partir de incorporar o modificar los arcos que expresen las restricciones del problema en cuestión.
- El modelo probabilístico transformado es empleado en la estimación de los parámetros para el paso de generación de los nuevos individuos.

En el Algoritmo 2 se muestra el pseudo-código del algoritmo (CLEDA), propuesto en este trabajo.

Algoritmo 2. CLEDA_{FDA}

1. $k \leftarrow 0$. Generar R soluciones de forma aleatoria
2. DO {
3. *Evaluar los individuos en la función objetivo $f(x)$*
4. *Seleccionar un conjunto S de soluciones donde $|S| \leq R$*
5. *Aprendizaje del modelo probabilístico $p^S(x, k)$ a partir de S*
6. $p_m^S(x, k)$ = *Extensión aprendizaje y transformación $p^S(x, k)$ considerando restricciones*
7. *Generar R nuevos puntos a partir de $p(x, k+1) \approx p_m^S(x, k)$*
8. *Mejora de los individuos en R a partir de análisis de reservas.*
9. $k \leftarrow k + 1$
10. } UNTIL *Hasta que se cumpla el criterio de parada*

Como parte del paso de aprendizaje del modelo probabilístico en la implementación de este algoritmo se proponen emplear el algoritmo de Maximización Restringida (RSMAX2 *Restricted Maximization*) [26], el cual es una implementación más general del MMHC (*Max-Min Hill-Climbing*) [21] que puede utilizar cualquier combinación de estrategias de aprendizaje de redes basadas en puntuaciones con métodos basados en restricciones.

Entre los métodos de aprendizaje basados en puntuaciones el algoritmo RSMAX2 acepta las siguientes alternativas:

- Hill-Climbing (hc): la búsqueda basada en el escalador de colinas aplicada en la construcción del modelo probabilístico [26].
- Tabu Search (tabu): como una modificación del escalador de colinas capaz de escapar de los óptimos locales [26].

Entre los métodos de aprendizaje basados en restricciones el algoritmo RSMAX2 acepta las siguientes alternativas:

- *Grow-Shrink (GS)* [26]
- *Incremental Association Markov Blanket (IAMB)* [19]
- *InterLeaved-IAMB (INTER.IAMB)* [42]
- *Fast Incremental Association Markov Blanket, (FAST.IAMB)* [20]
- *Max-Min Parents and Children (MMPC)* [21]

El algoritmo GS utiliza la información de la manta de Markov para inducir la estructura del modelo probabilístico y se caracteriza por su escalabilidad. A pesar de sus ventajas, el algoritmo GS puede fallar en casos extremos, motivo por el cual surgen los algoritmos IAMB e INTER.IAMB. La diferencia entre el GS y el IAMB es que este último emplea una “heurística dinámica” para ordenar a las variables. El INTER.IAMB por su parte elimina algunos falsos positivos en la fase del cultivo lo que lo hace más confiable, aunque

generalmente es más lento que los anteriores.

Para resolver el mencionado problema surge el FAST.IAMB [20], un algoritmo que reúne la confiabilidad y la eficiencia del INTER.IAMB y la rapidez del GS y el IAMB. Por su parte el algoritmo MMPC [21] permite el trabajo con grafos no dirigidos basado en la detección de vecinos a partir de la maximización y minimización de medidas de asociación entre nodos detectadas en iteraciones previas.

Una vez realizado el proceso de aprendizaje, en el paso 6 el modelo probabilístico aprendido $p^S(x, k)$ es extendido y transformado a partir del análisis de las restricciones y se logra un nuevo modelo probabilístico $p_m^S(x, k)$. Las transformaciones y extensiones al aprendizaje dependerán del problema en cuestión como se explica a continuación.

En los problemas planificación de proyectos del tipo MMRCPSPP están presentes las siguientes restricciones:

- Restricciones en el uso de recursos renovables.
- Restricciones asociadas a recursos no renovables.
- Restricciones asociadas a la precedencia entre tareas con cuatro tipos de relaciones entre las tareas: relaciones inicio-inicio, inicio-fin, fin-inicio y fin-fin [29].

En este trabajo fue implementado el algoritmo CLEDA_{FDA}, para la solución del problema de planificación de proyectos considerando las siguientes alternativas de transformación y aprendizaje del modelo probabilístico a partir de las restricciones:

- La poda de arcos del modelo probabilístico que incumplan las restricciones del problema. Por ejemplo, la probabilidad de asignación a una tarea j , de una fecha de inicio menor que las fechas de fin de las tareas que la anteceden, se pone a cero y ese arco es podado.
- Aprendizaje del modelo probabilístico tomando en cuenta dos alternativas diferentes: el aprendizaje del modelo probabilístico considerando las relaciones de precedencia de tareas o el aprendizaje considerando relaciones entre tareas que pueden ser ejecutadas de forma paralela, que compiten por los recursos.
- Aprendizaje de las mejores combinaciones de tareas-modos de ejecución que maximizan la cantidad de tareas que pueden iniciar el mismo día, aplicable en problemas MMRCPSPP.
- Aprendizaje de estrategias de mejora de las soluciones, que permiten la aplicación de métodos de aprendizaje por refuerzo en la mejora de las soluciones. Este aprendizaje presupone que en el paso 8 se guardó como parte de la solución la estrategia de mejora empleada.

En el paso 7 se generan nuevos individuos como soluciones candidatas que podrán ser mejoras a partir del análisis de las reservas de recursos en el siguiente paso.

