

**EMPLEO DE METAHEURISTICAS EN EL DISEÑO DE ESTRUCTURAS
APORTICADAS DE CONCRETO.**

EDER JAIR DURÁN CÁCERES

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECHANICAS
ESCUELA DE INGENIERIA CIVIL
ESPECIALIZACIÓN ESTRUCTURA
2019**

**EMPLEO DE METAHEURISTICAS EN EL DISEÑO DE ESTRUCTURAS
APORTICADAS DE CONCRETO.**

EDER JAIR DURÁN CÁCERES

**TRABAJO DE GRADO PARA OPTAR AL TITULO DE ESPECIALISTA EN
ESTRUCTURAS**

DIRECTOR

**OSCAR JAVIER BEGAMBRE CARRILLO
DOCTOR EN INGENIERÍA DE ESTRUCTURAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECHANICAS
ESCUELA DE INGENIERIA CIVIL
ESPECIALIZACIÓN ESTRUCTURA**

2019

AGRADECIMIENTOS

*A Dios, a mis padres por su constante apoyo y consejo durante esta etapa
Bonita de mi vida, a mí hermano por impulsarme cada día a ser mejor
y a cada uno de mis amigos por estar siempre a mi lado apoyándome.*

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. OBJETIVOS	19
1.1. OBJETIVO GENERAL	19
1.2. OBJETIVOS ESPECIFICOS.....	19
1.3. ALCANCE	19
2. ANALISIS BIBLIOMETRICO.....	20
2.1. CRITERIOS DE BÚSQUEDA EMPLEADOS Y RESULTADOS OBTENIDOS	20
2.2. REVISTAS IDENTIFICADAS	24
2.3. ALGORITMOS EMPLEADOS CON MAYOR FRECUENCIA.....	26
2.3.1. Algoritmos genéticos (GA)	27
2.3.2. Recocido simulado (SA).....	28
2.3.3. Optimización por enjambre de partículas (PSO).....	30
2.3.4. Búsqueda de la armonía (HS).....	31
2.3.5. Optimización por colonia de hormigas (ACO)	32
2.3.6. Algoritmos Big Bang Big Crunch.....	34

2.3.7. El algoritmo búsqueda de sistema cargado CSS	35
2.3.8. Algoritmo de la luciérnaga (FA).....	36
2.3.9. Algoritmo murciélago (BA)	37
2.3.10. El algoritmo optimización de rayos RO	38
2.4. LISTA DE SOFTWARE COMERCIA	39
3. METODOS EMPLEADOS EN OPTIMIZACIÓN	42
3.1. METODOS EXACTOS.....	43
3.1.1. La programación dinámica.....	44
3.1.2. El algoritmo de ramificación y límite.....	45
3.1.3. Programación de restricciones.....	45
3.2. ALGORITMOS APROXIMADOS	45
3.3. METAHEURÍSTICA	47
4. ALGORITMOS METAHEURISTICOS Y DISEÑO DE ESTRUCTURAS APORTICAS DE CONCRETO REFORZADO	52
4.1. CLASIFICACIÓN DE LAS METAHEURÍSTICAS.....	54
4.1.1. Basadas en la naturaleza.....	54
4.1.2. Sistemas con memoria y sin memoria	56
4.1.3. Deterministas y estocásticos.....	56
4.1.4. Búsqueda basada en población y búsqueda basada en una sola solución. 57	
4.1.5. Iterativo y codicioso	57
4.2. CONCEPTOS COMUNES PARA LAS METAHEURÍSTICAS.....	57
4.2.1. Representación.....	58

4.2.2. Función objetivo	59
4.3. DESCRIPCIÓN DE LAS METAHEURÍSTICAS MÁS UTILIZADAS.	61
4.3.1. Recocido simulado (SA).....	62
4.3.2. El método de búsqueda tabú (TS)	66
4.3.3. Algoritmos genéticos (GA) y algoritmos evolutivos (EA)	70
4.3.4. Algoritmos de colonias de hormigas (ACO)	76
4.3.5. Algoritmo de optimización de enjambre de partículas (PSO).....	81
4.4. MULTIOBJETIVO EN LAS METAHEURÍSTICAS.....	82
4.5. ESTRUCTURAS APORTICADAS DE CONCRETO.	87
4.5.1. Definición	87
4.5.2. Métodos de análisis estructural.....	87
4.5.3. Enfoques de diseño	90
5. METAHEURÍSTICAS EN EL DISEÑO ESTRUCTURAL DE PORTICOS DE CONCRETO	92
5.1. FUNCIONES OBJETIVO EN PROBLEMAS DE DISEÑO ESTRUCTURAL ...	93
5.1.1. Costo estructural.....	94
5.1.2. Peso estructural	96
5.1.3. Energía de deformación.....	97
5.1.4. Energía potencial	97
5.1.5. Desplazamientos	98
5.1.6. Energía sísmica	98
5.1.7. Estructura de control.....	99

5.2. PROBLEMAS DE OPTIMIZACIÓN DE DISEÑO ESTRUCTURAL.....	100
5.2.1. Diseño de barras o elementos	103
5.2.2. Diseño topológico	105
5.3. RESULTADOS DE APLICACIONES DE METAHEURISTICAS EN ESTRUCTURAS APORTICADAS DE CONCRETO.....	107
6. CONCLUSIONES Y TRABAJOS FUTUROS.....	113
BIBLIOGRAFIA.....	117

LISTA DE TABLAS

Tabla 1. Información de las revistas científicas relacionadas en la figura 5.....	25
Tabla 2. Algoritmos metaheurísticos basados en la naturaleza.....	54
Tabla 3. Resultados de aplicaciones de metaheurísticas en edificaciones aportadas	108

