

ALGORITMO DE ESTIMACIÓN DE DISTRIBUCIÓN CON TRATAMIENTO DE RESTRICCIONES EN EL MODELO PROBABILÍSTICO EN PROBLEMAS DE SCHEDULING

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

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

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

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

ABSTRACT

Proper management of projects has become an element of vital importance for the solution to problems of society. In this context, there are two main objectives to be solved: the construction of project plans which may comply with the constraints of the problem and the optimization in terms of cost and time of those plans. To this end, all activities of project must be organized in such way that the constraints related to the precedence among them and the availability of renewable and non-renewable resources at each instant of time are met. The aim of this paper is to present an Estimation of Distribution Algorithm (EDA), which incorporates the handling of the constraints in the probabilistic model, for the construction of optimal or quasi-optimal project plans. For the validation of the algorithm, authors used both the PSPLib database repository for the development of the scheduling research and the databases of projects of the Repository for Research in Project Management, University of Infomatic Sciences.

KEYWORDS: Scheduling, Estimation of Distribution Algorithms, Optimization, Constraints Handling.

MSC: 90B35

RESUMEN

La adecuada gestión de los proyectos, se ha convertido en un elemento de vital importancia para la solución a problemáticas de la sociedad. En este contexto, existen dos objetivos fundamentales a resolver: la construcción de planes de proyectos que cumplan con las restricciones del problema en cuestión y la optimización respecto a los objetivos costo y tiempo de dichos planes. Para ello, todas las actividades del proyecto han de ser organizadas de tal modo que se cumplan las restricciones relacionadas con la precedencia entre las mismas y la disponibilidad de los recursos renovables y no renovables en cada instante de tiempo. El objetivo de este trabajo es presentar un algoritmo evolutivo con estimación de distribución (EDA), que incorpora el tratamiento de restricciones en el modelo probabilístico, para la construcción de planes óptimos o cuasi óptimos de proyectos. Para la validación del algoritmo se emplean el repositorio de bases de datos PSPLib para el desarrollo de investigaciones de "scheduling" y además bases de datos de proyectos del Repositorio de Investigaciones en Gestión de Proyectos de la Universidad de las Ciencias Informáticas.

PALABRAS CLAVES: Planificación de proyectos, algoritmos con estimación de distribución, optimización, tratamiento de restricciones.

1. INTRODUCCIÓN

La gestión de proyectos mantiene una tendencia creciente en la resolución de problemas en distintas esferas de la sociedad. En este escenario es esencial el desarrollo de planificaciones que permitan al gestor hacer estimaciones razonables de recursos, del costo y del tiempo del proyecto. Por ello, la planificación de forma óptima o cuasi óptima de los proyectos es muy importante y está matizada por los siguientes factores:

- Existen un conjunto de restricciones asociadas a la precedencia de las actividades que determinan la factibilidad o no de las soluciones.

¹gaafarsadeq@yahoo.com

- Generalmente, los proyectos se realizan con una cantidad limitada de recursos renovables, recursos que pueden ser usados solo una vez en cada instante del proyecto, pero más de una vez durante el tiempo de vida del proyecto, como los recursos humanos y el equipamiento.
- Existen también recursos no renovables limitados que se caracterizan porque solo pueden ser usados una vez durante el tiempo de vida del proyecto; ejemplo de esto son los materiales de papel, el combustible, etc.
- Además, en los proyectos existen varios modos para la realización de cada actividad que influyen directamente en la cantidad de recursos a emplear y en el tiempo de ejecución de la misma.

Los problemas de planificación de proyectos se clasifican en cuatro categorías fundamentales y ellas son:

- *Problema de planificación de proyectos con recursos limitados (RCPSP)*: su objetivo es hallar el momento de inicio de cada actividad de tal forma que se minimice el tiempo de duración del proyecto [8]. La modelación del problema incluye como restricción la cantidad de recursos disponibles para el proyecto, pero no distingue entre las características específicas de los recursos que pueden influir en el tiempo de duración de las actividades.
- *Problema de planificación de múltiples proyectos con recursos limitados (RCMPSP)*: implica el desarrollo simultáneo de múltiples proyectos con recursos limitados [9]. Este problema es similar al anterior, pero considerando situaciones donde se quiere optimizar múltiples planes de proyectos con un fondo de recursos compartidos.
- *Problema de planificación de proyectos con recursos limitados y actividades que se pueden realizar de múltiples modos (MMRCPSP)*: involucra la selección de un modo de ejecución o procesamiento para cada actividad y la asignación de un tiempo de inicialización o finalización de las actividades [3][4], de tal manera que no se violen las relaciones de precedencia ni se sobrepase la disponibilidad de los recursos y la duración del proyecto se minimice. A través de los modos permite modelar que recursos similares, por ejemplo, personas, pero con diferentes características pueden influir significativamente en el tiempo de duración de las actividades.
- *El problema de planificación de múltiples proyectos con recursos limitados y múltiples modos (MMRCMPSP)*: combina los dos conceptos anteriores [25].

En la resolución de estos problemas se han empleado varias técnicas metaheurísticas [25]. Se analizan a continuación los resultados reportados en la bibliografía. En [35][18] se emplean algoritmos basados en la metaheurística enjambre de partículas (PSO). En particular Medrano en [18] propone los algoritmos PSO-J y el PSO-R para la solución de problemas RCMPSP. En esta propuesta no se tiene en cuenta los diferentes modos y su influencia en el balance entre costo y tiempo; además, ubica a todas las actividades en un único listado y no propone cómo manejar el enfoque de múltiples proyectos.

En [5][2] y [26] se propone el uso de la búsqueda tabú para resolver el problema RCPSP de manera iterativa. En este escenario se destacan dos enfoques, el primero basado en la generación de nuevas soluciones en las vecindades de las existentes y el segundo que genera nuevas soluciones a partir de la información de la ruta crítica. Luego Klein [15] propone la llamada Búsqueda Tabú Reactiva (*Reactive Tabu Search*), basada en la representación serial de actividades mediante un Esquema Generador de Soluciones (por sus siglas en inglés, *Schedule Generator Scheme*) (SGS). Las vecindades de soluciones se obtienen con movimientos de intercambio que incluyen desplazamientos de los predecesores y sucesores de las actividades intercambiadas. Estas soluciones no enfrentan el problema de múltiples modos.