Luego en el paso 8 “*Mejora de los individuos en R a partir de análisis de reserva*” se diseña como un paso que puede ser aplicado en problemas donde hay presencia de recursos renovables y/o no renovables, como el problema MMRCPSPP. En el presente trabajo se desarrolla este paso en el problema de planificación de proyectos que se explica en la sección de análisis de resultados. En este paso se analizan las reservas de recursos respecto a la satisfacción de restricciones de las soluciones generadas con la $p_m^S(x, k)$ considerando los siguientes elementos:

- Análisis de reservas diarias en el uso de los recursos renovables, y las posibilidades de aprovechamiento de estas reservas.
- Selección de estrategia para la generación de una nueva combinación tarea-modo para el aprovechamiento de las reservas de recursos renovables, maximizando la cantidad de tareas que pueden iniciar el mismo día. La generación de nuevas combinaciones provocará la obtención de nuevas soluciones que están en la vecindad de la solución anterior con reserva a partir de la cual se generó.
- Entre las estrategias de generación de nuevas combinaciones se pueden emplear:
 - La generación de una nueva combinación de forma estocástica potenciando aquellos modos que minimicen el consumo de recursos.
 - La generación de una nueva combinación de forma estocástica potenciando aquellos modos que maximicen el consumo de recursos.
 - Selección de forma aleatoria de nuevas combinaciones de tareas-modos.
- Aprendizaje de la estrategia de mejora de las soluciones, elemento que podrá ser aprendido en la construcción del modelo probabilístico de la siguiente iteración del algoritmo.

De esta forma el algoritmo CLEDA puede ser adaptado a la resolución de diferentes problemas en

dependencia del tipo de restricciones del problema en concreto. En la siguiente sección se valida el algoritmo propuesto en diferentes escenarios.

3.1 PROBLEMAS DE OPTIMIZACIÓN CLÁSICOS EMPLEADOS PARA LA VALIDACIÓN

Existe un conjunto de problemas clásicos de optimización cuyo análisis es recomendado para el desarrollo de investigaciones. Estos problemas se destacan por las peculiaridades de sus funciones entre las que se encuentran: *FOneMax*, *FPlateau*, *FQuadratic* y *FSixPeaks* que se describen a continuación.

FOneMax: es una función lineal cuyo máximo global es igual a n (cantidad de variables) y se alcanza si todas las variables toman el valor uno.

$$FOneMax(x_1, \dots, x_n) = \sum_{j=1}^n x_j \quad (3.1.1)$$

FPlateau: conocida también como el problema “royal road” de 3bits. Su solución corresponde a un vector n -dimensional, tal que $n = 3 \times k$. Su máximo global es igual a k .

$$FPlateau(x) = \sum_{j=1}^k f(x_{3j-2}, x_{3j-1}, x_{3j}) \quad (3.1.2)$$

donde:

$$f(x_1, x_2, x_3) = \begin{cases} 1 & \text{si } x_1 = 1 \text{ y } x_2 = 1 \text{ y } x_3 = 1 \\ 0 & \text{en otro caso} \end{cases}$$

FQuadratic: la solución de esta función se compone por un vector n -dimensional, tal que $n = 2 \times k$. Su máximo global es igual a k . Se define matemáticamente como:

$$Quadratic(x) = \sum_{i=1}^k f(x_{2i-1}, x_{2i}) \quad (3.1.3)$$

donde:

$$g(s, t) = \begin{cases} 0.9, & \text{si } s = 0 \text{ y } t = 0 \\ 1.0, & \text{si } s = 1 \text{ y } t = 1 \\ 0.0, & \text{en otro caso} \end{cases}$$

FSixPeaks: desarrollada por [17]. Este problema puede definirse como sigue:

$$FSixPeaks(u, v) = \max \left\{ \begin{array}{l} cola(0, u), \\ cabeza(1, u), \\ cola(1, u), \\ cabeza(0, u) \end{array} \right\} + R(u, v) \quad (3.1.4)$$

donde:

$cola(c, u)$ = la cantidad de $c(s)$ seguidos al final de u

$cabeza(c, u)$ = la cantidad de $c(s)$ seguidos al inicio de u

$$R(u, v) = \begin{cases} n & \text{si } cola(0, u) > v, & \text{y } cabeza(1, u) > v \text{ o} \\ & \text{cola}(1, u) > v, & \text{y } cabeza(0, u) > v \\ 0 & \text{en otro caso} \end{cases}$$

Esta función tiene cuatro máximos globales cuyo valor es el mismo, $n+(n-(v+1))$. Ellos se alcanzan si:

- 1- Se forma la secuencia de $(1, 1, \dots, 1, 0, 0, \dots, 0)$ con $v+1$ números 1 al inicio de la cadena y el resto 0 .
- 2- Se forma la secuencia de $(0, 0, \dots, 0, 1, 1, \dots, 1)$, con $v+1$ primeros valores en 0 y el resto 1 .
- 3- Los otros dos máximos globales se alcanzan si hay $v+1$ de 1 al final encabezados por todos 0 o con $v+1$ de 0 encabezados por todos 1 .

Además, esta función tiene dos óptimos locales cuyo valor es igual a n y se alcanzan si todos los bits son 0 s o 1 s. En los experimentos realizados en esta investigación el valor de v es del 30% de n . Optimizar esta función requiere de aumentar el tamaño de población, lo que implica el aumento también de la cantidad de evaluaciones.

En el experimento se utilizaron los mismos parámetros de Larrañaga y colaboradores [17], para luego hacer una comparación entre sus resultados y los de la propuesta de esta investigación en sus diferentes configuraciones.

4. RESULTADOS Y DISCUSIÓN

En esta sección se exponen los resultados de dos grupos de experimentos que comparan las diferentes propuestas CLEDA con otros algoritmos presentados en la introducción de este trabajo. El primer grupo de experimentos se basa en la comparación sobre los problemas clásicos de optimización descritos en la sección anterior, mientras que en el segundo grupo se comparan los resultados en la resolución de problemas de planificación de proyectos del tipo MMRCPS.