LISTA DE FIGURAS

Figura 1. Criterio de búsqueda de información en la plataforma SCOPUS [5].	20
Figura 2. Resultados de búsqueda en Scopus de publicaciones por año relacionado con la aplicación de metaheurísticas en la optimización de estructuras aporticadas de concreto [5].....	21
Figura 3. Resultados de búsqueda en Scopus del porcentaje de uso de las metaheurísticas en las diferentes ramas del conocimiento [5].....	22
Figura 4. Resultados de búsqueda en Scopus del número de publicaciones por año, relacionada con la optimización de estructuras aporticadas de concreto en diferentes países por medio de algoritmos metaheurísticos [5].	23
Figura 5. Resultados de búsqueda en Scopus del número de publicaciones por año en las revistas científicas, relacionadas con la optimización de estructuras aporticadas de concreto en diferentes países por medio de algoritmos metaheurísticos [5].....	24
Figura 6. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando algoritmos genéticos [5].	28
Figura 7. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo de recocido simulado [5].	29
Figura 8. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando optimización por enjambre de partículas [5].	30
Figura 9. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando optimización por búsqueda de la armonía [5].	31
Figura 10. Resultados de búsqueda en Scopus de la optimización de estructuras aporticadas empleando optimización por colonia de hormigas [5].....	33

Figura 11. Resultados de búsqueda en Scopus de la optimización de estructuras aporticadas empleando Big Bang Big Crunch [5].....	34
Figura 12. Resultados de búsqueda en Scopus de la optimización de estructuras aporticadas empleando búsqueda del sistema cargado [5].....	35
Figura 13. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo de la luciérnaga [5].	36
Figura 14. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo murciélago [5].....	37
Figura 15. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo optimización de rayos [5].....	38
Figura 16. Métodos de optimización [6].	47
Figura 17. Se evidencian los dos criterios contradictorios, exploración (diversificación) contra la explotación (intensificación). Es razonable relacionar las soluciones únicas con la exploración, mientras que las basadas en una población están más orientadas a l.....	48
Figura 18. Genealogia de las metaheurísticas [3].	51
Figura 19. Modelo genérico de algoritmo metaheurístico [31].	53
Figura 20. Comparación entre la técnica de recocido (annealing) y templado (quenching) [1].	63
Figura 21. Diagrama de flujo básico técnica de recocido [1].....	65
Figura 22. Estructura básica de algoritmo “tabú simple” [33].	68
Figura 23. Diagrama de flujo de algoritmo tabú simple [1].	69
Figura 24. Modelo básico algoritmo genético simple [30].	71
Figura 25. Diagrama de cruzamiento de único punto [36].	73

Figura 26. Diagrama de ejemplo de mutación [36].	73
Figura 27. Esquema representativo del funcionamiento de un algoritmo evolutivo básico [35].	74
Figura 28. Algoritmo evolutivo más utilizado en las metaheurísticas [30].	75
Figura 29. Comportamiento de las hormigas Reales [50].	78
Figura 30. Estructura general de un algoritmo de colonia de hormigas [33].	80
Figura 31. Definición matemática formal de un problema multiobjetivo [30].	84
Figura 32. Definición matemática de optimalidad de Pareto [30].	84
Figura 33. Definición matemática del dominio de Pareto [56].	85
Figura 34. Definición matemática del conjunto de Pareto [30].	85
Figura 35. Definición matemática frente de Pareto [56].	86
Figura 36. Tiempo y exactitud de las estimaciones para un edificio de \$ 2,000,000 [32].	95
Figura 37. 1) Optimización de área. 2) Optimización de tamaño. 3) optimización de forma. 4) Optimización de topología. [37]	104
Figura 38. Clasificación del diseño topológico. 1) Sin optimización topológica con dos variantes a) barras b) Estructuras continuas. 2) Optimización topológica discreta. 3) Optimización topológica continua [30].	107

RESUMEN

TÍTULO: EMPLEO DE METAHEURÍSTICAS EN EL DISEÑO DE ESTRUCTURAS APORTICADAS DE CONCRETO*

AUTOR: EDER JAIR DURÁN CÁCERES**

PALABRAS CLAVE: OPTIMIZACIÓN, METAHEURÍSTICAS, ESTRUCTURAS, PORTICOS, CONCRETO.

DESCRIPCIÓN:

Hoy en día, muchos problemas de optimización de la vida real son complejos y requieren el uso de algoritmos adecuados para obtener soluciones satisfactorias. Como resultado, en áreas como la ingeniería civil, existe un interés creciente en los algoritmos metaheurísticos, porque presentan múltiples ventajas (capacidad de resolver problemas complejos y una buena adaptabilidad a diferentes problemas de optimización estructural). El presente trabajo busca analizar y describir la producción científica de los últimos años, así como los conceptos básicos y aplicaciones relacionadas con el uso de algoritmos metaheurísticos en el diseño de estructuras aporticadas de concreto. Lo anterior, con el fin de mostrar la tendencia de la aplicación de estos algoritmos en la optimización de este tipo de estructuras. El estudio usa la información obtenida en los motores de búsqueda especializados y el análisis bibliométrico, concerniente al tema "algoritmos metaheurísticos en diseño de pórticos de concreto. Los resultados indican que, aunque el crecimiento del uso de estos algoritmos fue relativamente lento, se espera que la diversificación y el desarrollo de nuevas técnicas y operadores especializados para problemas multiobjetivo, sean generadores de avances prometedores en la optimización de diseño de estructuras aporticadas de concreto.

* Monografía

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil. Director: Oscar Javier Begambre Carrillo.