Otros autores [13] siguen enfoques híbridos y combinan varias metaheurísticas para resolver el problema MMRCPSP. En particular Gómez propone el uso del algoritmo MOVNS (*Multiobjective Variable Neighborhood Search*) que comienza con una solución S y genera una vecindad para esta solución a partir de cambios pequeños en la solución S . Para la construcción de la vecindad, el autor utiliza dos métodos: el primero consiste en intercambiar actividades en la secuencia de la solución S , mientras que el otro es provocar una mutación de la solución a partir de insertar una actividad en la secuencia. Como principal señalamiento a esta estrategia está en el tratamiento de las restricciones asociadas a la precedencia y límites en el uso de recursos renovables y no renovables.

Por otra parte, Tirkolaee y colaboradores en [32] proponen el algoritmo MOSA (*Multi-objective Simulated Annealing*) para la solución del problema MMRCPSP utilizando una simple función de probabilidad que intenta generar soluciones basadas en el óptimo de Pareto. Esta técnica no está diseñada para el tratamiento de restricciones dentro del modelo probabilístico y además la búsqueda parte de un solo punto, elemento que provoca que se consuma más tiempo para lograr la convergencia.

La metaheurística más empleada en la resolución de problemas de planificación son los algoritmos genéticos (AG) [11][34][31][17]. A pesar de la versatilidad de AG, en su comportamiento tradicional están concebidos para el trabajo en espacios de búsqueda con variables independientes [27]. Wang et al. en [34] presentan un AG para la optimización multiobjetivo en un entorno multiproyecto, se consideran como objetivos la duración total y los costos de financiamiento determinados a partir de la cadena crítica. El enfoque multiobjetivo que emplean los autores es apropiado, pero el método que emplean no maneja adecuadamente el trabajo con las restricciones asociadas a los recursos limitados o a las asociadas a la relación entre las actividades. En [6] se proponen también los AG pero bajo un enfoque centrado en la minimización de los recursos renovables y no renovables sin dar un peso adecuado al alcance del proyecto, la precedencia entre actividades o a los diferentes modos de ejecución de las mismas. Zughen en [36] propone un AG para solucionar problemas MMRCMPSP en la solución de problemas de líneas de ensamblaje pero este algoritmo no puede ser aplicado de forma natural a la gestión de proyectos porque no considera las relaciones de precedencia entre las actividades ni la existencia de recursos renovables y no renovables. Por su parte Schmidt y colaboradores en [31] emplean un AG para resolver el problema RCMPSP, como elemento positivo se basan en el análisis de la prioridad de los proyectos para la planificación de las actividades pero emplean en la búsqueda operadores de cruzamiento uniforme en su variante estándar, hecho que afecta las obtención de soluciones factibles respecto a las relaciones entre las actividades.

En la solución de problemas de planificación también se reporta el uso de los algoritmos con estimación de distribución (EDA). En [1][4] se propone el algoritmo RK-EDA diseñado para resolver problemas basados en permutaciones y lo aplican en la solución del problema de asignación de recursos en las actividades de planificación. En particular el tratamiento de las restricciones en los EDA es un problema poco tratado, esto se evidencia en el estudio realizado por Coello en 2018 [10] asociado al tratamiento de las restricciones en las metaheurísticas. En este trabajo, de un análisis de 1438 publicaciones se encuentra que solo ocho reflejaban artículos del empleo de algoritmos de estimación de distribución, siendo este particular un punto abierto a la investigación.

En general en las soluciones encontradas en la bibliografía no se tratan adecuadamente las restricciones asociadas a los recursos limitados o a la relación que existe entre las actividades. Estas deficiencias provocan la generación de soluciones no factibles durante el proceso de búsqueda y limita la aplicabilidad de los mismos en la resolución de problemas de planificación de forma efectiva. Como una alternativa a esta problemática se propone, en este trabajo, la solución al problema de planificación de proyectos a partir de los algoritmos de estimación de distribuciones, pero incorporando el tratamiento de las restricciones en el aprendizaje del modelo probabilístico.

El trabajo se estructura de la siguiente forma: en la sección 2 se presenta la modelación matemática del problema de optimización; en la sección 3 se expone el diseño del algoritmo propuesto para la construcción de planes óptimos o cuasi óptimos y en la sección 4 se analizan los resultados de la investigación y finalmente se presentan las conclusiones del trabajo.

2. NOTACIÓN Y MODELACIÓN MATEMÁTICA DEL PROBLEMA DE OPTIMIZACIÓN

En el entorno de esta investigación trataremos el problema de planificación de proyectos con recursos limitados y actividades que se pueden realizar de múltiples modos (MMRCPS). Este problema en particular ha sido presentado por diversos autores como dos subproblemas: el primer subproblema asociado a la asignación del modo, a las actividades, y el otro, a la asignación de recursos como parte de las actividades de planificación. Como parte de la notación se plantea que:

- Un proyecto contiene un conjunto de actividades $j \in J$. Estas actividades tienen un conjunto $M = \{1, \dots, m\}$ modos de procesamiento.
- Cada actividad j puede consumir r_{jmk} unidades de recurso del tipo k , según el modo $m \in M$ en que se ejecute.
- Existen $k \in K$ tipos de recursos, cuya disponibilidad en el horizonte T del proyecto es R_k , tal que $K = K_v \cup K_{nv}$. K_v representa el conjunto de los recursos renovables en el contexto del proyecto (personal y equipamiento) y K_{nv} de la categoría no renovable (papel u otros).