4.1 Resultados experimentales en problemas clásicos de optimización

Para disminuir el sesgo en la comparación entre los diferentes algoritmos se establecen los siguientes parámetros, comunes a los algoritmos:

- Método de selección por truncamiento con un tamaño de la población a seleccionar del 30%.
- Poblaciones de igual tamaño para las comparaciones.
- Siempre se realizaron 100 corridas de experimentos para cada algoritmo y en la comparación se promedian los resultados.
- Condición de parada en todos los casos es: encontrar el óptimo o fijar una cantidad máxima de evaluaciones en 100000. Los parámetros específicos en la validación del algoritmo de cada función objetivo se presentan en la Tabla 1.

Tabla 1. Parámetros empleados en pruebas con las funciones objetivo

<i>Función objetivo/ Parámetro</i>	<i>FOneMax</i>	<i>FSixPeaks</i>	<i>FPlateau</i>	<i>FQuadratic</i>
Tamaño de población	512	1600	512	800
Cantidad de variables	128	50	105	128
Óptimo global	128	84	35	64
Óptimo local	-	50	-	-

El análisis de los resultados se enfoca en la comparación respecto a las siguientes variables:

- Eficiencia de los algoritmos de acuerdo a la cantidad de evaluaciones que ejecutan.
- Eficacia de los algoritmos respecto al porcentaje de veces en que encuentran el óptimo y a la media de los máximos valores encontrados.

Se presentan a continuación los resultados de comparación de diferentes algoritmos.

4.1.1 Resultados respecto a la función FOneMax

En esta subsección se presentan los resultados de la comparación de las diferentes configuraciones del algoritmo CLEDA con otros algoritmos en la función FOneMax (ver Tabla 2).

En la tabla 2 se muestra que el algoritmo CLEDA en cada una de sus configuraciones alcanzó el óptimo. Se compara CLEDA en sus diferentes configuraciones con los algoritmos MIMIC [17], GA [17] y EBNA [17] (en sus dos variantes). Se demuestra que en cuanto a eficacia ningún algoritmo es dominante sobre los otros, excepto GA que obtuvo resultados más discretos.

Respecto a la eficiencia el algoritmo EBNA_{BIC} fue dominante, aunque se señala que la combinación de CLEDA_{HC+IAMB} también obtuvo resultados satisfactorios.

Tabla 2. Comparación de los resultados obtenidos con CLEDA sobre la función FOnemax

<i>Algoritmo</i>	<i>% de veces que encontró el óptimo</i>	<i>Media de la cantidad de evaluaciones</i>	<i>Media de los máximos alcanzados</i>
CLEDA _{HC+MMPC}	100	6476.8±300.6	128
CLEDA _{FDA}	100	6067.2±187.5	128
CLEDA _{HC+IAMB}	100	5964.8±250.5	128
CLEDA _{HC+FAST.IAMB}	100	6067.2±381.5	128
CLEDA _{HC+INTER.IAMB}	100	6067.2±187.6	128
CLEDA _{TABU+MMPC}	100	6553.6±267.8	128
CLEDA _{TABU+GS}	100	6016.0±366.7	128
CLEDA _{TABU+IAMB}	100	6092.8±157.6	128
CLEDA _{TABU+FAST.IAMB}	100	6195.2±282.8	128
CLEDA _{TABU+INTER.IAMB}	100	6092.8±228.9	128
UMDA	100	7356	128
MIMIC	100	7382	128
EBNA _{PC}	100	7421	128
EBNA _{BIC}	100	1369	128
GA	11	36214	125.75

4.1.2 Resultados respecto a la función FSixPeaks

En la Tabla 3 se muestra que los algoritmos CLEDA_{HC+INTER.IAMB} y CLEDA_{FDA} alcanzaron los mejores resultados en comparación con el resto de los algoritmos, considerando un balance entre la eficiencia y la eficacia. Con respecto a la eficacia, el GA llegó a alcanzar el mayor valor promedio de los máximos con 83.11 aunque alcanzó solo 70 veces el óptimo global, siendo sus resultados menores que el CLEDA_{HC+INTER.IAMB}. Respecto a la eficiencia, el GA requirió 1.5 veces más evaluaciones que el CLEDA. Por lo anteriormente explicado el GA no es dominante respecto al CLEDA.

Por otra parte, el algoritmo MIMIC fue el más eficiente con 20269 evaluaciones, pero solo alcanzó el 57.86 como promedio de valores máximos y encontró el óptimo en 17 ocasiones, lo cual lo ubica entre los peores respecto a esta variable.

Tabla 3. Comparación de los resultados obtenidos con CLEDA con otros algoritmos en la función FSixPeaks

<i>Algoritmo</i>	<i>% de veces que encontró el óptimo</i>	<i>Media de la cantidad de evaluaciones</i>	<i>Media de los máximos alcanzados</i>
CLEDA _{FDA}	65	31066.66±5126.0	72.1
CLEDA _{HC+IAMB}	50	32000.00±3109.8	68.55
CLEDA _{HC+FAST.IAMB}	45	35022.22±3040.4	65.3
CLEDA _{HC+INTER.IAMB}	75	32213.30±3721.3	75.5

UMDA	18	63332	62.10
MIMIC	17	20269	57.86
EBNA _{PC}	0	47655	51.55
EBNA _{BIC}	62	43444	71.9
GA	70	49884	83.11

4.1.3 Resultados respecto a la función FPlateau

Los resultados de comparación de diferentes combinaciones del algoritmo CLEDA respecto a la función FPlateau se muestran la Tabla 4. En este caso, no se encontraron en la bibliografía consultada resultados de otros algoritmos respecto a esta función.

