



Revista EIA, ISSN 1794-1237 /
e-ISSN 2463-0950
Año XVII/ Volumen 17/ Edición N.34
Julio-Diciembre de 2020
Reia34023 pág 1-18

Publicación científica semestral
Universidad EIA, Envigado, Colombia

**PARA CITAR ESTE ARTÍCULO /
TO REFERENCE THIS ARTICLE /**

Candia García, C.D.; López Castro,
L.F.; Jaimes Suárez, S.A. (2020).
Selección óptima del portafolio de
proyectos utilizando metaheurísticas
de población y trayectoria meta-
optimizadas. Revista EIA, 17(34),
Julio-Diciembre, Reia34023. [https://
doi.org/10.24050/reia.v17i34.1399](https://doi.org/10.24050/reia.v17i34.1399)

✉ *Autor de correspondencia:*

Candia García, C.D. (Cristian David):
Escuela Colombiana de Ingeniería
Julio Garavito, AK.45 No.205-59
(Autopista Norte). Decanatura
de ingeniería industrial, Bogotá,
Colombia.. Correo electrónico:
cristian.candia@mail.escuelaing.edu.co

Recibido: 03-01-2020
Aceptado: 18-06-2020
Disponible online: 15-11-2020

Selección óptima del portafolio de proyectos utilizando metaheurísticas de población y trayectoria meta-optimizadas

✉ **CRISTIAN DAVID CANDIA GARCÍA¹**
LUIS FRANCISCO LÓPEZ CASTRO¹
SONIA ALEXANDRA JAIMES SUÁREZ¹

1. Escuela Colombiana de Ingeniería Julio Garavito

Resumen

Este artículo aborda el problema de selección de portafolio de proyectos para la adjudicación de interventorías de obra pública a través de concursos de méritos abiertos (CMA) supervisados por el Instituto Nacional de Vías (INVIAS) en Colombia. En esta modalidad, cada concursante presenta un portafolio único de proyectos históricos para cuantificar su experiencia como interventor. Como alternativa al uso de hojas de cálculo en Excel con procedimientos limitados de enumeración exhaustiva, se evaluó un algoritmo genético meta-optimizado (GA) y un procedimiento de búsqueda voraz adaptativo probabilista meta-optimizado (GRASP) para el caso de estudio de una Compañía con 207 contratos de trayectoria en el sector. Ambas metaheurísticas consiguieron encontrar puntajes de valoración óptimos para distintas instancias de prueba, sin embargo, el algoritmo GA presentó un mejor desempeño consistentemente en todas las instancias de evaluación, encontrando en algunos casos hasta 10 portafolios óptimos en menos de 9 minutos.

Palabras clave: algoritmo genético, GRASP, meta-optimización, selección de portafolio de proyectos.

Optimal Project Portfolio Selection Using Meta-Optimized Population and Trajectory-Based Metaheuristics

Abstract

This article addresses the problem of project portfolio selection for the awarding of public works audits through open merit competitions (CMA) supervised by the National Roads Institute in Colombia - INVIAS. In this modality, each competitor presents a unique portfolio of historical projects to quantify its experience. As an alternative to the use of

Excel spreadsheets with limited procedures of exhaustive enumeration, a meta-optimized genetic algorithm (GA) and a meta-optimized greedy randomized adaptive search procedure (GRASP) were evaluated for the case study of a company with 207 experience career contracts. Both metaheuristics were able to find optimal assessment scores for different test instances, however, the GA algorithm consistently performed better in all assessment instances, finding in some cases up to 10 optimal portfolios in less than 9 minutes.

Keywords: genetic algorithms, GRASP, meta-optimization, project portfolio selection.

1. Introducción

Los lineamientos normativos en materia de contratación para la ejecución de obras públicas en Colombia, confieren a las Entidades Estatales la tarea de monitorear de manera permanente la utilización de los recursos destinados a la ejecución de dichas obras a través de un ente interventor (Colombia Compra Eficiente, 2017). La selección de la entidad interventora se realiza típicamente a través de un Concurso de Méritos Abierto (CMA), en el que se realiza una valoración cuantitativa de la experiencia del postulante, tomando como referencia un portafolio de proyectos históricos que el concursante ha ejecutado previamente durante su trayectoria en el sector.

El número de experiencias históricas que el candidato selecciona para conformar el portafolio de postulación, depende de las condiciones establecidas por la Entidad Estatal; para el caso de INVIAS el portafolio deberá tener de 4 a 6 proyectos en el portafolio (Instituto Nacional de Vías, 2017). Por lo anterior, el concursante se enfrenta a la tarea de seleccionar un grupo de proyectos que le permita maximizar el puntaje de valoración teniendo en cuenta las restricciones del CMA.

El número total de alternativas de selección del portafolio para el postulante está condicionado por la cantidad histórica de proyectos que ha ejecutado y el número de contratos admitidos en el portafolio de proyectos. De esta manera, el número total de posibles alternativas está dado por la siguiente expresión:

$$NTC = C_{\min}^N + C_{\min+1}^N + \dots + C_{\max}^N \quad (1)$$

Donde:

- NTC: Número total de combinatorias para portafolios con un tamaño mínimo *min* y un tamaño máximo *max*.
- N: Número total de contratos ejecutados por el participante a lo largo de su trayectoria.
- min: Número mínimo de proyectos permitidos en el portafolio (4 para el caso de INVIAS).
- max: Número máximo de proyectos permitidos en el portafolio (6 para INVIAS).
- C: Número de combinaciones sin repetición.

Este trabajo toma como referencia el caso de estudio de una *Compañía* que trabajaba con un esquema exploración exhaustiva formulado de manera rudimentaria en hojas de Excel, restringidas a la exploración completa del espacio de solución para un máximo de 30 contratos históricos, debido a las limitaciones de almacenamiento y velocidad de cálculo de las hojas de Excel con las que se trabaja; a pesar de usar solo 30 contratos históricos, el *N* actual de la *Organización* a más de 207 proyectos, motivo

por el cual descarta gran parte de su experiencia histórica bajo el riesgo de no considerar soluciones que puedan representar mejores puntajes.

Este problema no ha sido documentado en investigaciones anteriores, sin embargo, es posible establecer que el problema literario con mayor proximidad al caso de interés es el problema teórico de selección de portafolio de proyectos conocido en inglés como Project Portfolio Selection Problem (PPSP). De manera general, el PPSP consiste en seleccionar un portafolio de proyectos que serán ejecutados en un horizonte de tiempo definido, teniendo en cuenta múltiples criterios y restricciones que a menudo son conflictivos como en el caso de la rentabilidad y el riesgo asociado a la inversión para cada proyecto que compone el portafolio. La naturaleza combinatoria del PPSP, relacionado con el número de posibles alternativas de selección de proyectos que integrarán un portafolio determinado, en adición a que no existen algoritmos determinísticos que puedan resolver el problema en tiempos polinomiales en proporción al tamaño de la instancia así como en el caso de estudio, confieren a la familia de problemas derivada del PPSP una complejidad computacional NP-Hard (Mira *et al.*, 2012).

Debido a la naturaleza combinatoria y la complejidad NP-Hard del problema de estudio, los métodos de solución metaheurísticos han sido documentados por numerosas investigaciones como alternativas de solución promisorias, debido a que ofrecen buenas soluciones en tiempos computacionales admisibles para la mayor parte de aplicaciones.

Se ha documentado el uso metaheurísticas de población basadas en GA para la selección de óptima del portafolio de proyectos teniendo en cuenta múltiples objetivos como la rentabilidad de la inversión y el riesgo asociado a la ejecución de cada proyecto presente en el portafolio seleccionado (Yu *et al.*, 2012); así mismo se ha utilizado este esquema evolutivo para la optimización del portafolio en múltiples mercados de electricidad ibéricos, logrando maximizar la rentabilidad de la inversión (Faia *et al.*, 2016); se han reportado aplicaciones del GA para maximizar el valor presente neto de la ejecución del portafolio, respetando restricciones de recursos económicos y humanos en un horizonte de tiempo dado (Martínez-Vega *et al.*, 2019); y se reporta el uso metaheurísticas Cukoo Search (CS) para la selección de portafolio en el Mercado de Valores de Teherán, con el objetivo de maximizar el beneficio esperado (Shadkam *et al.*, 2015).