ABSTRACT

TITLE: USE OF METAHEURISTICS IN REINFORCED CONCRETE FRAMES DESIGN*

AUTHOR: EDER JAIR DURÁN CÁCERES**

KEYWORDS: OPTIMIZATION, METAHEURISTICS, STRUCTURAL, FRAME, CONCRETE.

DESCRIPTION:

Nowadays, many real-life optimization problems are very complex and require the use of suitable algorithms to obtain satisfactory solutions. As a result, in areas such as civil engineering, there is a growing interest in metaheuristic algorithms, because they have multiple advantages (ability to solve complex problems and good adaptability to different structural optimization problems). This work intends to analyze and describe the scientific literature of recent years, as well as the basic concepts and applications related to the use of metaheuristic algorithms in reinforced concrete frames design. The foregoing, in order to show the trend of application of these algorithms in the optimization of this type of structures. The study uses information obtained in specialized search engines and bibliometric analysis, concerning the topic "metaheuristic algorithms in structural engineering". The results indicate that, although the growth rate of the use of these algorithms was relatively slow, it is expected that diversification and the development of new techniques and specialized operators for multiobjective problems, will generate promising advances in the optimization for reinforced concrete frames design.

* Monograph

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil. Director: Oscar Javier Begambre Carrillo.

INTRODUCCIÓN:

La palabra *heurística* proviene de la palabra griega “*heuriskein*”, que se interpreta como el arte de descubrir nuevos caminos para resolver las adversidades. Respecto al sufijo *meta*, también perteneciente a la cultura griega significa “metodología de nivel superior”. Partiendo de esto, distintos autores definen los métodos de búsqueda metaheurística como reglas generales de nivel superior que tienen amplia utilidad en el desarrollo de heurísticas subyacentes utilizadas para resolver situaciones de optimización específica ¹. En efecto, es importante definir la complejidad del problema, para esto es necesario tener en cuenta las instancias de entrada ya que juegan un papel importante a la hora de calificar la complejidad del problema. Ahora bien, la naturaleza de la situación (problema) a optimizar también es un factor que influye en la decisión de utilizar o no una metaheurística, esto debido a que algunos problemas de mediano y gran tamaño cuentan con características específicas para las cuales ya se desarrollaron algoritmos de optimización. donde finalmente el tiempo de resolución es el último factor a tener en cuenta ².

La literatura da muestra de algunos problemas en los cuales no es necesario utilizar la metaheurística, por ejemplo, para encontrar un árbol de expansión mínimo o una ruta más corta en un gráfico. Otro caso, la resolución de problemas de optimización polinómica, en estos se recomiendan utilizar los algoritmos de optimización exactos de tiempo. En efecto, independientemente del tipo de problema que se quiera solucionar es prudente estudiar si es posible reducirlo a

¹ P. Christensen, *An Introduction to Structural Optimization*, Solid Mechanics and Its Applications, 1st ED. (Dordrecht: Springer Netherlands, 2008), CLIII <<https://doi.org/10.1007/978-1-4020-8666-3>>.

² Patrick Siarry, *Metaheuristics*, ed. by Patrick Siarry, 1st edn (Cham: Springer International Publishing, 2016) <<https://doi.org/10.1007/978-3-319-45403-0>>.

un problema clásico, para así apoyarse en los algoritmos de optimización más utilizados.

Partiendo de la visión general del uso de las metaheurística y algoritmos de optimización, se puede afirmar que estos son una herramienta de gran ayuda en múltiples áreas de la ingeniería. Estas técnicas son necesarias a la hora de abordar problemas como la optimización de tiempos de cálculo en complejos de redes de transporte, estructuras de hormigón, que no se pueden resolver con métodos matemáticos exactos. Es necesario definir el término “función objetivo”, como la función matemática que define el fenómeno o problema a optimizar. En la ingeniería civil, generalmente, la función objetivo minimiza la cantidad y por ende el costo del material de la estructura. Esto evita la inexactitud que acarreaban las técnicas de prueba y error para determinar los parámetros estructurales del diseño. Esto toma especial importancia cuando el marco es estáticamente indeterminado (común en la práctica), en donde era necesario definir arbitrariamente las propiedades de los miembros que conformaban la sección transversal para que el análisis estructural permitiera determinar la respuesta del marco ³.

Lo planteado anteriormente expone un problema relevante cuando se trabaja con pórticos de tamaño real a gran escala, debido a que el número de grupos miembros aumenta, los ensayos numéricos tienden a ser muy cuantiosos por lo que el diseñador realiza algunas pruebas y selecciona la que proporcione un diseño fiable, teniendo en cuenta las disposiciones del código de diseño. Como resultado no se obtiene el diseño más idóneo, hablando en términos de cantidad de material ⁴.

³ Cheng y. Franklin and Z. Truman Kevin, *Structural Optimization Dynamic and Seismic Applicatios*, ed. by Spon Press, 1st edn (New York, 2010).

⁴ P Duysinx and P Tossings, ‘Structural Optimization’, *Structural Optimization*, 10.October (2011), 1–10.

Es aquí donde el presente escrito toma una relevancia particular donde es pertinente y necesario desarrollar una investigación acerca del tipo de técnicas que involucran algoritmos de optimización y cuanto se están implementando para solucionar los problemas expuestos anteriormente. Teniendo en cuenta lo anterior, este trabajo se encuentra dividido en cinco capítulos los cuales son: el capítulo dos dedicado al análisis bibliométrico del tema. En el capítulo tres se presenta un marco teórico de los conceptos básicos relacionados con optimización. En el capítulo cuatro se describen algunas de las metaheurísticas más conocidas aplicadas en problemas de optimización de estructuras aporticadas de concreto. En el capítulo cinco se presenta aplicaciones de metaheurísticas en el diseño estructural de estructuras aporticadas de concreto. Finalmente, el capítulo seis está dedicado a las conclusiones y trabajos futuros relacionados con el tema.