Es notorio en la Tabla 4 que todas las combinaciones de CLEDA lograron encontrar el óptimo en el 100% de las veces, por tanto, en cuanto a la variable eficacia no hay un algoritmo dominante sobre otro. Sin embargo, respecto a la eficiencia se muestra que el mejor es el CLEDA_{FDA} porque requirió menos evaluaciones para alcanzar el óptimo.

Tabla 4. Comparación de los resultados obtenidos con CLEDA con diferentes algoritmos en la función FPlateau

<i>Algoritmo</i>	<i>% de veces que encontró el óptimo</i>	<i>Media de la cantidad de evaluaciones</i>	<i>Media de los máximos alcanzados</i>
CLEDA _{HC+MMPC}	100	6988.8±448.1	35
CLEDA _{FDA}	100	6246.4±267.9	35
CLEDA _{HC+IAMB}	100	6553.6±267.9	35
CLEDA _{HC+FAST.IAMB}	100	6579.2±381.5	35
CLEDA _{HC+INTER.IAMB}	100	6451.2±306.3	35
CLEDA _{TABU+MMPC}	100	6758.4±457.9	35
CLEDA _{TABU+GS}	100	6400.0±262.7	35
CLEDA _{TABU+IAMB}	100	6425.6±388.7	35
CLEDA _{TABU+FAST.IAMB}	100	6476.8±505.9	35
CLEDA _{TABU+INTER.IAMB}	100	6656.0±287.7	35

4.1.4 Resultados respecto a la función FQuadratic

Se presentan los resultados de comparación de diferentes combinaciones del algoritmo CLEDA respecto a la función FQuadratic (ver Tabla 5).

La Tabla 5 muestra que la mayoría de las combinaciones de CLEDA lograron encontrar el óptimo, pero solo cuatro de las combinaciones lo encontraron el 100% de las veces. Las configuraciones con mejores resultados fueron CLEDA_{TABU+GS}, CLEDA_{HC+MMPC}, CLEDA_{FDA} y CLEDA_{TABU+MMPC}. Sin embargo, respecto a la eficiencia se muestra que el mejor es el CLEDA_{HC+MMPC} que obtuvo una convergencia con 13400 evaluaciones como promedio, siendo menos evaluaciones que el resto.

Tabla 5. Comparación de los resultados obtenidos con CLEDA con otros algoritmos en la función FQuadratic

<i>Algoritmo</i>	<i>% de veces que encontró el óptimo</i>	<i>Media de la cantidad de evaluaciones</i>	<i>Media de los máximos alcanzados</i>
CLEDA _{HC+MMPC}	100	13400.00±680.55	64
CLEDA _{FDA}	100	14960.00±961.60	64
CLEDA _{HC+IAMB}	67	16421.00±723.86	63.97
CLEDA _{HC+FAST.IAMB}	61	17176.47±632.52	63.95
CLEDA _{HC+INTER.IAMB}	90	16088.89±496.49	63.99
CLEDA _{TABU+MMPC}	100	13440.00±328.31	64
CLEDA _{TABU+GS}	100	15000.00±573.08	64

CLEDA _{TABU+IAMB}	97	16757.90±381.39	63.99
CLEDA _{TABU+FAST.IAMB}	85	16752.94±619.46	63.99

4.2 Resultados experimentales en problemas de planificación de proyectos con restricciones

En esta subsección se presentan los resultados de la aplicación del algoritmo propuesto en la resolución del problema de planificación de proyectos con recursos limitados y tareas que se ejecutan de múltiples modos (MMRCPS). Se realizan dos grupos de experimentos con instancias de las bases de datos “n0.mm” y “c15.mm” del repositorio PSPLib (*Project Scheduling Problem Library*) [15][25] en su variante de múltiples modos.

En los experimentos se comparan:

- Los algoritmos: CLEDA_{FDA}, CLEDA_{HC+INTER.IAMB}, FDA tradicional (TRAD_FDA) con tratamiento de restricciones basada en penalización, en las bases de datos “c15.mm”.
- El algoritmo CLEDA_{FDA}, CLEDA_{HC+INTER.IAMB}, FDA tradicional (TRAD_FDA), hGMEDA [38] y Robust hGMEDA [38] en las instancias n041_1, n042_1, n043_1, n044_1 y n045_1.
- En las comparaciones se consideran la eficacia a partir de la variable “*Makespan*”, y la eficiencia a partir del análisis de la variable “*Tiempo de Ejecución*” (segundos). Se efectúan 20 corridas de los algoritmos.

En la Tabla 6. se comparan los resultados del CLEDA con otros EDA propuestos en el trabajo de Tian y colaboradores en [38] sobre la media de la variable “*Makespan*”.

Tabla 6. Comparación de los resultados respecto a la variable *Makespan* con la bibliografía

Instancia/ Algoritmo	CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA}	TRAD_FDA	Robust hGMEDA	hGMEDA
n041_1	29.8	29.7	39.45	29.9	29.1
n042_1	29.95	29.85	35.65	35.1	34.2
n043_1	33.15	33.05	43.45	39.3	37.9
n044_1	25	25.05	33.15	32.2	31.3
n045_1	46.65	47	57.65	42.7	41.1

En la Tabla 6 se muestra como el CLEDA_{FDA} logra minimizar el *Makespan* en los problemas n042_1 y n043_1 con mejores resultados que el resto de los algoritmos. Por otra parte, el CLEDA_{HC+INTER.IAMB} obtiene ligeramente mejor resultado que el CLEDA_{FDA} en la instancia n044_1 superando de esta manera el resultado del TRAD_FDA, Robust hGMEDA y hGMEDA.