En el MMRCPS existen dos decisiones: la selección de un modo de procesamiento para cada actividad (asignación de modo) y la determinación del tiempo de inicio de cada actividad. Teniendo en cuenta esto, Talbot en [28] propone un modelo lineal con variables binarias donde se definen las variables de la forma

siguiente: $x_{jmt} = 1$ si la actividad j se ejecuta en el modo m y comienza en el instante $t \in [es_j, ls_j]$. Donde es_j representa el tiempo de comienzo más temprano de la actividad j . El valor ls_j es el tiempo de comienzo más tardío en que puede iniciar una actividad j . La variable de decisión toma el valor $x_{jmt} = 0$ en otro caso.

La formulación de este problema siguiendo un modelo lineal en enteros se representa de la siguiente forma:

$$\text{minimizar } \sum_{m \in M} \sum_{t=es_n}^{ls_n} tx_{jmt} \quad (2.1)$$

Donde se minimiza el tiempo de finalización t de todas las actividades, sujeta a las siguientes restricciones:

$$\sum_{m \in M} \sum_{t=es_j}^{ls_j} x_{jmt} = 1 \quad \forall j \in J \quad (2.2)$$

$$\sum_{m \in M} \sum_{t=es_h}^{ls_h} (t + d_{hm})x_{hmt} \leq \sum_{m \in M} \sum_{t=es_j}^{ls_j} tx_{jmt} \quad \forall (h, j) \in J, h \in H \quad (2.3)$$

$$\sum_{j \in J} \sum_{m \in M} r_{jmk} \sum_{q=\max\{t-d_{jm}, es_j\}}^{\min\{t-1, ls_j\}} x_{jmq} \leq |R_k| \quad \forall k \in K_v, t = 1, \dots, T \quad (2.4)$$

$$\sum_{j \in J} \sum_{m \in M} r_{jmk} \sum_{t=es_j}^{ls_j} x_{jmt} \leq |R_k| \quad \forall k \in K_{nv} \quad (2.5)$$

La restricción expresada en la ecuación 2.2 indica que todas las actividades son ejecutadas solo una vez durante el desarrollo del proyecto y que son realizadas en alguno de los modos posibles. La restricción 2.3 se refiere a las restricciones asociadas a la precedencia entre las actividades siendo J el conjunto de todas las actividades y H el conjunto de actividades antecesoras de la actividad j . La restricción 2.4 guarda relación con la cantidad de recursos renovables que se disponen para la ejecución de un proyecto, mientras que la restricción 2.5 expresa la cantidad de recursos no renovables disponibles para la ejecución de un proyecto.

3. DISEÑO DE UN ALGORITMO EDA PARA LA RESOLUCIÓN DEL PROBLEMA

Los algoritmos con estimación de distribución (EDA) constituyen una familia de algoritmos para resolver disímiles problemas de optimización y surgen como alternativa a las dificultades de los algoritmos genéticos (AG). La esencia de estos algoritmos está en el proceso de identificación de las funciones de distribución probabilística y la generación de los individuos a partir de esta distribución.

Los EDA tienen su origen en trabajos de Muehlenbein [22][20][21]. Los análisis de Muehlenbein expresan que el comportamiento de los AG pueden aproximarse con el Algoritmo con Distribución Marginal Univariada (UMDA) [22], extendiendo el trabajo previo desarrollado en [19] y que ambos algoritmos fallan en los escenarios de optimización donde existen interrelación entre las variables [27].

En los EDA, una población inicial de puntos (individuos), es mejorada iterativamente mediante la aplicación de operadores estocásticos. Cada punto codifica una solución que tiene asignado un valor de adecuación (fitness) que lo diferencia de otros puntos de la población con relación a la función objetivo.

En esta sección se presenta el Algoritmo_1 que constituye un algoritmo EDA para la resolución del problema MMRCPSPP presentado con en la sección anterior. En el Algoritmo_1, se exponen los componentes básicos de la propuesta, donde en el paso 7 del algoritmo se construye el modelo probabilístico que define el comportamiento de los individuos, mientras que en el paso 8 se transforma dicho modelo inyectando las restricciones del problema en el propio proceso de aprendizaje.

Algoritmo_1. Se muestra el pseudo-código del algoritmo propuesto en este trabajo

1. $t = 1$
2. Cargar restricciones del problema $R = load_constraints$
3. $P(t) = InicializarPoblacion(R)$

4. EvaluarPoblacion ($P(t)$, R)
5. Mientras no se cumpla la condición de parada hacer
6. $P_s(t)$ = SeleccionarIndividuos ($P(t)$)
7. gp = EstimarDistribucion ($P_s(t)$)
8. gp^R = Incorporar restricciones al modelo (gp)
9. $P(t+1)$ = Generar nuevos individuos a partir de (gp^R)
10. $P(t+1)$ = Reemplazar ($P(t)$, $P(t+1)$)
11. EvaluarPoblación ($P(t+1)$, R)
12. $t = t + 1$
13. Fin Mientras
14. Obtener solución final $P(t)$

Se definen a continuación los siguientes componentes del modelo:

- Las restricciones del problema de optimización se presentan de la siguiente forma:
 - Restricciones de precedencia, donde se conoce de cada actividad cuáles son sus actividades sucesoras.
 - Restricciones de recursos, donde se conoce por cada actividad el tiempo de duración, la cantidad de recursos renovables y no renovables por cada modo posible en que la actividad puede ser ejecutada.
 - Restricciones asociadas a la cantidad de recursos renovables y no renovables disponibles para el desarrollo de proyecto.
- *Definición de la población:* cada población es de tamaño fijo especificado como parámetro y se compone de un conjunto de individuos, donde cada uno de los cuales constituye una solución al problema de planificación. La población inicial representa soluciones generadas aleatoriamente.
- *Diseño del individuo:* cada individuo constituye una posible solución al problema de planificación y está formado por una secuencia de actividades. Cada actividad tiene dos atributos relevantes que son: la fecha de inicio de la actividad (s) y el modo en el cual fue realizada (m). Además, existe un conjunto de atributos complementarios de cada actividad que son estimados a partir del modo y la fecha de inicio. Por ejemplo, la fecha de fin de una actividad se puede calcular a partir de la fecha de inicio y la duración de la misma. A partir de las restricciones, se conocen la duración y la cantidad de recursos renovables y no renovables por cada actividad según el modo en que se ejecuta.
- *Método de evaluación y función objetivo:* se evalúan los individuos tomando en cuenta la función objetivo descrita en la ecuación 3.1, donde:
 - $i \in I$ representa los individuos de cada población.
 - f_j representa el día de finalización de la actividad j .
 - f_{ij} representa la fecha de fin de la actividad j del individuo i .
 - $cost_{ij}$ representa el costo de la ejecución de la actividad j del individuo i calculado a partir de la sumatoria de los costos de los recursos asociados a dicha actividad.