Dentro de los métodos de solución de trayectoria aplicados al PPSP es posible encontrar metaheurísticas basadas en GRASP, buscando minimizar el riesgo acumulado del portafolio para un horizonte de tiempo fijo (Mira *et al.*, 2012); Metaheurísticas de Randomized Priority Search (RPS) para la selección de portafolio de proyectos con restricciones de recursos dinámicas con ventanas de tiempo (García, 2014), algoritmos de Tabu Search (TS) para la selección y programación simultánea de proyectos buscando maximizar el beneficio total esperado de la ejecución del portafolio, respetando interdependencias y fechas de terminación para el portafolio a ejecutar en un horizonte dado (Kumar *et al.*, 2019), entre otros.

Por otra parte, encontramos metaheurísticas híbridas aplicadas al PPSP tales como: simheurísticas basadas en Variable Neighborhood Search (VNA) buscando maximizar el valor presente neto de la inversión, considerando flujos de efectivo y tasas de descuento estocásticas e incluyendo restricciones como el riesgo máximo permitido por el inversionista (Panadero *et al.*, 2018); algoritmos basados Artificial Bee Colony (ABC) y Particle Swarm Optimization (PSO) para seleccionar el portafolio de proyectos que minimiza el riesgo de inversión y maximiza el valor de retorno esperado, encontrando mejores desempeños en términos de calidad de respuesta en comparación con el obtenido por algoritmos puros de ABC, GA, SA y PSO (Chen, 2015); modelos híbridos de clustering (K-Means) y GA para la selección de portafolio de proyectos en áreas de investigación y desarrollo (Eshlaghy and Razi, 2015);

algoritmos de dos etapas basados en técnicas de clustering y metaheurísticas Firefly Algorithm (FA) para balancear el riesgo de la inversión (Faezy Razi and Shadloo, 2017); así mismo, modelos híbridos de GA y Analytic Hierarchy Process (AHP) para determinar el portafolio de proyectos que maximiza el retorno de la inversión y minimiza el riesgo (Ghayour *et al.*, 2015), entre otros.

Así mismo, es posible encontrar gran diversidad de metaheurísticas aplicadas a problemas de optimización combinatoria tales como: algoritmos de Simulated Annealing (SA), que han sido utilizados en problemas de Traveling Sales Problem (TSP) con el objetivo de minimizar la distancia total recorrida, alcanzando buenos resultados en tiempos computacionales 7.8 veces más cortos que los obtenidos por algoritmos clásicos de secuenciación (Cetin, 2018); Metaheurísticas Greedy Randomized Adaptive Search Procedure (GRASP), ampliamente utilizados en problemas como el TSP modificado que permite múltiples visitas por nodo (Interian and Ribeiro, n.d.), y otros problemas como el *Job Shop Scheduling Problem* (JSSP) de complejidad NP-Hard, buscando minimizar diferentes funciones de utilidad como el promedio de tardanzas, el tiempo de finalización del último trabajo, el tiempo medio de flujo, entre otros (Baykasoğlu and Karaslan, 2017).

En el caso de procedimientos de solución metaheurísticos basados en población, encontramos diversos algoritmos aplicados a problemas de optimización combinatoria tales como: Genetic Algorithm (GA) utilizados en el TSP (Osaba *et al.*, 2014), casos de *Inventory Routing Problems* (IRPs) con múltiples locaciones y productos perecederos en los que se busca minimizar el costo de gestión total del inventario (Hiassat *et al.*, 2017); Memetic Algorithm (MA) con similitudes estructurales a los GA, utilizados de igual manera en la solución de *Flow Shop Scheduling Problems* (FSSP) (Deng and Wang, 2017); algoritmos ABC para el *Set Covering Problem* (SCP) de complejidad NP-Complete, con el que se han conseguido mejores desempeños que los reportados en diversas investigaciones para algoritmos de ACO (Crawford *et al.*, 2014), entre otros.

A partir de lo anterior, se evaluó el uso de las metaheurísticas GA y GRASP para la selección óptima del portafolio de proyectos en CMA auditados por INVIAS en Colombia. Los parámetros de las metaheurísticas propuestas fueron meta-optimizados utilizando un algoritmo de optimización bayesiana; Cada metaheurística fue evaluada para 8 CMA distintos y los resultados de desempeño se muestran en los apartados de resultados y discusión.

2. Modelo matemático de evaluación del postulante

La valoración cuantitativa de la experiencia de cada proponente establecida por INVIAS, tiene en cuenta dos dimensiones globales: la acreditación de la experiencia del postulante a través del promedio de tiempo de ejecución de los contratos presentados en el portafolio (Puntaje T) y la acreditación de la experiencia del concursante a través del promedio de facturación mensual total computada para los contratos que conforman el portafolio presentado (Puntaje F). El orden de elegibilidad de los proponentes se define a través de la puntuación total (Puntaje T) del portafolio de proyectos que el concursante presenta y que como máximo, puede alcanzar un valor total de 1,000 a través de distintos criterios tal y como se muestra en la **Tabla 1** (INVIAS, 2018).

Cuando el CMA establece incentivar la vinculación de trabajadores con discapacidad, se otorga 10 puntos al postulante que cumpla con esta condición (Puntaje I), de modo tal que para estos casos, la máxima puntuación por acreditación de tiempo de experiencia será de 295 y la acreditación por promedio de facturación mensual alcanzará como máximo los 595 puntos; si este incentivo por vinculación de personal con discapacidad no es establecido por el pliego, los puntajes de acreditación de estos dos factores como máximo podrán ser 300 y 600 respectivamente, de modo tal que el

puntaje máximo alcanzado por un postulante nunca excederá el valor de 1,000 como se detalla en la **Tabla 1**.

TABLA 1. CRITERIOS DE CALIFICACIÓN DE LA EXPERIENCIA DEL PROPONENTE	
Criterio	Puntaje máximo otorgable
Puntaje P: Acreditación del promedio de tiempo de experiencia.	295 ó 300
Puntaje F: Promedio de facturación mensual Total.	595 ó 600
Puntaje N: Apoyo a la Industria Nacional.	100
Puntaje I: Vinculación de trabajadores con discapacidad en planta de personal.	0 ó 10
<i>Puntaje T: Puntaje Total del portafolio</i>	<i>1,000</i>

Valoración de la experiencia del proponente a través de acreditación del promedio de tiempo de experiencia

La acreditación del promedio de tiempo de experiencia del portafolio de proyectos presentado por el participante es calculada tomando como referencia la siguiente expresión (INVIAS, 2018):

$$Puntaje P = \begin{cases} Pmaxp * \frac{Ppp}{Pph} & Si Ppp < Pph \\ Pmaxp * \frac{(3*Pph) - Ppp}{2*Pph} & Si Ppp \geq Pph \\ 0 & Si Ppp \leq 0.5 * Po \\ 0 & Si Ppp \geq 3 * Po \end{cases} \quad (2.1)$$

Donde:

- Puntaje P = Puntaje acreditación del promedio de tiempo de experiencia.
- Pmaxp = Puntaje máximo a otorgar por criterio de plazo (295 ó 300 dependiendo del CMA).
- Ppp = Plazo promedio de los contratos acreditados como válidos por proponente en meses (redondeado a la milésima), calculado de acuerdo con la siguiente expresión:

$$Ppp = (\sum_{j=1}^N P_j) / N \quad (2.1a)$$

Donde:

- Pj = Plazo del contrato j-ésimo en el portafolio presentado por el postulante (en meses).
- N = Número de contratos acreditados como válidos.
- Pph = Promedio de plazos acreditados a los proponentes con habilidad general (redondeado a la milésima), que se calculará de acuerdo con la siguiente expresión:

$$Pph = \frac{\sum_{n=1}^M (Ppp_n) + (Prop_{mod} * Po)}{M + Prop_{mod}} \quad (2.1b)$$

Donde:

- M = Número total de propuestas con habilidad general.
- Ppp = Plazo promedio de los contratos acreditados como válidos por proponente en meses (redondeado a la milésima).
- Prop_{mod} = Número de propuestas presentadas por módulo.
- Po = Plazo oficial del módulo en meses.