En el segundo grupo de experimentos se analiza el comportamiento de los algoritmos CLEDA_{HC+INTER.IAMB}, CLEDA_{FDA} y una implementación del algoritmo FDA tradicional (TRAD_FDA) en las bases de datos: n041_1, n042_1, n043_1, n044_1, n045_1 y las 10 instancias de la base de datos c15_12, pero esta vez, analizando las dos variables siguientes:

- La eficacia en la búsqueda de soluciones a partir del análisis de la variable “*Makespan*” en 20 corridas. Esta variable representa el tiempo de ejecución del proyecto en días y es el resultado de la optimización del cronograma o plan del proyecto.
- Se evalúa la eficiencia empírica de los algoritmos a partir del análisis de la variable “*Tiempo de Ejecución*” de los algoritmos en las 20 corridas. La unidad de medida de esta variable es segundos.

En las comparaciones se usa la aplicación *IBM SPSS Statistics* versión 25 [14]. En cada caso se aplica primero el análisis descriptivo de las variables. Se realiza la prueba de normalidad empleando el test de Shapiro-Wilk [8] y se concluye que los datos no siguen una distribución normal. Se aplica para la comparación el test no-paramétrico de Wilcoxon [8] para dos muestras relacionadas aplicando simulación de Monte Carlo con intervalo de confianza del 95% y con un valor de significación de 0.05.

Como resultado de las comparaciones, los algoritmos se organizan en grupos de forma tal que:

- Los algoritmos en el mismo grupo no presentan diferencias significativas entre ellos, mientras que los algoritmos en grupos diferentes presentan resultados significativamente diferentes.
- Los algoritmos en el “Grupo A” reportan mejores resultados que los algoritmos en el “Grupo B” tal

que: "resultados Grupo A" >"resultados Grupo B" >"resultados Grupo C".

Las Tablas 7 y 9 muestran los resultados de las comparaciones respecto a la variable “*Makespan*”, mientras que las Tablas 8 y 10 reflejan los resultados respecto a la variable “*Tiempo de Ejecución*” en las bases de datos “n0.mm” y “c15.mm” respectivamente.

Tabla 7. Comparación de los resultados respecto a la variable *Makespan* (instancias n041_1, n042_1, n043_1, n044_1 y n045_1)

<i>Base de datos</i>	<i>Algoritmos</i>		<i>Significación (p-value)</i>	<i>Grupos</i>
n041_1	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.588	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
	CLEDA _{FDA}	TRAD_FDA	0.000	Grupo B: TRAD_FDA
n042_1	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.593	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
	CLEDA _{FDA}	TRAD_FDA	0.000	Grupo B: TRAD_FDA
n043_1	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.317	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
	CLEDA _{FDA}	TRAD_FDA	0.000	Grupo B: TRAD_FDA
n044_1	CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA}	0.317	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA}
	CLEDA _{HC+INTER.IAMB}	TRAD_FDA	0.000	Grupo B: TRAD_FDA
n045_1	CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA}	0.413	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
	CLEDA _{HC+INTER.IAMB}	TRAD_FDA	0.000	Grupo B: TRAD_FDA

Las comparaciones muestran que, respecto a la variable “*Makespan*”, no hay diferencias significativas entre los resultados obtenidos por los algoritmos CLEDA_{FDA} y CLEDA_{HC+INTER.IAMB}, obteniendo mejores resultados que el TRAD_FDA.

Tabla 8. Los resultados respecto a la variable *Tiempo de Ejecución* (instancias n041_1, n042_1, n043_1, n044_1 y n045_1)

<i>Base de datos</i>	<i>Algoritmos</i>		<i>Significación (p-value)</i>	<i>Grupos</i>
n041_1	TRAD_FDA	CLEDA _{FDA}	0.000	Grupo A: TRAD_FDA
	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.756	Grupo B: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
n042_1	CLEDA _{HC+INTER.IAMB}	TRAD_FDA	0.881	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} ,
	CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA}	0.823	TRAD_FDA
n043_1	CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA}	0.191	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA}
	CLEDA _{HC+INTER.IAMB}	TRAD_FDA	0.000	Grupo B: TRAD_FDA
n044_1	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.411	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
	CLEDA _{FDA}	TRAD_FDA	0.000	Grupo B: TRAD_FDA
n045_1	TRAD_FDA	CLEDA _{FDA}	0.000	Grupo A: TRAD_FDA
	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.247	Grupo B: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}

Respecto a la variable “*Tiempo de Ejecución*”, no hay diferencias significativas entre los algoritmos CLEDA_{FDA} y CLEDA_{HC+INTER.IAMB} que obtuvieron mejores resultados en las instancias n042_1, n043_1 y n044_1. En las instancias n041_1 y n045_1, el algoritmo TRAD_FDA logró mejores resultados.

Tabla 9: Resultados de comparación usando Wilcoxon test, variable *Makespan*, base de datos c15_12

<i>Base de datos</i>	<i>Algoritmos</i>		<i>Significación (p-value)</i>	<i>Grupos</i>
c15_12_1	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	1.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} ,
	CLEDA _{FDA}	TRAD_FDA	0.083	TRAD_FDA

c15_12_2	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.497 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA
c15_12_3	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	1.000 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA
c15_12_4	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.655 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA
c15_12_5	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.655 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA
c15_12_6	CLEDA _{HC+INTER.IAMB} CLEDA _{FDA}	CLEDA _{FDA} TRAD_FDA	0.046 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} Grupo B: CLEDA _{FDA} Grupo C: TRAD_FDA
c15_12_7	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.723 0.001	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA
c15_12_8	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.285 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} Grupo B: TRAD_FDA
c15_12_9	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.822 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA
c15_12_10	CLEDA _{HC+INTER.IAMB} CLEDA _{HC+INTER.IAMB}	CLEDA _{FDA} TRAD_FDA	0.581 0.000	Grupo A: CLEDA _{HC+INTER.IAMB} , CLEDA _{FDA} , Grupo B: TRAD_FDA