1. OBJETIVOS

1.1. OBJETIVO GENERAL.

Revisar el estado actual del uso de los algoritmos de optimización en el diseño de estructuras aporticadas en concreto.

1.2. OBJETIVOS ESPECIFICOS.

- Identificar la tendencia de uso de algoritmos de optimización metaheurísticos en el diseño de estructuras aporticadas de concreto.
- Identificar y describir los algoritmos metaheurísticos más usados en la optimización de estructuras aporticadas de concreto en la actualidad.
- Describir las funciones objetivo empleadas por las metaheurísticas en la optimización de estructuras aporticadas de concreto.

1.3. ALCANCE.

- En el presente trabajo se busca revisar, de forma crítica, el uso actual de las metaheurísticas en la optimización de estructuras aporticadas de concreto a partir de artículos publicados desde el año 2000 hasta el año 2019 en SCOPUS ⁵.

⁵ Elsevier B.V., 'Scopus', 2019 <<https://www.scopus.com/>> [accessed 8 March 2019].

2. ANALISIS BIBLIOMETRICO

Se toma en cuenta el análisis bibliométrico que proporciona la base de datos SCOPUS ® la cual contiene un registro de artículos científicos relacionados con el tema investigado.

2.1. CRITERIOS DE BÚSQUEDA EMPLEADOS Y RESULTADOS OBTENIDOS

En este contexto, se realizó una búsqueda en la plataforma SCOPUS ® con los siguientes criterios: palabras clave (reinforced concrete, frames, structural optimization), fecha desde el año 2000 hasta el año 2019 en la búsqueda se permitió explorar en todo tipo de documento (paper, ponencias, patentes, entre otros), se obtuvieron los resultados mostrados en las figuras 1 a 4.

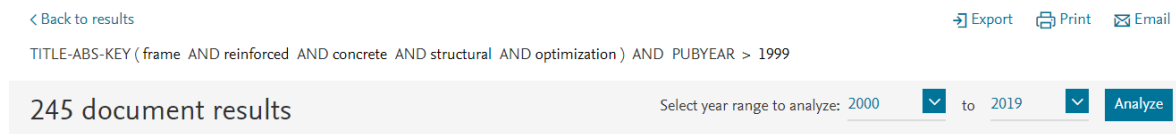


Figura 1. Criterio de búsqueda de información en la plataforma SCOPUS [5].