Valoración de la experiencia del proponente a través del promedio de facturación mensual total

La acreditación del promedio de facturación mensual total de cada postulante se calcula teniendo en cuenta las siguientes condiciones (INVIAS, 2018):

$$Puntaje F = \begin{cases} Pmaxf - \left(\frac{Pmaxf}{(perc * PFMO)^2}\right) * (PFMT - (perc * PFMO))^2 & \text{Si } PFMT \leq perc * PFMO \\ Pmaxf - \left(\frac{1}{Vprop_{mod}}\right) * \left(\frac{Pmaxf}{(perc * PFMO)^2}\right) * (PFMT - (perc * PFMO))^2 & \text{Si } PFMT > perc * PFMO \\ 0 & \text{Si } \frac{PFMT}{PFMO} \geq Relacion_{po} \end{cases}$$

Donde:

- Puntaje F = Puntaje por acreditación del promedio de facturación.
- Pmaxf = Puntaje máximo a otorgar por criterio de facturación (595 ó 600 dependiendo del CMA).
- PFMT = Promedio de facturación mensual total del proponente (redondeado a la milésima), calculado de acuerdo con la siguiente expresión.

$$PFMT = \frac{\sum_{j=1}^N VFA_j}{\sum_{j=1}^N n_j} \quad (2.2a)$$

Donde:

- VFA_j = Valor de facturación actualizado del contrato j-ésimo en el portafolio presentado por el postulante.
- n_j = Plazo de ejecución del contrato j-ésimo en el portafolio presentado por el postulante.

PFMO = Promedio mensual de facturación oficial, calculado a partir de la siguiente relación.

$$PFMO = \frac{PO}{Po} \quad (2.2b)$$

Donde:

- PO = Presupuesto oficial del módulo en Salarios Mínimos Mensuales Legales Vigentes (SMMLV).
- Po = Plazo oficial establecido en el pliego de condiciones.

perc = Porcentaje aplicado al PFMO, que se establece de acuerdo con las centésimas de la Tasa de cambio Representativa del Mercado (TRM) que rijan a los 2 días hábiles siguientes a la fecha de cierre del proceso, de acuerdo con lo estipulado en la siguiente tabla.

TABLA 2. PORCENTAJES PFMO CON BASE EN RANGOS TRM

Rango de variación TRM	Porcentaje del promedio mensual de facturación a aplicar
0.00-0.24	45%
0.25-0.49	50%
0.50-0.74	55%
0.75-0.99	60%

$VProp_{mod}$ = Valor por número de propuestas presentadas por módulo. Se determina de acuerdo con la siguiente tabla:

TABLA 3. VALORES DE PROPUESTAS POR LOTE

No. Propuestas presentadas por módulo	$VProp_{mod}$
1-10	2
11-20	3
21-30	4
31-40	5
41-50	6
51-60	7
61-70	8
71-80	9
≥ 81	10

$Relacion_{po}$ = Máximo número de veces del PFMT permitido, calculada de acuerdo con la siguiente expresión.

$$Relacion_{po} = perc * (\sqrt{Vprop_{mod}} + 1) \quad (2.2c)$$

Donde:

- $perc$ = Porcentaje aplicado al PFMO, que se establece de acuerdo con las centésimas de la Tasa de cambio Representativa del Mercado (TRM) que rija a los 2 días hábiles siguientes a la fecha de cierre del proceso.
- $VProp_{mod}$ = Valor por número de propuestas presentadas por módulo.

De este modo el Puntaje T a maximizar se define como la sumatoria de los puntajes individuales definidos anteriormente.

3. Materiales y métodos

Los datos requeridos para la construcción y evaluación de los métodos de solución propuestos fueron suministrados por una *Compañía* con más de 20 años de trayectoria en interventoría, diseño y construcción de obras públicas en Colombia. El portafolio histórico de proyectos históricos ejecutados por la *Compañía* corresponde a información interna de uso confidencial que asciende a 207 proyectos. Adicionalmente, se recolectaron ocho CMA's distintos, con sus respectivos parámetros de valoración y resultados alcanzados por los postulantes, disponibles en el Sistema Electrónico para la Contratación Pública (SECOP), para concursos de méritos supervisados por INVIAS.

Por otra parte se programaron las metaheurísticas GA y GRASP para la selección del portafolio de proyectos en el contexto de estudio. Estos métodos de solución

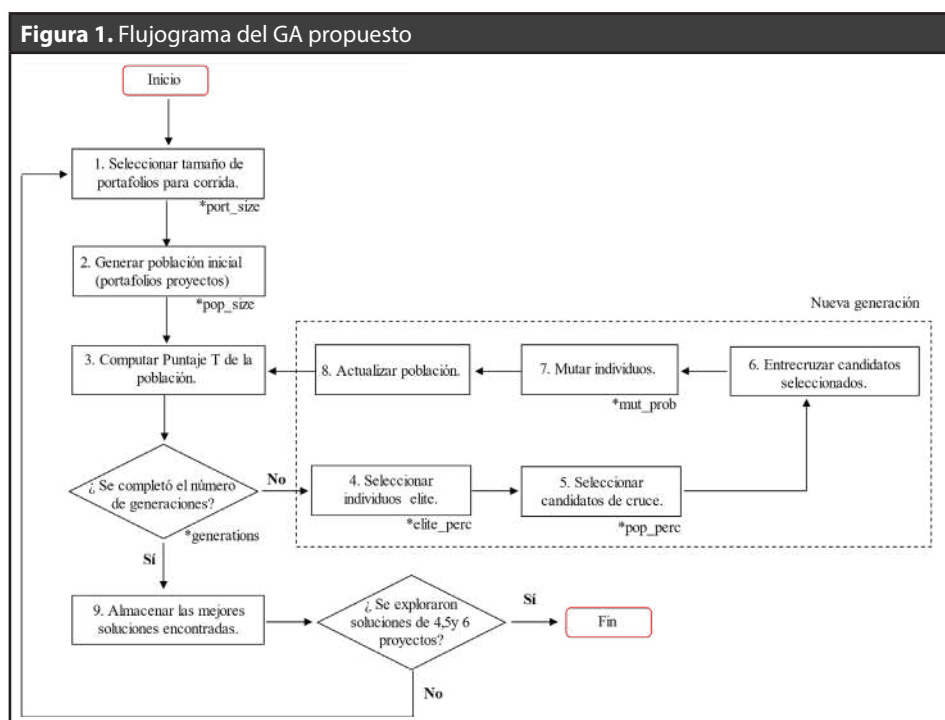
metaheurísticos fueron meta-optimizados utilizando un algoritmo de optimización bayesiana y el criterio de negocio de la Organización. Así mismo, se realizó una evaluación comparativa en términos de calidad de respuesta y tiempos computacionales para distintos pliegos de condiciones.

Todos los cálculos computacionales realizados durante el estudio, fueron ejecutados en la estación de trabajo portátil ASUS N551JX, que cuenta con un procesador Intel® Core™ i7-4720HQ @ 2.60 GHz, 8 GB de memoria RAM DDR3 y tarjeta de vídeo integrada NVIDIA® GeForce® GTX 950M de 4GB DDR3. Se utilizó el lenguaje de programación Python 3.6.8 para la parametrización y la evaluación de los algoritmos de solución programados, así mismo, se trabajó con Visual Basic for Applications (VBA) embebido en Excel, para el despliegue de los algoritmos a través de una interfaz de usuario parametrizable.

Diseño del GA propuesto

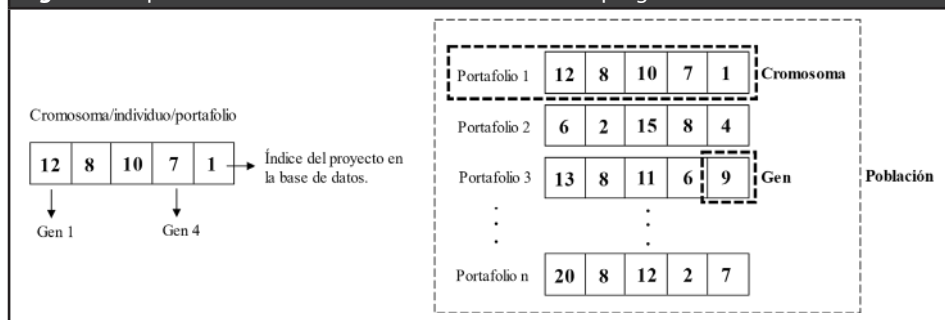
Los GA se ubican dentro de la familia de metaheurísticas de población basadas en analogías con la evolución biológica, que contempla las siguientes etapas básicas: selección natural, reproducción, mutación y supervivencia de los individuos más aptos. El desempeño de los GA no está condicionado de manera estricta por un conocimiento profundo del problema de estudio, de modo tal que representan mayor probabilidad de alcanzar óptimos globales en la mayoría de las aplicaciones (Yu *et al.*, 2012).