Respecto a la variable “Makespan” en la base de datos c15_12 no hay diferencias significativas entre los algoritmos CLEDA_{HC+INTER.IAMB} y CLEDA_{FDA}, que además obtuvieron los mejores resultados en el 80% de las instancias. En la instancia c15_12_6, el algoritmo CLEDA_{HC+INTER.IAMB} reportó mejores resultados que el resto de los algoritmos, mientras que en la instancia c15_12_1, no se encontraron diferencias significativas entre los tres algoritmos. En el 90% de las instancias, el TRAD_FDA obtuvo los peores resultados. Se demuestra que, en las bases de datos c15_12, los algoritmos con tratamiento de restricciones en el modelo probabilístico reportan mejores resultados que el FDA clásico.

Tabla 10: Resultados de comparación usando Wilcoxon test, variable *Tiempo de Ejecución*, base de datos c15_12

<i>Base de datos</i>	<i>Algoritmos</i>		<i>Significación (p-value)</i>	
c15_12_1	TRAD_FDA CLEDA _{FDA}	CLEDA _{FDA} CLEDA _{HC+INTER.IAMB}	0.006 0.073	Grupo A: TRAD_FDA Grupo B: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB}
c15_12_2	CLEDA _{FDA} CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB} TRAD_FDA	0.681 0.002	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB} Grupo B: TRAD_FDA
c15_12_3	CLEDA _{FDA} CLEDA _{HC+INTER.IAMB}	CLEDA _{HC+INTER.IAMB} TRAD_FDA	0.005 0.033	Grupo A: CLEDA _{FDA} Grupo B: CLEDA _{HC+INTER.IAMB} Grupo C: TRAD_FDA
c15_12_4	CLEDA _{FDA} CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB} TRAD_FDA	0.601 0.000	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB} Grupo B: TRAD_FDA
c15_12_5	CLEDA _{FDA} CLEDA _{HC+INTER.IAMB}	CLEDA _{HC+INTER.IAMB} TRAD_FDA	0.011 0.014	Grupo A: CLEDA _{FDA} Grupo B: CLEDA _{HC+INTER.IAMB} Grupo C: TRAD_FDA
c15_12_6	CLEDA _{FDA} CLEDA _{HC+INTER.IAMB}	CLEDA _{HC+INTER.IAMB} TRAD_FDA	0.006 0.000	Grupo A: CLEDA _{FDA} Grupo B: CLEDA _{HC+INTER.IAMB} Grupo C: TRAD_FDA
c15_12_7	TRAD_FDA CLEDA _{FDA}	CLEDA _{FDA} CLEDA _{HC+INTER.IAMB}	0.000 0.000	Grupo A: TRAD_FDA Grupo B: CLEDA _{FDA}

				Grupo C: CLEDA _{HC+INTER.IAMB}
c15_12_8	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.737	Grupo A: CLEDA _{FDA} , CLEDA _{HC+INTER.IAMB} TRAD_FDA
	CLEDA _{FDA}	TRAD_FDA	0.903	
c15_12_9	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.011	Grupo A: CLEDA _{FDA} Grupo B: CLEDA _{HC+INTER.IAMB} Grupo C: TRAD_FDA
	CLEDA _{HC+INTER.IAMB}	TRAD_FDA	0.000	
c15_12_10	CLEDA _{FDA}	CLEDA _{HC+INTER.IAMB}	0.001	Grupo A: CLEDA _{FDA} Grupo B: CLEDA _{HC+INTER.IAMB} Grupo C: TRAD_FDA
	CLEDA _{HC+INTER.IAMB}	TRAD_FDA	0.003	

Respecto a la variable “*Tiempo de ejecución*” en la base de datos c15_12, hay diferencias significativas. En las instancias c15_12_1 y c15_12_7, el mejor algoritmo fue el TRAD_FDA siendo el peor en el 70% de las instancias. El algoritmo CLEDA_{FDA} reportó los mejores resultados en el 80% de las instancias, mientras que no se encontraron diferencias significativas entre los algoritmos CLEDA_{FDA} y CLEDA_{HC+INTER.IAMB} en las instancias c15_12_2, c15_12_4 y c15_12_8.

5. CONCLUSIÓN

Se demuestra que los EDA que incluyen el tratamiento de restricciones en el modelo probabilístico reportan resultados superiores en comparación con otros algoritmos reportados en la bibliografía. Esta técnica garantiza la generación de soluciones factibles con mayor eficacia y eficiencia.

Para la optimización de las funciones objetivo, el algoritmo CLEDA_{HC+IAMB} obtuvo los mejores resultados considerando simultáneamente las siguientes variables: porciento de veces que encuentra el máximo, promedio de los máximos encontrados y la eficiencia respecto a la cantidad de evaluaciones. Se reportan resultados positivos también en el CLEDA_{FDA}. En las funciones FOneMax y FPlateau se logran resultados satisfactorios de todos los algoritmos, efecto provocado por la simplicidad de estas funciones. Sin embargo, en la función FSixPeaks se identifican mayores diferencias entre los algoritmos. Respecto a la eficacia en esta función los algoritmos genéticos alcanzaron el valor más cercano al óptimo en 70 ocasiones, pero requirieron un número mayor de evaluaciones. En la función FQuadratic la mayoría de los algoritmos analizados obtuvieron buenos resultados, pero las configuraciones con mejores resultados fueron CLEDA_{FDA} y CLEDA_{HC+MMPC}.