$$\text{minimizar } (O_1, O_2) \quad (3.1)$$

donde:

$$O_1 = \sum_{j \in J} cost_{ij} \quad (3.2)$$

$$O_2 = \max_{j \in J} f_j \quad (3.3)$$

Sujetas a las siguientes restricciones:

$$\sum_{j \in J} g(r_k, j) \leq R_k \quad \forall k \in K_{nv} \quad (3.4)$$

$$\sum_{j \in J} g(r_k, j, t) \leq 1 \quad \forall k \in K_v, t = 1, \dots, T \quad (3.5)$$

$$s_j \leq f_j \quad \forall j \in J \quad (3.6)$$

$$f_h \leq s_j \quad \forall h \in H, \forall j \in J \quad (3.7)$$

La ecuación 3.4 presenta la restricción que garantiza que no se exceda el uso de los recursos materiales y servicios disponibles. Donde $g(r_k, j)$ representa la cantidad de veces que se está usando el recurso $k \in K_{nv}$ considerando todas las actividades $j \in J$ durante el desarrollo del proyecto.

La ecuación 3.5 presenta la restricción que garantiza la disponibilidad de los recursos humanos y equipamiento r_k en el momento t para la ejecución de la actividad $j \in J$ (el recurso $k \in K_v$ no es compartido para más de una actividad al mismo momento t). $g(r_k, j, t)$ representa la cantidad de veces que se está usando el recurso k considerando todas las actividades $j \in J$ en un mismo tiempo t .

La ecuación 3.6 representa la restricción que el momento de inicio de la actividad j siempre es menor que el día de finalización de la misma.

La ecuación 3.7 representa la restricción que garantiza que no se violen las relaciones de precedencia entre las actividades, donde h es el conjunto de las actividades antecesoras de j .

- *Definición de la estrategia de selección:* método de selección basado en la optimización Pareto [12] considerando los objetivos costo y tiempo. El proceso de selección se realiza por iteraciones hasta completar una cantidad igual al 30% de la población. En cada iteración se seleccionan las soluciones dominantes que son incluidas en la población seleccionada, estas soluciones dominantes dejan de ser consideradas en la siguiente iteración del mismo proceso de selección.
- *Definición de la condición de parada:* encontrar el óptimo o alcanzar el máximo número de generaciones.

4. RESULTADOS Y DISCUSIÓN

Para demostrar la aplicabilidad de la propuesta se realizan dos grupos de experimentos:

1. En el experimento 1 se comparan los resultados del Algoritmo_1 con un algoritmo genético y con los reportados en el repositorio de bases de datos “j30 1.mm”, “m5 10.mm”, “c15 9.mm”, “c15 10.mm” y “r5 64.mm” del repositorio de PSPLib [23][16] para el problema MMRCPS.
2. En el experimento 2 se comparan los resultados del Algoritmo_1 con tres implementaciones de un algoritmo genético en un caso de estudio real, tomado del Repositorio de Datos para Investigaciones en Gestión de Proyectos de la Universidad de Ciencias de Informáticas [24].

En el primer grupo de experimentos se compara el Algoritmo_1 con un algoritmo genético AG_PSP. Se centra en buscar soluciones que minimicen además del tiempo, los recursos empleados y el costo de su realización de forma simultánea. En este caso, se empleó la optimización de planes basada en la ecuación (3.1). En particular, se realizan pruebas con las siguientes bases de datos:

- “c15 9.mm” y “c15 10.mm” que contienen 18 actividades cada una con múltiples modos, 2 tipos de recursos no renovables y 2 tipos de recursos renovables. Esta base de datos es empleada con frecuencia en numerosas investigaciones.
- “m5 10.mm” que contiene 18 actividades y hasta 5 modos para cada una de ellas. Esta base de datos se escoge porque representa la base de datos con el mayor número de modos que se dispone en PSPLib.
- “r5 64.mm” que contiene 18 actividades con múltiples modos, 5 tipos de recursos renovables, 5 tipos de recursos no renovables y los costos asociados. Se escoge esta base de datos porque representa la base de datos disponible en PSPLib con el mayor número de tipo de recursos.
- “j30 1.mm” que contiene 30 actividades con múltiples modos, 2 tipos de recursos no renovables, 2 tipos de recursos renovables y los costos asociados. Al igual que “c15 9.mm” y “c15 10.mm”, esta base de datos es empleada con frecuencia en este tipo de investigaciones.

La población inicial fue construida de forma aleatoria con un tamaño de 100 individuos. Se selecciona el 30% de los individuos para la generación de la nueva población. En el caso del AG_PSP el cruzamiento es de un punto y la probabilidad de cruzamiento es de 0.8, mientras que la probabilidad de mutación es 0.2. De una generación a la siguiente sobreviven los mejores individuos y la condición de parada son de 100 generaciones. Se aplicó el test de Wilcoxon [7][14] para comparar los resultados de los diferentes algoritmos, en todas las bases de datos, considerando muestras apareadas con intervalos de confianza del 99% y una diferencia significativa con nivel de significación (p-value) menor que 0.05. Para ello se emplea la herramienta RStudio [29] en su versión 0.99.1249. La comparación se realiza respecto a la variable tiempo

La Tabla 1 muestra como el Algoritmo_1 supera al AG_PSP en la mayoría de las instancias de la base de datos “c15 9.mm”.