Para el caso de estudio, se programó un GA utilizando Python 3.6.8. El flujograma de la metaheurística se detalla en la **Figura 1**.



El algoritmo inicia con la creación de una población aleatoria de individuos/portafolios, cada uno de estos últimos conformado por port_size proyectos. En la **Figura 2** se detalla la estructura de un cromosoma compuesto por cinco genes (port_size = 5); cada gen es representado por un número entero que corresponde al índice del proyecto en la base de datos histórica de experiencias ejecutadas por la Organización.

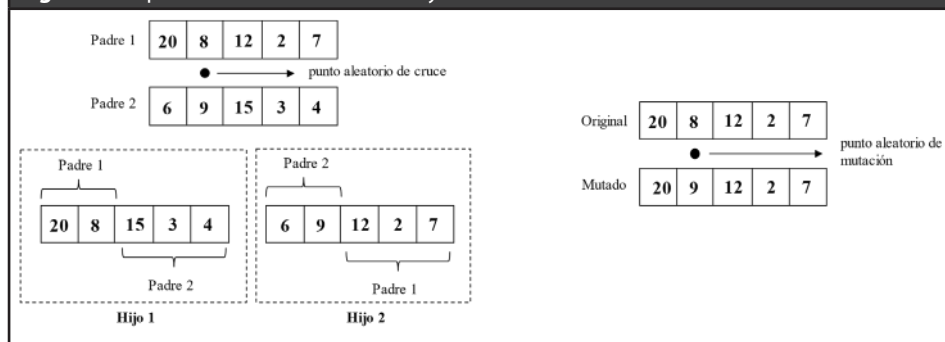
Figura 2. Esquema de codificación de soluciones en el GA programado



Una vez creada la población inicial, cada individuo es evaluado y rankeado de acuerdo con la función objetivo del problema, el Puntaje T. El objetivo tras cada iteración es evolucionar la población anterior mejorando los individuos/portafolios que la conforman. En primer lugar, una proporción de individuos élite (*elite_perc*) pasa automáticamente a la siguiente generación por su alto Puntaje T; mientras que la proporción restante de portafolios requeridos para mantener el tamaño de la población inicial (*pop_size*) es construida durante las etapas de entrecruzamiento y mutación de los individuos más aptos (*pop_perc*).

El entrecruzamiento consiste en seleccionar un punto de corte aleatorio del cromosoma, para combinar el material genético de ambos padres y dar lugar a los hijos que completarán el tamaño requerido de la población tal y como se detalla en la **Figura 3**. Por otra parte, cada individuo resultante podrá ser mutado con una probabilidad (*mut_prob*) para permitir al GA escapar de óptimos locales.

Figura 3. Esquema de entrecruzamiento y mutación de individuos del GA



Esta mutación consiste en seleccionar un gen aleatorio del cromosoma del individuo, para cambiar este por el índice de un proyecto que no esté presente dentro del individuo a ser mutado tal y como se detalla en la **Figura 3**.

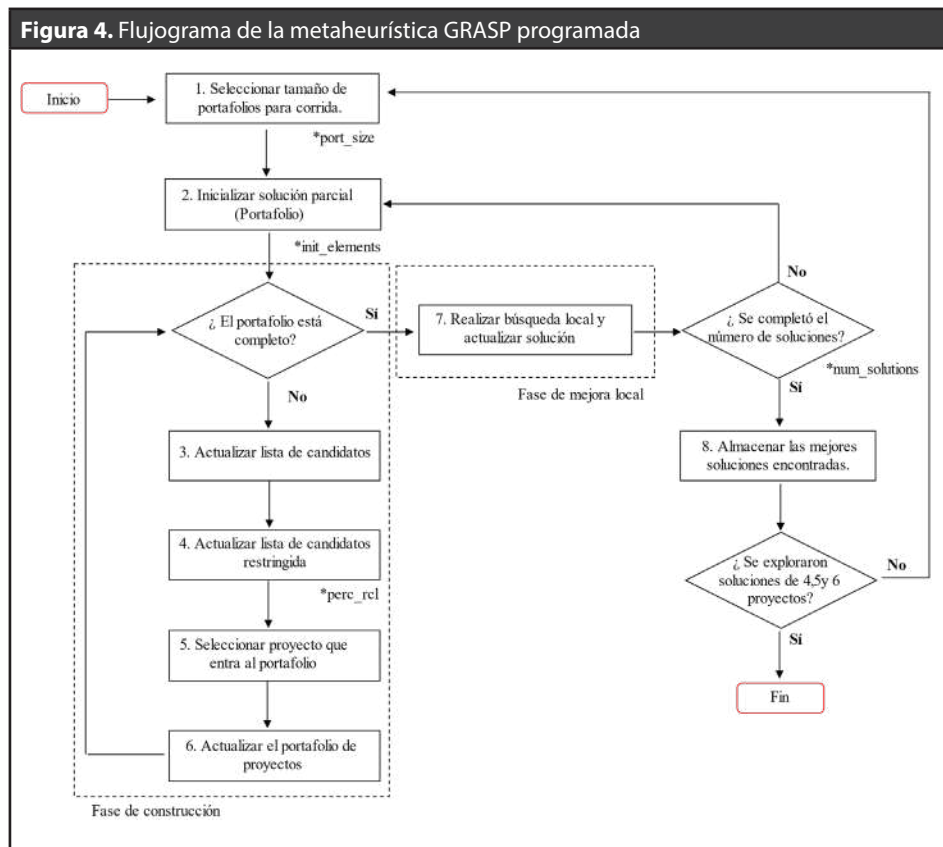
Como se resaltó en la **Figura 1**, el procedimiento de elitismo-entrecruzamiento-mutación, se repite hasta alcanzar el número de iteraciones deseado (*generations*). Una vez se alcanza este se almacenan los mejores 10 individuos únicos resultantes luego de múltiples generaciones de evolución en función de su Puntaje T.

Diseño del GRASP propuesto

La metaheurística GRASP es un algoritmo de trayectoria que se basa en la combinación de procedimientos de búsqueda avariciosos y mejoras locales a través de búsquedas de vecindario iterativas. El componente avaricioso de la metaheurística mide el beneficio de añadir cada uno de los elementos en pro de elegir lo más conveniente

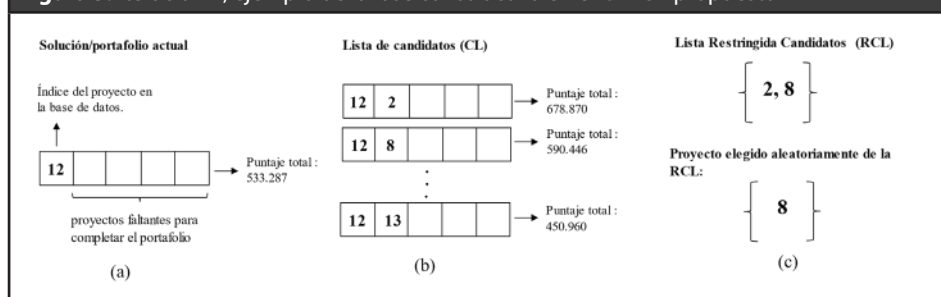
en cada iteración, mientras que la búsqueda de vecindario permite una mejora de la solución alcanzada en la etapa constructiva a través de la exploración de movimientos en el vecindario de soluciones (Resende and Ribeiro, 2016).

Para el problema de selección de portafolio de proyectos se implementó la metaheurística GRASP en Python 3.6.8, siguiendo el flujograma de operaciones detallado en la **Figura 4**.