Para la optimización de los problemas *Scheduling*, los algoritmos CLEDA_{HC+INTER.IAMB} y CLEDA_{FDA} obtuvieron los mejores resultados en las instancias de la base de datos “n0.mm” y superiores a los reportados en la bibliografía. Estos mismos algoritmos también alcanzaron resultados satisfactorios en la base de datos “c15_12.mm”, especialmente respecto a la variable *Makespan* donde superaron al FDA tradicional en el 100% de las 10 instancias. Respecto a la eficiencia, el CLEDA_{FDA} logró los mejores resultados que el resto de los algoritmos en el 80% de las instancias.

Como trabajo futuro se pretende aumentar la cantidad de bases de datos y la complejidad de las mismas en la comparación de los algoritmos. También se puede extender el análisis de cada problema desde un enfoque multiobjetivo considerando el balance entre la duración y el costo de ejecución del proyecto.

RECEIVED: APRIL, 2020.

REVISED: MAY, 2020.

REFERENCIAS

- [1] AFSHAR, M. R., SHAHHOSSEINI, V., and SEBT, M. H. (2019): A Genetic Algorithm with a New Local Search Method for Solving the Multimode Resource-constrained Project Scheduling Problem. **International Journal of Construction Management**, 0 ,1–9.
- [2] ASTA, S., KARAPETYAN, D., KHEIRI, A., ÖZCAN, E., and PARKES, A. J. (2016): Combining Monte-Carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem. **Information Sciences**, 373, 476–498.

- [3] AYODELE, M. (2018): Effective and efficient estimation of distribution algorithms for permutation and scheduling problems, PhD Thesis, **Robert Gordon University**.
- [4] AYODELE, M., MCCALL, J., and REGNIER-COUDERT, O. (2016): BPGA-EDA for the multi-mode resource constrained project scheduling problem. **IEEE Congress on Evolutionary Computation (CEC), Vancouver, Canada**, 3417–3424.
- [5] AYODELE, M., MCCALL, J., and REGNIER-COUDERT, O., (2016): RK-EDA: A Novel Random Key Based Estimation of Distribution Algorithm. **14th International Conference on Parallel Problem Solving from Nature (PPSN XIV), Edinburgh**, 849–858.
- [6] AYODELE, M., MCCALL, J., and REGNIER-COUDERT, O. (2017): Estimation of Distribution Algorithms for the Multi-Mode Resource Constrained Project Scheduling Problem. **IEEE Congress on Evolutionary Computation (CEC), San Sebastian, Spain**, 1579–1586.
- [7] BALOUKA, N., and COHEN, I., (2019): A robust optimization approach for the multi-mode resource-constrained project scheduling problem. **European Journal of Operational Research**.
- [8] BOUZA HERRERA, C. N., and SISTACHS VEGA, V. (2004): Estadística. Teoría básica y ejercicios. La Habana: **Editorial Félix Varela**.
- [9] CHAKRABORTTY, R. K., ABBASI, A., and RYAN, M. J. (2020): Multi-mode resource-constrained project scheduling using modified variable neighborhood search heuristic. **International Transactions in Operational Research**, 27(1), 138–167.
- [10] CHEN, Y., SUN, X., GONG, D., ZHANG, Y., CHOI, J., and KLASKY, S. (2017): Personalized Search Inspired Fast Interactive Estimation of Distribution Algorithm and Its Application. **IEEE Transactions on Evolutionary Computation**, 21(4), 588–600.
- [11] DOERR, B., and KREJCA, M. (2020): A Simplified Run Time Analysis of the Univariate Marginal Distribution Algorithm on Leading Ones. ArXiv:2004.04978 [Cs].
- [12] DOERR, B., and ZHENG, W. (2020): Sharp Bounds for Genetic Drift in Estimation of Distribution Algorithms. **IEEE Transactions on Evolutionary Computation**, 99, 1–1.
- [13] ECHEGOYEN, C., SANTANA, R., MENDIBURU, A., and LOZANO, J. A. (2015): Comprehensive characterization of the behaviors of estimation of distribution algorithms. **Theoretical Computer Science**, 598, 64–86.
- [14] III, W. E. W. (2019): Using IBM® SPSS® Statistics for Research Methods and Social Science Statistics. **SAGE Publications**.
- [15] KOLISCH, R., and SPRECHER, A. (1997): PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program. **European Journal of Operational Research**, 96(1), 205–216.
- [16] KREJCA, M. S., and WITT, C. (2020): Theory of Estimation-of-Distribution Algorithms. In: B. Doerr and F. Neumann (Eds.), **Theory of Evolutionary Computation: Recent Developments in Discrete Optimization**. 405–442. Cham: Springer International Publishing.
- [17] LARRAÑAGA, P., ETXEBERRIA, R., LOZANO, J.A., PEÑA, J.M. (1999): Optimization by Learning and Simulation of Bayesian and Gaussian Networks, Technical Report EHU-KZZA-1K-4/99, **University of the Basque Country**, Department of Computer Science and Artificial Intelligence.
- [18] LIN, J., ZHU, L., and GAO, K. (2020): A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. **Expert Systems with Applications**, 140, 112915.
- [19] LING, Z., YU, K., WANG, H., LIU, L., DING, W., and WU, X. (2019): BAMB: A Balanced Markov Blanket Discovery Approach to Feature Selection. **ACM Transactions on Intelligent Systems and Technology**, 10(5), 1–25.
- [20] MA, S., SCHREINER, P., SEAQUIST, E., UGURBIL, M., ZMORA, R., and CHOW, L. (2020): Multiple Predictively Equivalent Risk Models for Handling Missing Data at Time of Prediction: with an Application in Severe Hypoglycemia Risk Prediction for Type 2 Diabetes. **Journal of Biomedical Informatics**, 103, 103379.
- [21] MADERA, J., and OCHOA, A. (2018): Evaluating the Max-Min Hill-Climbing Estimation of Distribution Algorithm on B-Functions. In: Y. Hernández Heredia, V. Milián Núñez, and J. Ruiz Shulcloper (Eds.), **Progress in Artificial Intelligence and Pattern Recognition**. 26–33. Cham: Springer International Publishing.
- [22] MARTÍNEZ-LÓPEZ, Y., MADERA, J., RODRÍGUEZ-GONZÁLEZ, A. Y., and BARIGYE, S. (2019): Cellular Estimation Gaussian Algorithm for Continuous Domain. **Journal of Intelligent and Fuzzy Systems**, 36(5), 4957–4967.