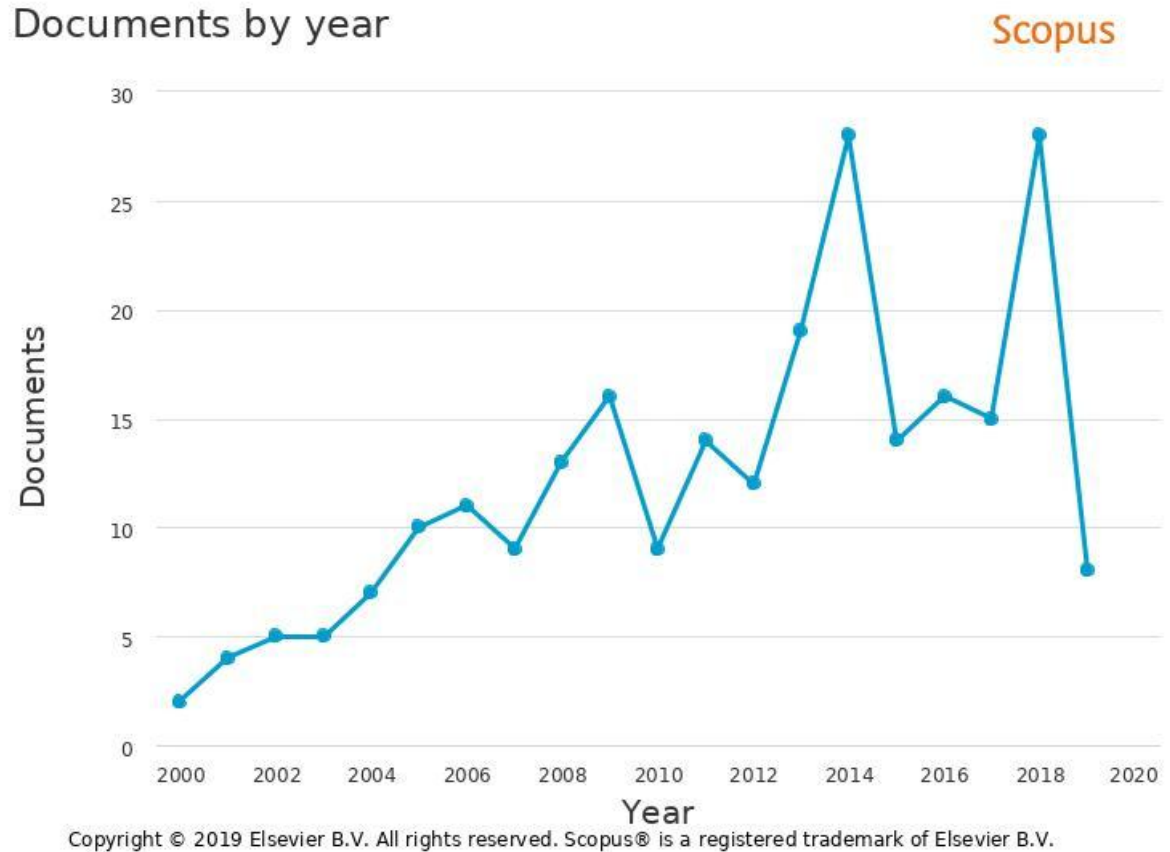


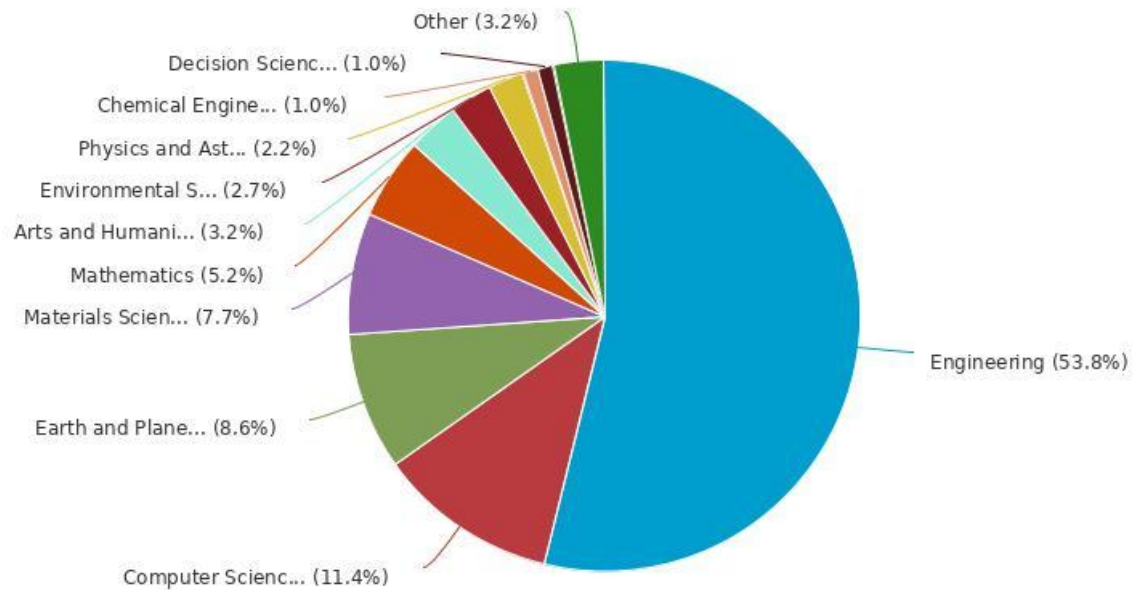
Figura 2. Resultados de búsqueda en Scopus de publicaciones por año relacionado con la aplicación de metaheurísticas en la optimización de estructuras aporticadas de concreto ⁶.

De acuerdo a la figura 2, Se evidenció que el uso de algoritmos en la optimización de pórticos de concreto tiene una tendencia al aumento en los últimos años, lo cual muestra que existe un creciente interés en la aplicación de estas herramientas de optimización en este tipo de estructuras.

⁶ Elsevier B.V.

Documents by subject area

Scopus



Copyright © 2019 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

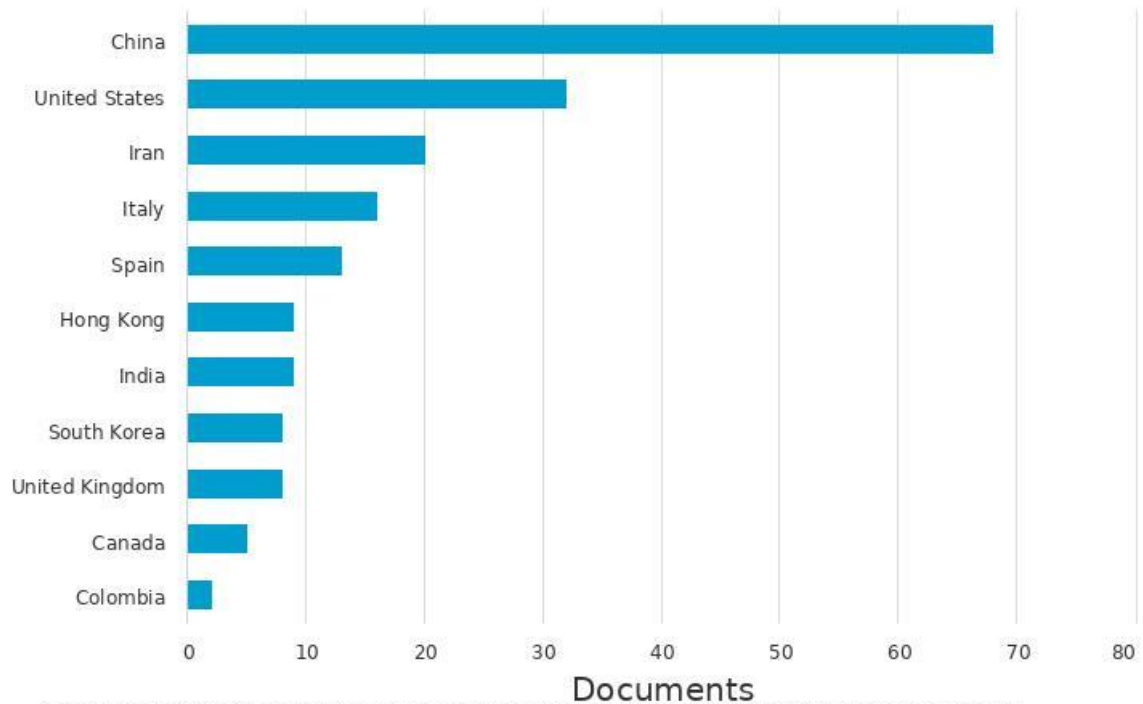
Figura 3. Resultados de búsqueda en Scopus del porcentaje de uso de las metaheurísticas en las diferentes ramas del conocimiento [5].