La solución inicia con un número de proyectos definido a través del parámetro `init_elements`. Estos proyectos insertados en la solución inicial, son tomados de la bolsa de contratos históricos ejecutados. En la **Ilustración 6 a**, puede apreciarse el ejemplo de lo anterior, para una solución que inicia con `init_elements = 1`, de modo tal que, en la fase constructiva se deberán incluir cuatro (4) proyectos restantes para completar el portafolio de cinco (5) contratos para un `port_size = 5` en dicha iteración.

Una vez se ha inicializado una solución, se actualiza la lista de candidatos (CL) para la iteración actual. Esta lista se encuentra conformada por la lista de proyectos que aún no se encuentran dentro de la solución y que representan una alternativa potencial de inclusión al portafolio para la siguiente iteración. Los elementos de la lista de candidatos están organizados de mayor a menor por valor de Puntaje T. Siguiendo el ejemplo de la **Figura 5**, el esquema b muestra un ranking de los proyectos teniendo en cuenta su aporte a la función fitness, dado que ya se encuentra en la solución el proyecto con índice 12; de esta manera, la mejor opción sería incluir el proyecto 2 para que la función fitness alcance un valor de 678.870, mientras que incluir el proyecto 13 incluso decrementaría el valor de la función fitness al pasar de 583.287 (únicamente con el proyecto 12) a 450.960 (con los proyectos 12 y 13).

Figura 5. Iteración 1, ejemplo de la fase constructiva en el GRASP propuesto

Una vez se ha actualizado la lista CL, se construye la lista restringida de candidatos (RCL), seleccionando una proporción de elementos en la lista de candidatos (perc_rcl) del top de la lista de candidatos que se fue rankeada previamente. Para el caso de ejemplo, la cantidad de elementos tomados de la lista de candidatos para conformar la lista restringida de candidatos es equivalente a 2. Finalmente, el proyecto que entra al portafolio en la iteración actual, es seleccionado al azar de los elementos que conforman la lista restringida de candidatos en la iteración actual, que para el caso de ejemplo equivale al proyecto con el índice 8. Este procedimiento se repite iterativamente hasta completar el número de proyectos requeridos en el portafolio, dado por el valor que aplique para el parámetro port_size para la iteración actual.

Una vez ha terminada la fase constructiva, se da inicio a una fase de mejora local, en la que el portafolio resultante de la fase constructiva es mejorado tomando un elemento de la lista de candidatos y revisando uno a uno su posible reemplazo por alguno de los proyectos incluidos en el portafolio resultante de la fase constructiva. Todo el proceso se repite hasta haber construido y mejorado la cantidad de soluciones definidas por el número de iteraciones (num_solutions). Al final, se ordenan de mayor a menor puntaje las soluciones construidas y se conservan los 10 mejores portafolios encontrados.

Meta-optimización de las metaheurísticas GA y GRASP

La parametrización de metaheurísticas condiciona en gran medida su rendimiento frente a un problema particular de estudio. A medida que crece el número de parámetros que a su vez pueden tomar un amplio rango de valores, el problema de parametrización se complejiza debido a las relaciones subyacentes propias de la combinación de parámetros, que no son identificables fácilmente a través de la experiencia del investigador y mucho menos generalizables para un amplio número de problemas de distinta naturaleza (Pedersen, 2010).

Teniendo en cuenta lo anterior, para la selección de parámetros del GA y el algoritmo GRASP propuesto, se utilizó el enfoque de búsqueda y selección de parámetros para algoritmos de optimización conocido como meta-optimización. Este esquema se centra en encontrar la parametrización óptima de un algoritmo de optimización que se aplica para solucionar un problema de base como en el caso de estudio (Neumüller *et al.*, 2012).

En esta etapa de meta-optimización se utilizó un algoritmo de optimización bayesiano desarrollado en el Massachusetts Institute of Technology (Pedersen, 2010). Algunas de las aplicaciones recientes de esta librería de gran reconocimiento se encuentran en áreas de planificación de inventarios multi escalón utilizando simulación-optimización (Agarwal, 2018); pronósticos de sistemas caóticos (Griffith *et al.*, 2019); parametrización de heurísticas de optimización para algoritmos de machine learning (Boryssenko and Herscovici, 2018), entre otros.

Para la evaluación de cada conjunto de parámetros en el caso de estudio, se formuló la siguiente función de desempeño *meta_fitness* a minimizar:

$$meta_fitness = \frac{\sum_{i=1}^5 \sum_{j=1}^{10} ((1,000 - Puntaje T_j) * 100)^2 * conv}{5} \quad (4.1)$$

Donde:

- *meta_fitness* = Función de desempeño del optimizador bayesiano a minimizar.
- *i* = réplica *i-ésima* de la metaheurística para un conjunto de parámetros dado.
- *j* = Portafolio *j-ésimo* con mejor puntaje encontrado por la metaheurística.
- Puntaje *T* = Puntaje total del portafolio *j-ésimo* en el ranking con los 10 mejores portafolios encontrados por la metaheurística bajo una parametrización dada.
- *conv* = factor de penalización por convergencia. Si el algoritmo termina con menos de 10 soluciones factibles no repetidas, se penaliza la función *meta_fitness* multiplicando por 100 cada una de las distancias evaluadas.

De esta manera, la función de evaluación *meta_fitness* a minimizar, no sólo tiene en cuenta el mejor portafolio encontrado (portafolio con la menor diferencia entre el puntaje máximo que se puede obtener (1,000) y su valor alcanzado Puntaje *T_j*), sino que además, contempla la efectividad del método de solución para hallar hasta 10 portafolios candidatos de ser escogidos por su ranking con base en el criterio de Negocio.

Para el GA, se definieron los siguientes rangos para cada parámetro en función del estado del arte, pruebas de experimentación preliminares y la experiencia del equipo:

pop_size = [1000, 35000]

mut_prob = [0.0, 0.8]

pop_perc = [0.1, 1.0]

elite_perc = [0.0, 1.0]

generations = [10,80]

Por otra parte, para el algoritmo GRASP se definieron los siguientes rangos de búsqueda de parametrización:

perc_rcl = [0.01, 0.5]

init_elements = [0, 4]

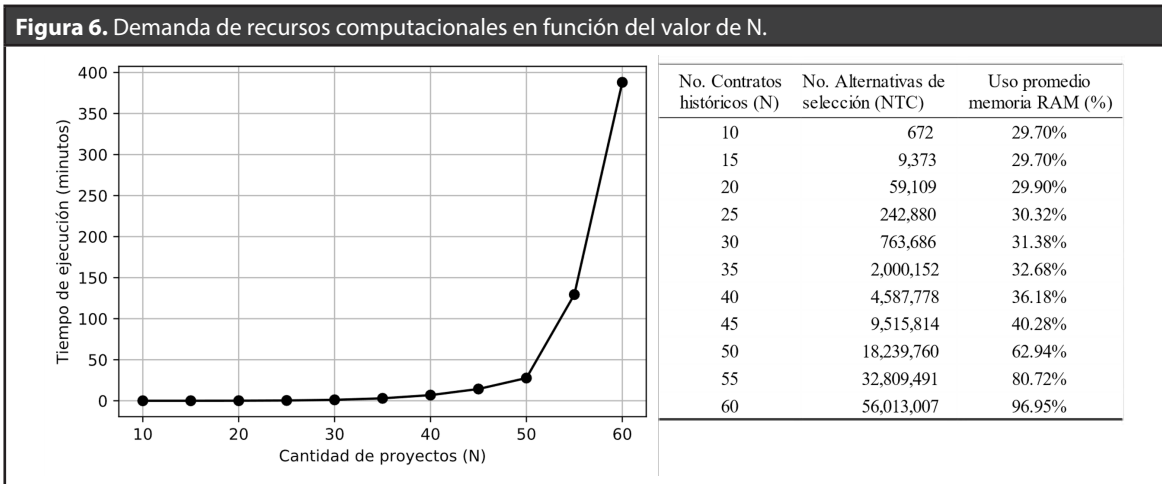
num_solutions = [100, 3500]

Cada conjunto de parámetros evaluado por el algoritmo Bayesiano, se computó para cinco (5) réplicas distintas y se computó su valor promedio. Los resultados obtenidos en la etapa de meta-optimización fueron validados utilizando 2 CMA adicionales.