- [23] MÜHLENBEIN, H., and MAHNIG, T. (1999): Convergence Theory and Applications of the Factorized Distribution Algorithm. **Journal of Computing and Information Technology**, 7, 19–32.
- [24] MÜHLENBEIN, H., MAHNIG, T., and OCHOA-RODRÍGUEZ, A. (1999): Schemata, Distributions and Graphical Models in Evolutionary Optimization. **J. Heuristics**, 5(2), 215–247.
- [25] MULTI MODE DATA SETS, [Online]. Available in: <http://www.om-db.wi-tum.de/psplib/getdata.cgi?mode=mm>. **Consulted** 31-01-2019.
- [26] MUTHUKUMARAN, K., SRINIVAS, S., MALAPATI, A., and MURTHY NETI, L. B. (2018): Software Defect Prediction Using Augmented Bayesian Networks (pp. 279–293). Presented at the **Advances in Intelligent Systems and Computing**, Poland: Springer International Publishing AG.
- [27] PELIKAN, M., GOLBERG, D. E., and CANTÚ-PAZ, E. (1999): BOA: The Bayesian optimization algorithm. **Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99**, Orlando, FL, 525–532
- [28] PELIKAN, M., and MUEHLENBEIN, H. (1999): The Bivariate Marginal Distribution Algorithm. In: R. Roy, T. Furuhashi, and P. K. Chawdhry (Eds.), **Advances in Soft Computing**. 521–535. Springer London.
- [29] PROJECT MANAGEMENT INSTITUTE, INC. (2017): A guide to the project management body of knowledge (PMBOK guide) (Sixth edition). Newtown Square, Pennsylvania 19073-3299 USA: **Project Management Institute**.
- [30] PUPO, I. P., VACACELA, R. G., PÉREZ, P. P., MAHDI, G. S. S., and PEÑA, M. (2020): Experiencias en el uso de técnicas de softcomputing en la evaluación de proyectos de software. **Investigación Operacional**, 41(1), 108–119.
- [31] RAO, M. S., PRASAD, K. L., and ANUSHA, P. (2019): Automatic Timetable Generation Using PBIL Algorithm. **I-Manager’s Journal on Information Technology**, 8(2), 31.
- [32] ROSŁON, J. H., and KULEJEWSKI, J. E. (2019): A hybrid approach for solving multi-mode resource-constrained project scheduling problem in construction. **Open Engineering**, 9(1), 7–13.
- [33] SANTANA, R. (2017): Gray-box optimization and factorized distribution algorithms: where two worlds collide. ArXiv:1707.03093 [Cs].
- [34] SHAHNAZARI-SHAHREZAEI, P., ZABIHI, S., and KIA, R. (2017): Solving a Multi-Objective Mathematical Model for a Multi-Skilled Project Scheduling Problem by Particle Swarm Optimization and Differential Evolution Algorithms. **Industrial Engineering and Management Systems**, 16(3), 288–306.
- [35] SOLIMAN, O. S., and ELGENDI, E. A. R. (2014): A Hybrid Estimation of Distribution Algorithm with Random Walk Local Search for Multi-mode Resource-Constrained Project Scheduling problems. **International Journal of Computer Trends and Technology**, 8(2), 57–64.
- [36] STITI, C., and DRISS, O. B. (2019): A new approach for the multi-site resource-constrained project scheduling problem. **Procedia Computer Science**, 164, 478–484.
- [37] SWINGLER, K. (2019): Learning and Searching Pseudo-Boolean Surrogate Functions from Small Samples. **Evolutionary Computation**, 1–22.
- [38] TIAN, J., HAO, X., and GEN, M. (2019): A hybrid multi-objective EDA for robust resource constraint project scheduling with uncertainty. **Computers and Industrial Engineering**, 130, 317–326.
- [39] TIAN, J., HAO, X., and MURATA, T. (2014): Markov Network based EDA and Robust Project Scheduling with Multi-mode Resource Constraint and Rework Uncertainty. **3rd Asian Conference on Information Systems** (370).
- [40] TIRKOLAEI, E. B., GOLI, A., HEMATIAN, M., SANGAIAH, A. K., and HAN, T. (2019): Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. **Computing**, 101(6), 547–570.
- [41] WANG, L., and FANG, C. (2012): A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem. **Expert Systems with Applications**, 39(3), 2451–2460.
- [42] YU, K., LIU, L., LI, J., DING, W., and LE, T. D. (2019): Multi-Source Causal Feature Selection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 1–1.
- [43] ZAMANI, R. (2019): An effective mirror-based genetic algorithm for scheduling multi-mode resource constrained projects. **Computers and Industrial Engineering**, 127, 914–924.
- [44] ZHOU, Z., ANGELIDIS, E., BOHN, D., and ROSE, O. (2017): A Two-phase Genetic Algorithm to Solve a Multi-objective Problem for Complex Assembly Lines. **Simulation in Produktion Und Logistik 2017**, 89.