Tabla 1. Resultados comparativos del Algoritmo_1 vs el AG_PSP con diferentes instancias de la base de datos “c15 9.mm”

Instancia	AG_PSP		Algoritmo_1		min_t reportado en la bibliografía
	min_t	min_cost	min_t	min_cost	
c15 9_1	18	300	18	294	18
c15 9_2	22	288	21	290	29
c15 9_3	20	288	20	278	22
c15 9_4	19	286	15	286	17
c15 9_5	21	264	21	264	21
c15 9_6	16	274	16	260	20
c15 9_7	22	304	22	300	24
c15 9_8	31	324	31	316	34
c15 9_9	28	272	27	258	28
c15 9_10	29	246	28	250	32

Respecto a la variable tiempo, el Algoritmo_1 reportó mejores resultados que el AG_PSP en el 40% de las instancias y resultados iguales en el 60% restantes. Aplicando el test de Wilcoxon para comparar los resultados de los diferentes algoritmos considerando muestras apareadas respecto a la variable tiempo, se obtuvieron los siguientes resultados, ver Tabla 2:

Tabla 2. Resultados de la comparación de los algoritmos respecto a la variable “tiempo”, en la base de datos “c15 9.mm”, aplicando el test de Wilcoxon [7][14]

Algoritmos	p-value	Observaciones
Algoritmo_1 vs AG_PSP	0.02939	existen diferencias significativas
AG_PSP vs Reportado_bibliografía	0.02034	existen diferencias significativas

En este caso se demuestra que en la base de datos “c15 9.mm” los algoritmos quedaron agrupados de la siguiente forma: *Grupo 1*= Algoritmo_1, *Grupo 2*= AG_PSP y *Grupo 3*= Reportado_bibliografía, lo que significa que los algoritmos del Grupo 1 reportan mejores resultados significativamente que los del Grupo 2 y los del Grupo 2 mejores que los del Grupo 3. Es importante destacar que, para estas instancias, el Algoritmo_1 logró mejores resultados que los reportados en la bibliografía.

Respecto a la variable costo, el Algoritmo_1 obtuvo resultados mejores que el AG_PSP en el 60% de los casos. Aplicando el test de Wilcoxon (Algoritmo_1 vs AG_PSP), se obtuvo p-value = 0.02091 y se demuestra que el Algoritmo_1 reportó mejores resultados que el AG_PSP.

Para la base de datos “c15 10.mm”, la Tabla 3 muestra como el Algoritmo_1 supera al AG_PSP en la mayoría de las instancias.

Tabla 3. Resultados comparativos del Algoritmo_1 vs el AG_PSP con diferentes instancias de las bases de datos “c15 10.mm”

Instancia	AG_PSP		Algoritmo_1		min_t reportado en la bibliografía
	min_t	min_cost	min_t	min_cost	
c15 10_1	21	306	21	302	21
c15 10_2	18	302	15	306	17
c15 10_3	24	280	23	270	23
c15 10_4	36	312	36	310	39
c15 10_5	16	290	13	278	13
c15 10_6	32	306	30	310	32

c15 10_7	15	306	15	306	15
c15 10_8	18	284	16	290	16
c15 10_9	14	256	14	258	12
c15 10_10	16	302	12	300	14

Respecto a la variable tiempo, el Algoritmo_1 reportó mejores resultados que el AG_PSP en el 60% de las instancias y resultados iguales en el 40% restantes. Aplicando el test de Wilcoxon para muestras apareadas respecto a la variable tiempo, se obtuvieron los siguientes resultados, ver Tabla 4:

Tabla 4. Resultados de la comparación de los algoritmos respecto a la variable “tiempo”, en la base de datos “c15 10.mm”, aplicando el test de Wilcoxon[7][14]

Algoritmos	p-value	Observaciones
Algoritmo_1 vs AG_PSP	0.008474	existen diferencias significativas
AG_PSP vs Reportado_bibliografía	0.9	no existen diferencias significativas

En este caso se demuestra que en la base de datos “c15 10.mm” los algoritmos quedaron agrupados de la siguiente forma: *Grupo 1*= Algoritmo_1, *Grupo 2*= (AG_PSP, Reportado_bibliografía), lo que significa que los algoritmos del Grupo 1 reportan mejores resultados significativamente que los del Grupo 2. Es importante destacar que, para estas instancias, el Algoritmo_1 logró mejores resultados que los reportados en la bibliografía.

Respecto a la variable costo, el Algoritmo_1 obtuvo mejores resultados en el 50% de los casos, y peores resultados en el 40% restantes en comparación con el algoritmo AG_PSP. Al aplicar el test de Wilcoxon (Algoritmo_1 vs AG_PSP), se obtuvo p-value = 0.3381, lo que demuestra que no existen diferencias significativas respecto a esta variable.

La Tabla 5 muestra como el Algoritmo_1 supera al AG_PSP en la mayoría de las instancias de la base de datos “m5 10.mm”

Tabla 5. Resultados comparativos del Algoritmo_1 con el AG_PSP con diferentes instancias de la base de datos “m5 10.mm”

Instancia	AG_PSP		Algoritmo_1		min_t reportado en la bibliografía
	min_t	min_cost	min_t	min_cost	
m5 10_1	21	284	15	302	15
m5 10_2	16	300	15	296	15
m5 10_3	22	304	16	302	17
m5 10_4	22	302	20	278	19
m5 10_5	14	310	13	302	13
m5 10_6	21	284	20	284	17
m5 10_7	25	316	23	322	23
m5 10_8	16	298	16	304	16
m5 10_9	17	278	14	268	13
m5 10_10	15	286	13	282	15

Respecto a la variable tiempo, el Algoritmo_1 reportó mejores resultados que el AG_PSP en el 100% de las instancias. Respecto a la variable costo, el Algoritmo_1 fue mejor en el 60% de las instancias. En la Tabla 5 se refleja también el mejor tiempo reportado en la bibliografía para estas instancias, donde el Algoritmo_1 logró resultados muy cercanos tomando en cuenta que este último optimiza el tiempo y el costo de forma simultánea. Aplicando el test de Wilcoxon para muestras apareadas respecto a la variable tiempo, se obtuvieron los siguientes resultados, ver Tabla 6:

Tabla 6. Resultados de la comparación de los algoritmos respecto a la variable “tiempo”, en la base de datos “m5 10.mm”, aplicando el test de Wilcoxon[7][14]

Algoritmos	p-value	Observaciones
Reportado_bibliografíavsAlgoritmo_1	0.3401	no existen diferencias significativas
Reportado_bibliografía vs AG_PSP	0.005757	existen diferencias significativas

En este caso se demuestra que en la base de datos “m5 10.mm” los algoritmos quedaron agrupados de la siguiente forma: *Grupo 1*= (Reportado_bibliografía, Algoritmo_1), *Grupo 2*= AG_PSP, lo que significa que los algoritmos del Grupo 1 reportan mejores resultados significativamente que los del Grupo 2. Respecto a la variable costo, el Algoritmo_1 obtuvo mejores resultados en el 60% de los casos, y peores resultados en el 30% restantes en comparación con el algoritmo AG_PSP. Al aplicar el test de Wilcoxon (Algoritmo_1 vs AG_PSP), se obtuvo p-value = 0.08616, y se demuestra que el Algoritmo_1 reportó mejores resultados que el AG_PSP.

La Tabla 7 muestra como el Algoritmo_1 supera al AG_PSP en la mayoría de las instancias de la base de datos r5 64.mm.

Tabla 7. Resultados comparativos del Algoritmo_1 con el AG_PSP con diferentes instancias de la base de datos“r5 64.mm”

Instancia	AG_PSP		Algoritmo_1		min_t reportado en la bibliografía
	min_t	min_cost	min_t	min_cost	
r5 64_1	28	978	28	968	28
r5 64_2	22	1050	21	1104	21
r5 64_3	16	1072	15	1108	15
r5 64_4	17	1078	17	1108	17
r5 64_5	22	1020	22	1064	22
r5 64_6	21	1122	18	1136	18
r5 64_7	35	944	35	924	35
r5 64_8	16	1090	16	1084	16
r5 64_9	15	1120	13	1172	13
r5 64_10	22	1100	22	1070	22

Respecto a la variable tiempo, el Algoritmo_1 reportó mejores resultados que el AG_PSP en el 40% de las instancias y resultados iguales en el 60% restantes. Respecto a la variable costo, el AG_PSP obtuvo mejores resultados. En la tabla se muestra también el mejor tiempo reportado en la bibliografía para estas instancias, donde el Algoritmo_1 logró resultados iguales a pesar de que este último optimiza las variables tiempo y costo de forma simultánea. Aplicando el test de Wilcoxon para muestras apareadas respecto a la variable tiempo, se obtuvieron los siguientes resultados, ver Tabla 8:

Tabla 8. Resultados de la comparación de los algoritmos respecto a la variable “tiempo”, en la base de datos “r5 64.mm”, aplicando el test de Wilcoxon [7][14]

Algoritmos	p-value	Observaciones
Algoritmo_1 vs Reportado_bibliografía	NA	no existen diferencias significativas
Algoritmo_1 vs AG_PSP	0.0328	existen diferencias significativas

En este caso se demuestra que en la base de datos “r5 64.mm” los algoritmos quedaron agrupados de la siguiente forma: *Grupo 1*= (Algoritmo_1, Reportado_bibliografía), *Grupo 2*= AG_PSP, lo que significa que los algoritmos del Grupo 1 reportan mejores resultados significativamente que los del Grupo 2.

Respecto a la variable costo, el Algoritmo_1 obtuvo peores resultados en comparación con el algoritmo AG_PSP. Al aplicar el test de Wilcoxon (Algoritmo_1 vs AG_PSP), se obtuvo p-value = 0.006258, y se demuestra que el AG_PSP reportó mejores resultados que el Algoritmo_1.

La Tabla 9 muestra que el Algoritmo_1 supera al AG_PSP en la mayoría de las instancias de la base de datos j30 1.mm probadas teniendo en cuenta el tiempo y el costo.

Tabla 9. Resultados comparativos del Algoritmo_1 con el AG_PSP con diferentes instancias de “j30 1.mm”

Instancia	AG_PSP		Algoritmo_1	
	min_t	min_cost	min_t	min_cost
j30 1_1	40	508	40	506
j30 1_2	37	592	37	582
j30 1_3	35	596	35	586
j30 1_4	37	610	38	600
j30 1_5	31	604	28	614
j30 1_6	37	574	33	588
j30 1_7	28	634	24	624
j30 1_8	36	500	34	522
j30 1_9	33	614	30	598
j30 1_10	26	602	25	594

Considerando la variable tiempo, el Algoritmo_1 reportó mejores resultados que el AG_PSP en el 60% de las instancias y resultados iguales en otros 30% del total de los casos. Aplicando el test de Wilcoxon para muestras apareadas respecto a la variable tiempo, se obtuvieron los siguientes resultados, ver Tabla 10:

Tabla 10. Resultados de la comparación de los algoritmos respecto a la variable “tiempo”, en la base de datos “j30 1.mm”, aplicando el test de Wilcoxon

Algoritmos	p-value	Observaciones
Algoritmo_1 vs AG_PSP	0.01682	existen diferencias significativas

En este caso se demuestra que en la base de datos “j30 1.mm” los algoritmos quedaron agrupados de la siguiente forma: *Grupo 1* = (Algoritmo_1), *Grupo 2* = AG_PSP. Para esta base de datos no existe resultados reportados en la bibliografía.

Respecto a la variable costo el Algoritmo_1 fue mejor en el 70% de los casos en comparación con el algoritmo AG_PSP. Al aplicar el test de Wilcoxon (Algoritmo_1 vs AG_PSP), se obtuvo p-value = 0.3211, lo que se demuestra que no existen diferencias significativas respecto a esta variable.

El segundo grupo de experimentos responde a la aplicación del Algoritmo_1 en un caso de estudio real que consta de 5 proyectos con 502 actividades y con 21 recursos agrupados en cuatro categorías que están relacionadas con las competencias fundamentales en el desarrollo de software (análisis, diseño, implementación y prueba).