4. Resultados

Complejidad computacional del problema

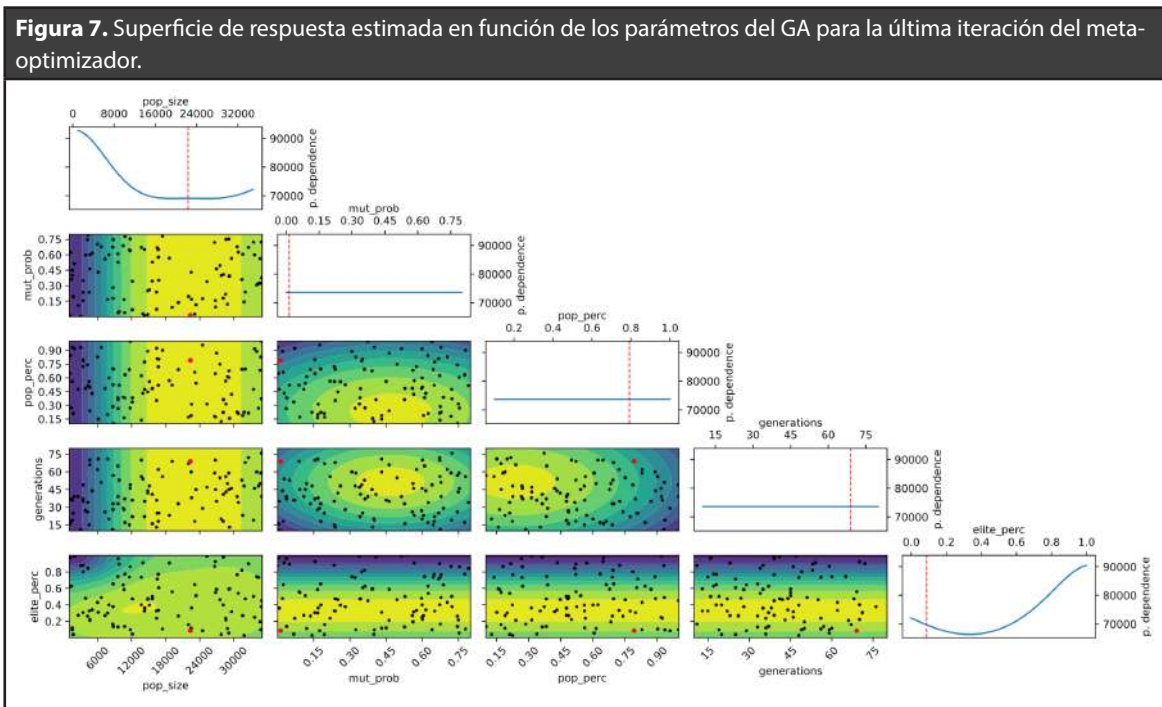
Para profundizar en la complejidad computacional del problema e identificar algunas características propias del espacio de solución de este, se realizaron distintas corridas experimentales utilizando un algoritmo de búsqueda exhaustiva, variando el número de contratos históricos (*N*). La demanda de recursos computacionales en función del NTC se presenta en la **Figura 6**.



Parámetros seleccionados para la metaheurística GA

La estimación de la función meta-fitness en función de los parámetros pop_size, mut_prob, pop_perc, generations y elite_perc para la iteración 100, se detalla en la **Figura 7**.

Las regiones amarillas representan los mejores valores encontrados para la función meta-fitness, mientras que las regiones azules representan los valores más altos en el problema de minimización. Así mismo, la diagonal del gráfico muestra la dependencia parcial encontrada por el optimizador bayessiano entre el valor de la función meta-fitness y los valores específicos que pueden tomar los distintos parámetros.



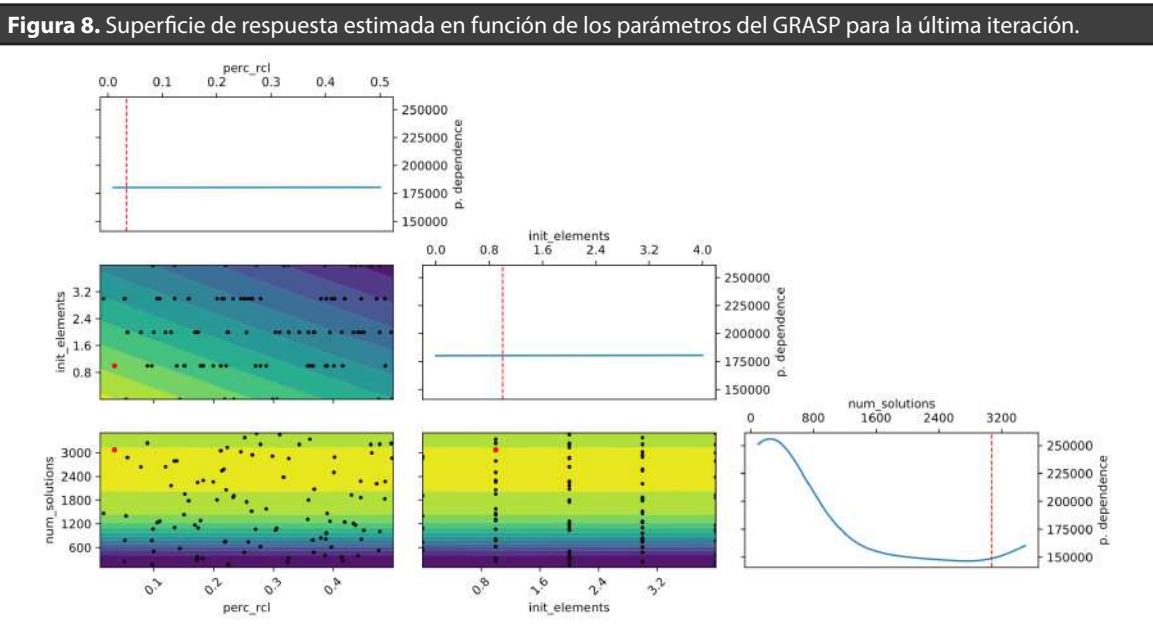
Los resultados de validación para las 5 mejores parametrizaciones encontradas a través del optimizador Bayesiano se detallan en la **Tabla 4**.

TABLA 4. RANKING DE LAS 5 MEJORES PARAMETRIZACIONES ENCONTRADAS PARA LA METAHEURÍSTICA GA

pop_size	mut_prob	pop_perc	generations	elite_perc	meta_fitness	Desv. estándar	Tiempo (min)	Ranking
22,349	0.012	0.792	69	0.087	7.249	0.106	8.078	1.000
14,299	0.008	0.476	35	0.361	10.997	0.302	2.623	3.667
12,128	0.067	0.337	36	0.046	43.204	26.099	2.295	4.667
27,790	0.399	0.126	20	0.042	12.121	0.396	3.157	4.667
28,899	0.121	0.615	50	0.113	13.217	0.450	7.689	5.000

Parámetros seleccionados para le metaheurística GRASP

Siguiendo la metodología de búsqueda de parámetros expuesta anteriormente para la metaheurística GA, en la **Figura 8** se detalla el espacio de solución para el problema de minimización de la función meta-fitness para la metaheurística GRASP, en la iteración número 100 del meta-optimizador.



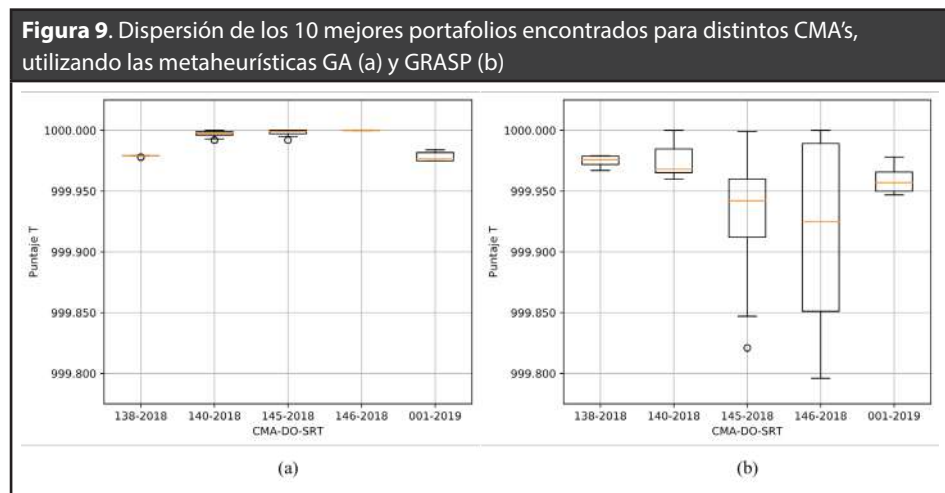
Partiendo de esta parametrización, se realizó una validación de las 5 mejores parametrizaciones encontradas en la etapa de meta-optimización, utilizando 2 pliegos de condiciones adicionales. Los resultados obtenidos se detallan en la **Tabla 5**.