En la figura 3 se indica que la búsqueda realizada destaca que los algoritmos metaheurísticos son empleados en mayor proporción en áreas de investigación de la ingeniería y ciencias de la computación. También se observa que su uso también es muy importante en otras áreas como Ciencias de la Tierra, Ciencia de los materiales y Matemáticas. Aunque su uso es menor comparado con las áreas mencionadas, también tienen importantes aplicaciones en áreas como Artes y Humanidades, Medio Ambiente, Física, Química y en la toma Ciencia de la Toma de Decisiones.

Documents by country or territory

Compare the document counts for up to 15 countries/territories.

Scopus



Copyright © 2019 Elsevier B.V. All rights reserved. Scopus® is a registered trademark of Elsevier B.V.

Figura 4. Resultados de búsqueda en Scopus del número de publicaciones por año, relacionada con la optimización de estructuras aporticadas de concreto en diferentes países por medio de algoritmos metaheurísticos [5].

La figura 4 indica que el volumen de publicaciones correspondientes a la investigación de algoritmos de optimización metaheurísticos se focaliza en pocos países del mundo. En el continente asiático los países líderes son China, Iran, Hong Kong, India y Corea del Sur. En Europa se encuentran países como Italia, España y Reino Unido. Para nuestro continente los líderes son Estados Unidos y Canada. En nuestro país Colombia, aunque el volumen de investigación pueda parecer bajo, es un área de investigación que se encuentra en crecimiento y es de gran interés en áreas como la Ingeniería.

2.2. REVISTAS IDENTIFICADAS

En las siguientes figuras se identifican las revistas científicas relacionadas con la optimización de estructuras aporticadas con metaheurísticas.

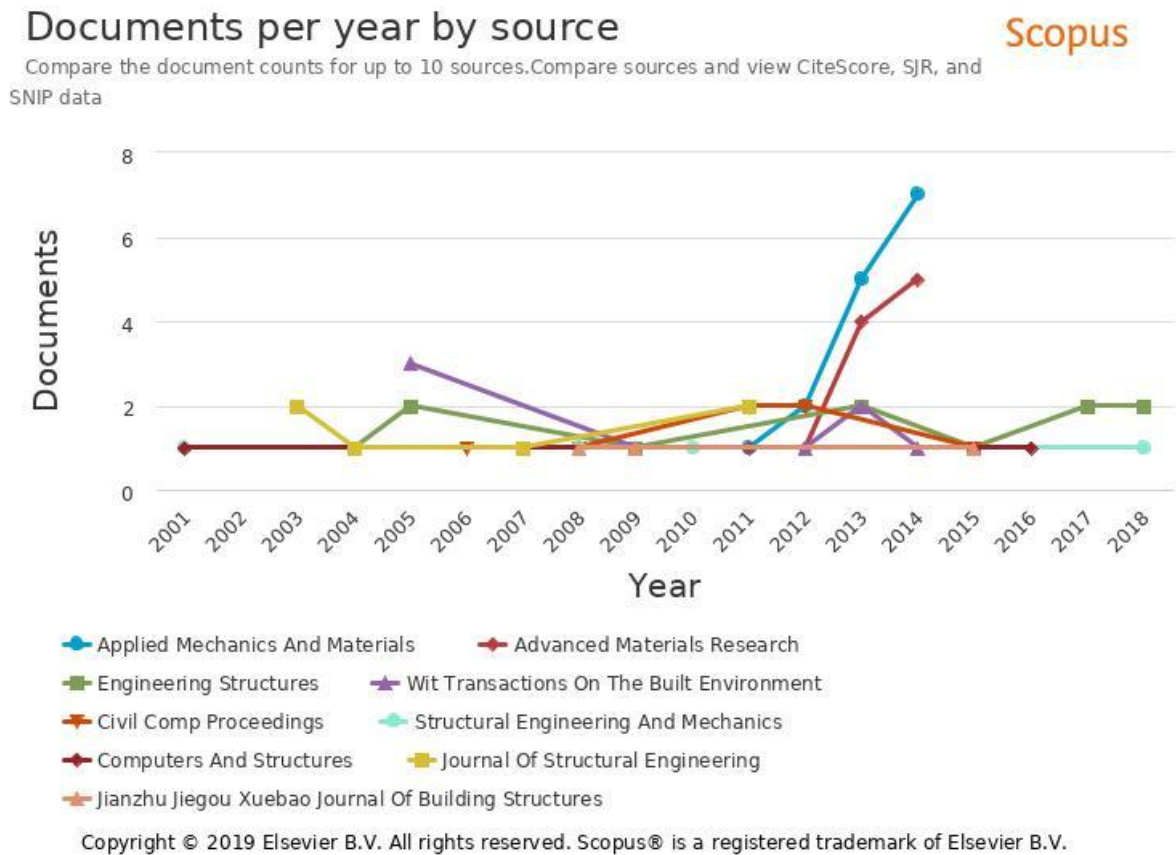


Figura 5. Resultados de búsqueda en Scopus del número de publicaciones por año en las revistas científicas, relacionadas con la optimización de estructuras aporticadas de concreto en diferentes países por medio de algoritmos metaheurísticos [5].