El algoritmo diseñado se va a comparar con algunas configuraciones del AG propuesto por [33] como sigue:

- *AG(Hor.Res)*: cruce en un punto horizontal y mutación basada en la reasignación de recursos.
- *AG(Ver.Res)*: cruce en dos puntos vertical y mutación basada en la reasignación de recursos.
- *AG(Heu.Res)*: cruce heurístico basado en intercambio de recursos y mutación basada en la reasignación de recursos.

En cada algoritmo se emplearon poblaciones de 120 individuos. Se ejecutaron 8 corridas con 80 generaciones cada una. Se compararon los resultados de la media de duración de cada proyecto en días. Se realizó un análisis descriptivo empleando el comando *summary*, ver Tabla 11.

Para la comparación de los resultados de los diferentes algoritmos se aplicó el test de *Wilcoxon* sobre la plataforma de análisis de datos en [29][30], arrojando los siguientes resultados mostrados en la Tabla 12.

Tabla 11. Análisis descriptivo, resultados desde R comando *summary* a partir del tiempo medio de duración de los proyectos (MediaT)

Algoritmo	Min	Median	Mean	Max
<i>AG(Hor.Res)</i>	157.0	161.0	161.6	197.0
<i>AG(Ver.Res)</i>	144.0	149.0	150.6	189.0
<i>AG(Heu.Res)</i>	135.0	137.0	138.2	159.0
<i>Algoritmo_1</i>	104.0	107.0	107.3	121.0

Tabla12. Resultados de la comparación de algoritmos con R

Algoritmos	V	p-value	Observaciones
<i>Algoritmo_1vs AG(Ver.Res)</i>	0	1.797e-05	se encuentran diferencias significativas
<i>AG(Heu.Res) vs AG(Ver.Res)</i>	184.5	0.01634	se encuentran diferencias significativas
<i>AG(Hor.Res) vs AG(Ver.Res)</i>	276	2.753e-0	se encuentran diferencias significativas

En base de la calidad de los resultados obtenidos, se formaron cuatro grupos: Grupo 1= Algoritmo_1, Grupo 2= AG (Ver.Res), Grupo 3= AG (Heu.Res), Grupo 4 = AG (Hor.Res) ; siendo el de mejores resultados el Grupo 1.

A pesar de que las soluciones encontradas son buenas, es importante destacar que en algunos casos las soluciones de los algoritmos genéticos no cumplían con todas las restricciones del problema. Se demuestra de esta forma la aplicabilidad de la propuesta algorítmica para la obtención de planes de proyectos cuasi óptimos. Se recomienda que el proceso de planificación sea un proceso asistido por los algoritmos y las herramientas computacionales, pero verificado y ajustado por expertos humanos especialistas en cada rama. Se considera que esta temática es un campo abierto a la investigación donde se pueden presentar disímiles situaciones para las cuales no hay suficientes soluciones en la bibliografía.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Los algoritmos evolutivos con estimación de distribución constituyen una herramienta robusta y factible, que puede ser aplicada en la resolución de problemas de planificación de proyectos en general y en la construcción de planes óptimos en particular, por considerar las dependencias existentes entre las variables del problema. En la propuesta se incluye el tratamiento de restricciones del problema dentro del modelo probabilístico del algoritmo.

La aplicación del algoritmo propuesto en la resolución del problema de planificación de múltiples modos del tipo MMRCPS, empleando varias bases de datos de la librería PSPLib en su variante multi-modo, logró resultados superiores a otros obtenidos por algoritmos genéticos y otros reportados en la bibliografía, a pesar de que fueron minimizados de forma simultánea el tiempo de ejecución, los recursos empleados y el costo de realización de esta actividad.

La propuesta algorítmica fue aplicada además a un caso de estudio real considerando 5 proyectos obteniendo mejores resultados que los algoritmos genéticos.

Como trabajo futuro se propone la mejora de los algoritmos para la generación de los modelos probabilísticos. Se recomienda seguir profundizando en la resolución a este problema por su complejidad y su aplicabilidad en la gestión de organizaciones orientadas a proyectos.