TABLA 5. RANKING DE LAS 5 MEJORES PARAMETRIZACIONES ENCONTRADAS PARA LA METAHEURÍSTICA GRASP

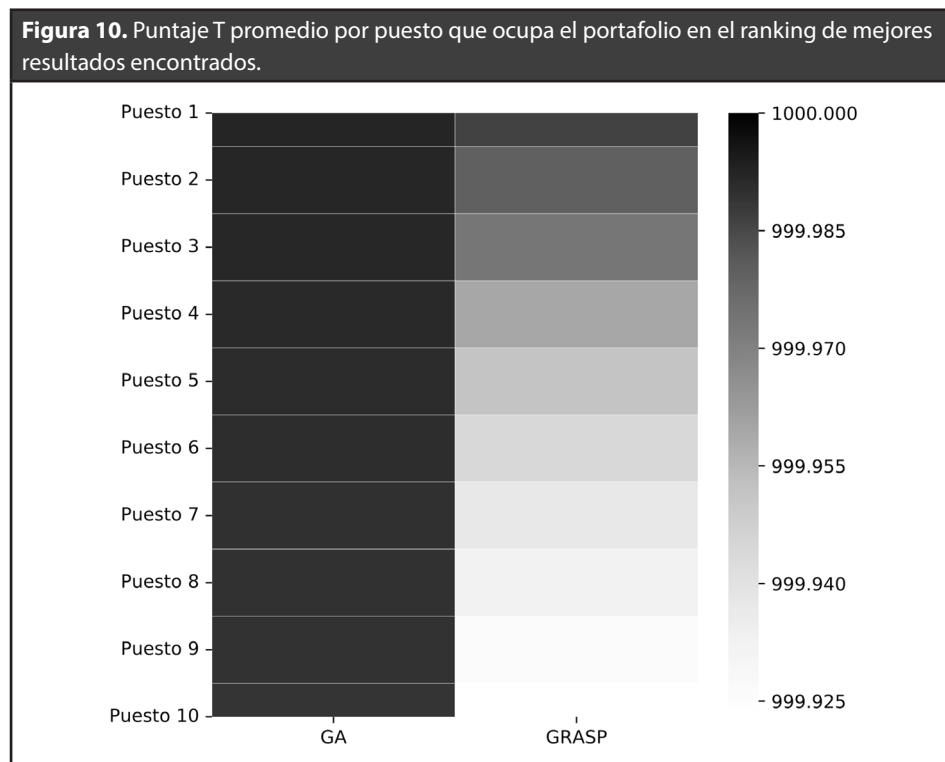
perc_rcl	init_elements	num_solutions	meta_fitness promedio	Desv. estándar promedio	Tiempo promedio (min)	Ranking promedio
0.034	1	3,076	288.604	236.593	9.672	1.000
0.016	2	1,467	463.022	457.238	2.678	3.667
0.056	2	2,878	613.946	428.480	7.310	3.333
0.054	0	1,400	826.848	431.078	5.076	3.667
0.089	1	3,241	874.215	437.745	10.144	4.000

Desempeño de las metaheurísticas GA y GRPAS

Para evaluar el desempeño de las metaheurísticas GA y GRASP se computaron los valores de Puntaje T promedio para los 10 mejores portafolios alcanzados, variando el pliego de condiciones. En la **Figura 9**, se muestra un diagrama de variabilidad en los puntajes alcanzados por ambas metaheurísticas.



Por otra parte, se analizó en detalle el desempeño alcanzado por cada método de solución en función del Puntaje T alcanzado por los portafolios en el ranking 10 de las mejores soluciones. De esta manera se evaluó la capacidad de los algoritmos para encontrar consistentemente hasta 10 portafolios de proyectos con Puntajes T muy cercanos al óptimo global del problema. Los resultados comparativos con relación al puesto alcanzado por el portafolio se detallan en la **Figura 10**.



5. Discusión

En la **Figura 6**, se observa que para valores de N superiores a 45 contratos históricos, el consumo de recursos computacionales marcó un crecimiento acelerado que terminó por consumir la totalidad de recursos del equipo cuando N alcanzó un valor de 60, con un tamaño del espacio de solución equivalente a cerca de 56 millones de alternativas de solución que terminaron por emplear más del 96% de la memoria RAM y demandaron un tiempo promedio de cómputo de alrededor de 7 horas. Esto ratifica que el uso de esquemas de enumeración exhaustiva para empresas con más de 45 proyectos de experiencia no es una alternativa viable, aún menos si se utilizan formulaciones estáticas en hojas de Excel.

Por otra parte, la mejor parametrización encontrada en la etapa de meta-optimización del GA, arrojó un valor `meta_fitness` promedio de 21.748, para un tiempo equivalente de ejecución 8.102 minutos. Los parámetros asociados a esta parametrización son: `pop_size` =22,349, `mut_prob` =0.012, `pop_perc` =0.792, `generations` =69 y `elite_perc` =0.087. Así mismo, se encontró que el parámetro de tamaño de población (`pop_size`), deteriora notable de la función `meta_fitness` para valores por debajo de 9,000 individuos; mientras que en el caso del parámetro de proporción de elitismo (`elite_perc`), valores mayores al 0.6 (60%), afectan el valor de la función objetivo tal y como se observa en la **Figura 7**. Durante la validación de parámetros para el GA, se encontró que la mejor parametrización encontrada en la etapa de meta-optimización, se mantuvo en el primer lugar en las instancias de validación, ratificando la robustez de la parametrización alcanzada por el meta-optimizador.

Para el caso de meta-optimización del algoritmo GRASP, no se encontraron dependencias parciales claras asociadas a los parámetros `perc_rcl` e `init_elements`, mientras que valores mayores a 2 para el parámetro `init_elements`, en conjunto con valores de `perc_rcl` por arriba de 0.1 (10%), causaban un deterioro notable de la función `meta_fitness` aumentando su valor. Esto indica particularmente que a medida que aumenta el número de proyectos introducidos aleatoriamente en la inicialización de la solución, el Puntaje T de los 10 mejores portafolios se deteriora consistentemente al perder el componente constructivo de la metaheurística. Los mejores valores encontrados para el algoritmo GRASP (`perc_rcl` =0.034, `init_elements` =1 y `num_solutions` =3,076), se mantuvieron en el primer lugar durante las validaciones realizadas al igual que en el caso de la metaheurística GA.

Los resultados de la **Figura 9**, muestran las diferencias entre los valores de Puntaje T para los 10 mejores portafolios encontrados por las metaheurísticas GA y GRASP. En esta ilustración es posible identificar que el algoritmo GA presentó una baja dispersión para los cinco CMA's de pruebas finales, resaltando que para el CMA-DO-SRT-146-2018, la metaheurística GA consiguió encontrar 10 portafolios con valor de 1,000 en las cinco réplicas realizadas, motivo por el cual su desviación en dicha instancia es nula. Por otra parte, la metaheurística GRASP presentó una mayor dispersión en todas las instancias de evaluación, debido a las diferencias entre el Puntaje T del mejor portafolio encontrado (portafolio en el puesto 1) y aquellos que proceden a este último en el ranking de las 10 mejores soluciones.

Lo anterior se muestra con mayor detalle en la **Figura 10**. En esta ilustración, el valor promedio de los portafolios que ocupan la primera y la segunda posición para el algoritmo GRASP presentaron un Puntaje T promedio cercano a 999.985, que decrece a medida que se analizan los portafolios ubicados en los puestos del 4 al 10. Por otra parte, para el algoritmo AG, los valores de Puntaje T promedio se mantienen cercanos al valor de 1,000 para todos los puestos en el ranking de mejores soluciones, confirmando la ventaja del AG para encontrar hasta 10 portafolios de proyectos con puntajes superiores a 999.970, superando en todos los casos de evaluación, los puntajes reales

alcanzados por las empresas ganadoras de las distintas convocatorias, cuyos portafolios presentaron valores de Puntaje T que oscilaron entre 999.368 y 999.970.