Tabla 1. Información de las revistas científicas relacionadas en la figura 5. ⁷⁸

REVISTA	PAÍS	IDIOMA	AREA TEMATICA	RANKING	FACTOR DE IMPACTO
Applied Mechanics And Materials.	Suiza	Inglés	Engineering Engineering (miscellaneous)	Q4	0.11
WIT Transactions on the Built Environment	Reino Unido	Inglés	Engineering Architecture Building and Construction Civil and Structural Engineering Safety, Risk, Reliability and Quality	Q3	0.12
Engineering Structures	Reino Unido	Inglés	Engineering :Civil and Structural Engineering	Q1	1.63
Structural Engineering and Mechanics	Corea del Sur	Inglés	Engineering: Building and Construction Civil and Structural Engineering Mechanical Engineering Mechanics of Materials	Q1	0.71
Advanced Materials Research	Suiza	Inglés	Engineering Engineering (miscellaneous)	Q2	0.121
Civil-Comp Proceedings	Reino Unido	Inglés	Engineering: Civil and Structural EngineeringEnvironmental Science: Environmental EngineeringComputer Science: Artificial IntelligenceComputer Science: Computational Theory and Mathematics	Q2	0.104
Jianzhu Jieqou Xuebao/Journal of Building Structures	China	Inglés	Engineering Building and Construction Civil and Structural Engineering	Q2	0.45
Computers and Structures	Reino Unido	Inglés	Computer Science Computer Science Applications Engineering Civil and Structural Engineering Mechanical Engineering Materials Science Materials Science (miscellaneous) Mathematics Modeling and Simulation	Q1	1.54
Journal of Structural Engineering	Estados Unidos	Inglés	Engineering Building and Construction Civil and Structural Engineering Mechanical Engineering Mechanics of Materials	Q1	1.91

⁷ Elsevier B.V.

⁸ Scimago Lab, 'Scimago Journal & Country Rank', 2019 <<https://www.scimagojr.com/>> [accessed 20 July 2019].

Donde el factor de impacto mide la frecuencia con la cual ha sido citado el artículo promedio de una revista en un año en particular. Es el índice bibliométrico más utilizado, sirve para comparar revistas y evaluar la importancia relativa de una revista concreta dentro de un mismo campo científico.

Se calcula generalmente con base en un periodo de dos años. Por ejemplo, el factor de impacto en el año 2018 para una determinada publicación puede calcularse como sigue:

X = Número de veces en que los artículos publicados en esta revista en el periodo 2016-2017 han sido citados por las publicaciones a las que se les da seguimiento a lo largo del año 2018.

Y = Número de artículos publicados en esta revista en el periodo 2016-2017.

Factor de impacto 2018 = X/Y .

El ranking (conocido como cuartil) es un indicador que sirve para evaluar la importancia relativa de una revista dentro del total de revistas de su área. Se calcula ordenando las revistas en orden descendente por factor de impacto. Se divide el total de revistas en cuatro partes iguales. Cada parte es un cuartil. Las revistas con el factor de impacto más alto se encuentran en el primer cuartil, las revistas con el factor de impacto más bajo se encuentran en el cuarto cuartil y en la parte central se encuentran el segundo y el tercero.

2.3. ALGORITMOS EMPLEADOS CON MAYOR FRECUENCIA.

Muchos de los problemas de optimización de diseño en ingeniería civil son altamente no lineales y con restricciones complejas, difíciles de resolver usando

los métodos de optimización tradicionales. En este caso los algoritmos metaheurísticos se convierten en una herramienta adecuada y efectiva para resolver este tipo de problemas.

A continuación, se presentan los algoritmos metaheurísticos que han sido usados con mayor frecuencia en la optimización de estructuras en ingeniería civil:

2.3.1. Algoritmos genéticos (GA): Es uno de los algoritmos más antiguos, y su aplicación data desde el año 1986. Se ha aplicado en problemas de optimización estructural, problemas de identificación estructural, diseño de puentes de largas luces, optimización topológica estructuras de techos en acero, optimización topológica de armaduras, etc. Se han usado también en el área de la ingeniería de transporte en problemas de flujo de tráfico y programación logística ⁹. En la figura 6 se aprecian las palabras claves usadas y los resultados obtenidos en la búsqueda realizada en SCOPUS desde el año 2000, relacionada con la utilización de algoritmos genéticos en la optimización de estructuras a porticadas de concreto.

⁹ Sinan Melih Nigdeli, Gebrail Bekda, and Xin-she Yang, *Metaheuristics and Optimization in Civil Engineering*, 2016, vii <<https://doi.org/10.1007/978-3-319-26245-1>>.

Scopus

Analyze search results

< Back to results

Export Print Email

TITLE-ABS-KEY (reinforced AND concrete AND frames AND structural AND optimization AND genetic AND algorithm) AND PUBYEAR > 1999