RECEIVED: MAY, 2019

REFERENCIAS

- [1] AYODELE, M., MCCALL, J. and REGNIER-COUDERT, O. (2016): RK-EDA: A Novel Random Key Based Estimation of Distribution Algorithm. **14th International Parallel Problem Solving from Nature Conference (PPSN XIV), Edinburgh.**
- [2] ARTIGUES, C., MICHELON, P. and REUSSER, S. (2003): Insertion techniques for static and dynamic resource-constrained project scheduling. **European Journal of Operational Research**, 149, 249–267.
- [3] AYODELE, M., MCCALL, J., REGNIER-COUDERT, O. (2016): BPGA-EDA for the multi-mode resource constrained project scheduling problem. **IEEE Congress on Evolutionary Computation (CEC), Vancouver**, 3417–3424.
- [4] 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**, 1579–1586.
- [5] BAAR, T., BRUCKER, P. and KNUST, S. (1999): Tabu Search Algorithms and Lower Bounds for the Resource-Constrained Project Scheduling Problem. In: Voß, S., Martello, S., Osman, I.H. and Roucairol, C. **Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization**. 1–18. Springer US. Boston.
- [6] BEŞIKCI, U., BILGE, Ü. and ULUSOY, G. (2014): Multi-mode resource constrained multi-project scheduling and resource portfolio problem. **European Journal of Operational Research**, 240, 22–31.
- [7] BOUZA HERRERA, C.N. and SISTACHS VEGA, V. (2004): Estadística: teoría básica y ejercicios. **Editorial Félix Varela**, Ciudad de la Habana.
- [8] BRUCKER, P., DREXL, A., MÖHRING, R., NEUMANN, K. and PESCH, E. (1999): Resource-constrained project scheduling: Notation, classification, models, and methods. **European Journal of Operational Research**, 112, 3–41.
- [9] BROWNING, T.R. and YASSINE, A.A. (2010): Resource-constrained multi-project scheduling: Priority rule performance revisited. **International Journal of Production Economics**, 126, 212–228.
- [10] COELLO, C.A.C. (2018): List of references on constraint-handling techniques used with evolutionary algorithms. Disponible en <https://www.cs.cinvestav.mx/~constraint/constbib.pdf>. **Consulted** 31-01-2019.
- [11] DONG, N., GE, D., FISCHER, M. and HADDAD, Z. (2012): A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects. **Advanced Engineering Informatics**, 26, 737–748.
- [12] EMMERICH, M.T.M. and DEUTZ, A.H. (2018): A tutorial on multiobjective optimization: fundamentals and evolutionary methods. **Natural Computing**, 17, 585–609.
- [13] GOMES, H.C., NEVES, F.A. and SOUZA, M.J. (2014): Multi-objective Metaheuristic Algorithms for the Resource Constrained Project Scheduling Problem with Precedence Relations. **Computers & Operations Research**, 44, 92-104.
- [14] HOLLANDER, M., WOLFE, D.A. and CHICKEN, E. (2013): The One-Sample Location Problem. In: Alding, D., Cressie, N., Fitzmaurice, G., Goldstein, H. and Johnstone, I. **Nonparametric Statistical Methods, Third Edition**. 39-112. John Wiley & Sons. Hoboken, New Jersey.
- [15] KLEIN, K. (2000): Project scheduling with time-varying resource constraints. **International Journal of Production Research**, 38, 3937–3952.
- [16] 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, 205–216.
- [17] KUEHN, M., ZAHID, T., VOELKER, M., ZHOU, Z. and ROSE, O. (2016): Investigation of Genetic Operators and Priority Heuristics for Simulation Based Optimization of Multi-Mode Resource Constrained Multi-Project Scheduling Problems (MMRCMPSP). **30th Conference on Modelling and Simulation, Regensburg.**
- [18] MEDRANO, B.E. (2013): Planificación de múltiples proyectos de desarrollo de software utilizando métodos metaheurísticos. Tesis de Maestría, **Universidad de La Habana**, La Habana.

- [19] MÜHLENBEIN, H. and VOIGT, H.-M. (1996): Gene Pool Recombination in Genetic Algorithms. In: Osman, I.H. und Kelly, J.P. **Meta-Heuristics**, 53–62. Springer US. Boston, MA.
- [20] MÜHLENBEIN, H. and PAAß, G. (1996): From recombination of genes to the estimation of distributions I. Binary parameters. In: Voigt H.-M., Ebeling W., Rechenberg I. and Schwefel HP. **Lecture Notes in Computer Science**, 1141, 178-187. Springer, Berlin, Heidelberg.
- [21] MÜHLENBEIN, H., MAHNIG, Y.A. and OCHOA, A. (1999): Schemata, Distributions and Graphical Models in Evolutionary Optimization. **Journal of Heuristics**, 5, 215–247.
- [22] MÜHLENBEIN, H. (1998): The Equation for Response to Selection and its Use for Prediction. **Evolutionary Computation**, 5, 303–346.
- [23] Multi Mode Data Sets [Online]. Disponible en: <http://www.om-db.wi.tum.de/psplib/getdata.cgi?mode=mm>. **Consulted** 31-01-2019.
- [24] PÉREZ, P.P., PUPO, I.P., HECHAVARRÍA, C.R., LUSARDO, C.R., SOSA, R.G. and LÓPEZ, S. T. (2019): Repositorio de datos para investigaciones en gestión de proyectos. **Revista Cubana de Ciencias Informáticas**, 13, 176–191.
- [25] PINHA, D.C., AHLUWALIA, R.S., SENNA, P. (2016): The Combinatorial Multi-Mode Resource Constrained Multi-Project Scheduling Problem. **International Journal of Supply and Operations Management**, 3, 1391–1412.
- [26] POSADA, M.C. (2010): Nuevos métodos meta heurísticos para la asignación eficiente, optimizada y robusta de recursos limitados. Tesis doctoral. **Universitat Politècnica de València**, Valencia.
- [27] RODRÍGUEZ, R.P., JÖNS S. and AGUIRRE, A.H. (2015): Solución de un problema de secuenciamiento en configuración jobshop flexible utilizando un Algoritmo de Estimación de Distribuciones. **Revista Iberoamericana de Automática e Informática industrial**, 12, 49-57.
- [28] TALBOT, F.B. (1982): Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case. **Management Science**, 28, 1197–1210.
- [29] TEAM, Rs., OTHERS. (2015): RStudio: integrated development for R. **RStudio Inc**, 42, 14.
- [30] The Comprehensive R Archive Network. [Online]. Disponible en: <https://cran.r-project.org/>. **Consulted** 14-12-2018.
- [31] THORSTEN, S., MATHIAS, K. and PAUL RICHARD, G. (2017): Design of Project-oriented Calculation Models for Job Priorities by Using a Customized Genetic Algorithm. In: Sigrid W. and Tim P. **Simulation in Produktion und Logistik 2017**, 99-108. Kassel University Press, Kassel.
- [32] 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, 547-570.
- [33] VACACELA, R.G., PUPO, I.P., VILLAVICENCIO, N., PÉREZ, P.P. and LUIS, S.B. (2016): Experiencias usando algoritmos genéticos en la planificación de proyectos. **Revista Cubana de Ciencias Informáticas**, 10, 71–86.
- [34] WANG, W., WANG, X., GE, X. and DENG, L. (2014): Multi-objective optimization model for multi-project scheduling on critical chain. **Advances in Engineering Software**, 68, 33–39.
- [35] ZHANG, H., LI, H. and TAM, C.M. (2006): Permutation-based particle swarm optimization for resource-constrained project scheduling. **Journal of Computing in Civil Engineering**, 20, 141–149.
- [36] ZHOU, Z., ANGELIDIS, E., BOHN D. and ROSE, O. (2017): A Two-phase Genetic Algorithm to Solve a Multiobjective Problem for Complex Assembly Lines. In: Sigrid W. and Tim P. **Simulation in Produktion und Logistik 2017**, 89-98. Kassel University Press, Kassel.