Esta diferencia en el desempeño del algoritmo GRASP podría reducirse abarcando una mayor parte del espacio de solución dentro de los tiempos permisibles por el negocio, mejorando su desempeño global si se contempla una evaluación parcial aleatoria de la lista CL en vez de su totalidad. Así mismo, podrían explorarse otras estrategias de mejora local que mejoren la calidad de respuestas del algoritmo y en especial su habilidad para encontrar hasta 10 portafolios con valores cercanos al Puntaje T óptimo.

Partiendo de los resultados anteriores, se determinó que el método de solución que más se adecua a los requerimientos de negocio es el algoritmo GA. Este algoritmo permitirá a la *Organización*, mejorar sus procesos de licitación a CMA's auditados por INVIAS. La herramienta desarrollada no sólo reduce los tiempos y la carga computacional asociada al uso de hojas de cálculo de Excel, sino que además aumenta la probabilidad de adjudicación de interventorías a través de la presentación de portafolios de proyectos con mayor puntaje. Lo anterior, se encuentra sujeto a la calidad en la estimación de variables aleatorias que intervienen en la función matemático de valoración como en el caso de la tasa que aplicará al concurso y la cantidad de postulantes que se presentarán al mismo, dado que es información que sirve de entrada al modelo de selección de portafolio de proyectos y que a la fecha, su estimación se encuentra a cargo del Departamento de Licitaciones de la *Compañía*.

Referencias

- Agarwal, A., 2018. Multi-echelon Supply Chain Inventory Planning using Simulation-Optimization with Data Resampling. arXiv:1901.00090 [math].
- Baykasoğlu, A., Karaslan, F.S., 2017. Solving comprehensive dynamic job shop scheduling problem by using a GRASP-based approach. *International Journal of Production Research* 55, 3308–3325. <https://doi.org/10.1080/00207543.2017.1306134>
- Boryszenko, A., Herscovici, N., 2018. Machine Learning for Multiobjective Evolutionary Optimization in Python for EM Problems, in: 2018 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting. Presented at the 2018 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting, pp. 541–542. <https://doi.org/10.1109/APUSNCURSINRSM.2018.8609394>
- Cetin, O., 2018. Parallelizing simulated annealing algorithm for TSP on massively parallel architectures. *Journal of Aeronautics and Space Technologies* 11, 75–85.
- Chen, W., 2015. Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem. *Physica A: Statistical Mechanics and its Applications* 429, 125–139. <https://doi.org/10.1016/j.physa.2015.02.060>
- Colombia Compra Eficiente, 2017. Guía para procesos de contratación de obra pública.
- Crawford, B., Soto, R., Cuesta, R., Paredes, F., 2014. Application of the Artificial Bee Colony Algorithm for Solving the Set Covering Problem [WWW Document]. *The Scientific World Journal*. <https://doi.org/10.1155/2014/189164>
- Deng, J., Wang, L., 2017. A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. *Swarm and Evolutionary Computation* 32, 121–131. <https://doi.org/10.1016/j.swevo.2016.06.002>
- Eshlaghy, A.T., Razi, F.F., 2015. A hybrid grey-based k-means and genetic algorithm for project selection. *International Journal of Business Information Systems* 18, 141–159. <https://doi.org/10.1504/IJBIS.2015.067262>
- Faezy Razi, F., Shadloo, N., 2017. A Hybrid Grey based Two Steps Clustering and Firefly Algorithm for Portfolio Selection. *Journal of Optimization in Industrial Engineering* 10, 49–59. <https://doi.org/10.22094/joie.2017.276>
- Faia, R., Pinto, T., Vale, Z., 2016. GA optimization technique for portfolio optimization of electricity market participation, in: 2016 IEEE Symposium Series on Computational

- Intelligence (SSCI). Presented at the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, Athens, Greece, pp. 1–7. <https://doi.org/10.1109/SSCI.2016.7849858>
- García, C., 2014. A metaheuristic algorithm for project selection and scheduling with due windows and limited inventory capacity. *Kybernetes* 43, 1483–1499. <https://doi.org/10.1108/K-11-2013-0245>
- Ghayour, F, Solimanpur, M., Mansourfar, G., 2015. Optimum portfolio selection using a hybrid genetic algorithm and analytic hierarchy process. *Studies in Economics & Finance* 32, 379–394. <https://doi.org/10.1108/SEF-08-2012-0085>
- Griffith, A., Pomerance, A., Gauthier, D.J., 2019. Forecasting Chaotic Systems with Very Low Connectivity Reservoir Computers. arXiv:1910.00659 [nlin, stat].
- Hiassat, A., Diabat, A., Rahwan, I., 2017. A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of Manufacturing Systems* 42, 93–103. <https://doi.org/10.1016/j.jmsy.2016.10.004>
- Instituto Nacional de Vías, 2017. Concurso de méritos abierto CMA-DO-SRN-003-2017.
- Interian, R., Ribeiro, C.C., n.d. A GRASP heuristic using path-relinking and restarts for the Steiner traveling salesman problem. *International Transactions in Operational Research* 24, 1307–1323. <https://doi.org/10.1111/itor.12419>
- INVIAS, 2018. Concurso de méritos abierto CMA-DO-SRT-063-2018.
- Kumar, M., Mittal, M.L., Soni, G., Joshi, D., 2019. A Tabu Search Algorithm for Simultaneous Selection and Scheduling of Projects, in: Yadav, N., Yadav, A., Bansal, J.C., Deep, K., Kim, J.H. (Eds.), *Harmony Search and Nature Inspired Optimization Algorithms*, *Advances in Intelligent Systems and Computing*. Springer Singapore, pp. 1111–1121.
- Martínez-Vega, D.A., Cruz-Reyes, L., Rangel-Valdez, N., Santillán, C.G., Sánchez-Solís, P., Villafuerte, M.P., 2019. Project Portfolio Selection with Scheduling: An Evolutionary Approach. 1 10, 25–31.
- Mira, C., Feijao, P., Souza, M.A., Moura, A., Meidanis, J., Lima, G., Schmitz, R., Bossolan, R.P., Freitas, I.T., 2012. A GRASP-based Heuristic for the Project Portfolio Selection Problem, in: 2012 IEEE 15th International Conference on Computational Science and Engineering. Presented at the 2012 IEEE 15th International Conference on Computational Science and Engineering (CSE), IEEE, Paphos, Cyprus, pp. 36–41. <https://doi.org/10.1109/ICCSE.2012.102>
- Neumüller, C., Wagner, S., Kronberger, G., Affenzeller, M., 2012. Parameter Meta-optimization of Metaheuristic Optimization Algorithms, in: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (Eds.), *Computer Aided Systems Theory – EUROCAST 2011, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 367–374.
- Osaba, E., Carballedo, R., Diaz, F., Onieva, E., Lopez, P., Perallos, A., 2014. On the influence of using initialization functions on genetic algorithms solving combinatorial optimization problems: A first study on the TSP, in: 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS). Presented at the 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), IEEE, Linz, Austria, pp. 1–6. <https://doi.org/10.1109/EAIS.2014.6867465>
- Panadero, J., Doering, J., Kizys, R., Juan, A.A., Fito, A., 2018. A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics*. <https://doi.org/10.1007/s10732-018-9367-z>
- Pedersen, M.E.H., 2010. *Tuning & Simplifying Heuristical Optimization* (phd). University of Southampton.
- Resende, M.G.C., Ribeiro, C.C., 2016. *Optimization by GRASP*. Springer New York, New York, NY. <https://doi.org/10.1007/978-1-4939-6530-4>
- Shadkam, E., Delavari, R., Memariani, F., Poursaleh, M., 2015. Portfolio Selection by the Means of Cuckoo Optimization Algorithm. *International Journal on Computational Science & Applications* 5, 37–46. <https://doi.org/10.5121/ijcsa.2015.5304>
- Yu, L., Wang, S., Wen, F., Lai, K.K., 2012. Genetic algorithm-based multi-criteria project portfolio selection. *Annals of Operations Research* 197, 71–86. <https://doi.org/10.1007/s10479-010-0819-6>