43 document results

Select year range to analyze: 2001 to 2019

Analyze

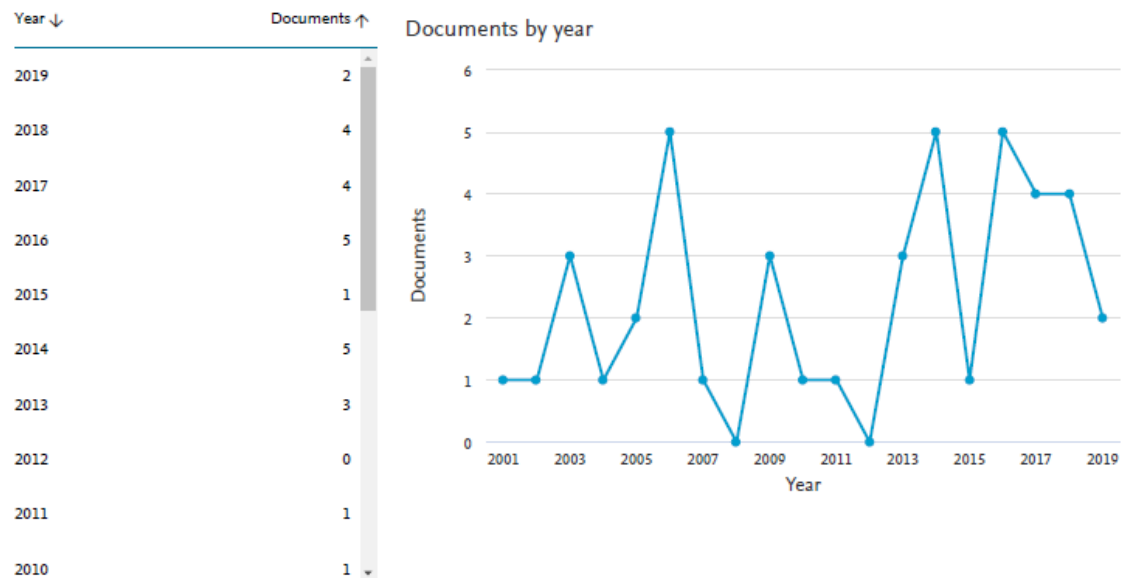


Figura 6. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando algoritmos genéticos [5].

A partir de los resultados de búsqueda reflejados en la figura 6, se puede observar que la tendencia del uso de los algoritmos genéticos en la optimización de estructuras aporticadas tiene una tendencia de aumento en los últimos años, lo cual muestra el interés por aplicar estos algoritmos en la optimización de este tipo de estructuras.

2.3.2. Recocido simulado (SA): Se ha usado en diversos problemas de optimización en ingeniería civil: Planeación de sistemas de transporte ferroviario,

colocación óptima de sensores, optimización de domos bajo excitación sísmica y optimización térmica de edificaciones ¹⁰. En la figura 7 se observa los resultados de la búsqueda realizada en SCOPUS relacionada con el uso del recocido simulado en la optimización de estructuras aporticadas de concreto reforzado.

Scopus

Analyze search results

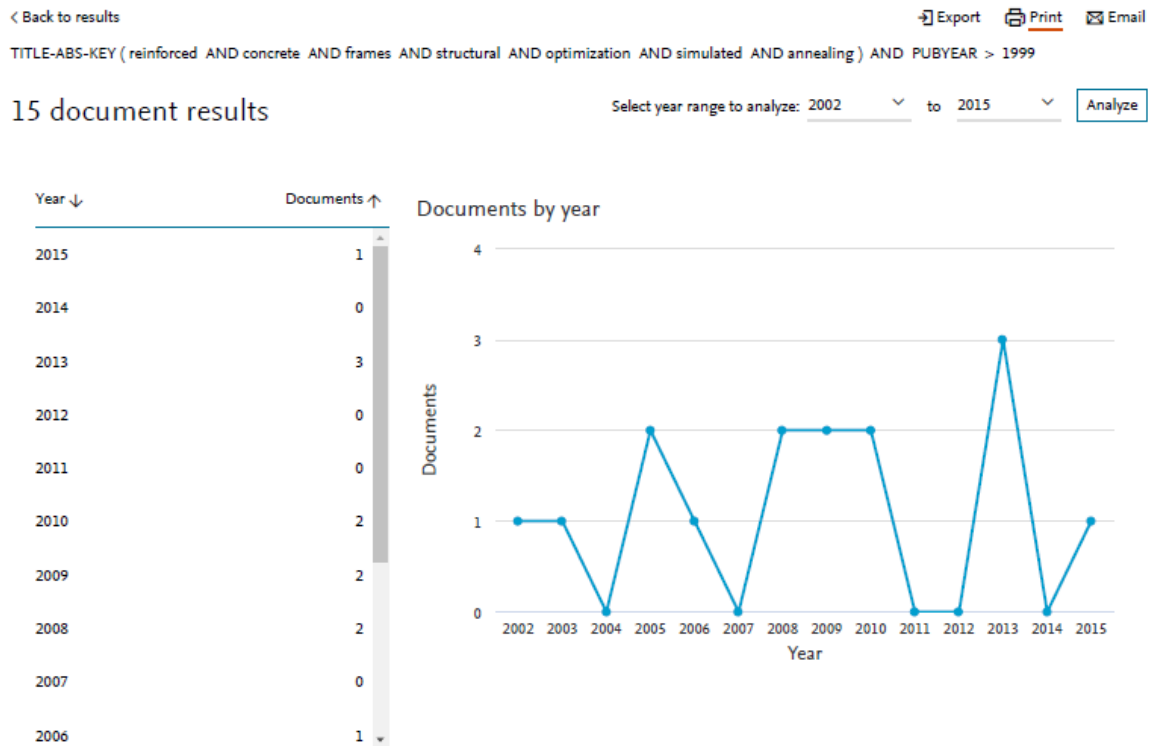


Figura 7. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo de recocido simulado [5].

¹⁰ Arpan Kumar Kar, 'Bio Inspired Computing - A Review of Algorithms and Scope of Applications', *Expert Systems with Applications*, 59 (2016), 20–32 <<https://doi.org/10.1016/j.eswa.2016.04.018>>.

De acuerdo a los datos obtenidos por medio de la búsqueda realizada, se puede observar que el uso del algoritmo recocido simulado ha sido constante durante los últimos años en la optimización de estructuras aporticadas de concreto.

2.3.3. Optimización por enjambre de partículas (PSO): Se ha aplicado en diversos problemas de optimización de diseño estructural, como el diseño de edificaciones de gran altura, optimización de tamaño en armaduras, análisis de estabilidad de taludes y sistemas de distribución de agua ¹¹. En la figura 8 se aprecian las palabras claves usadas y los resultados en la búsqueda realizada en SCOPUS desde el año 2000.

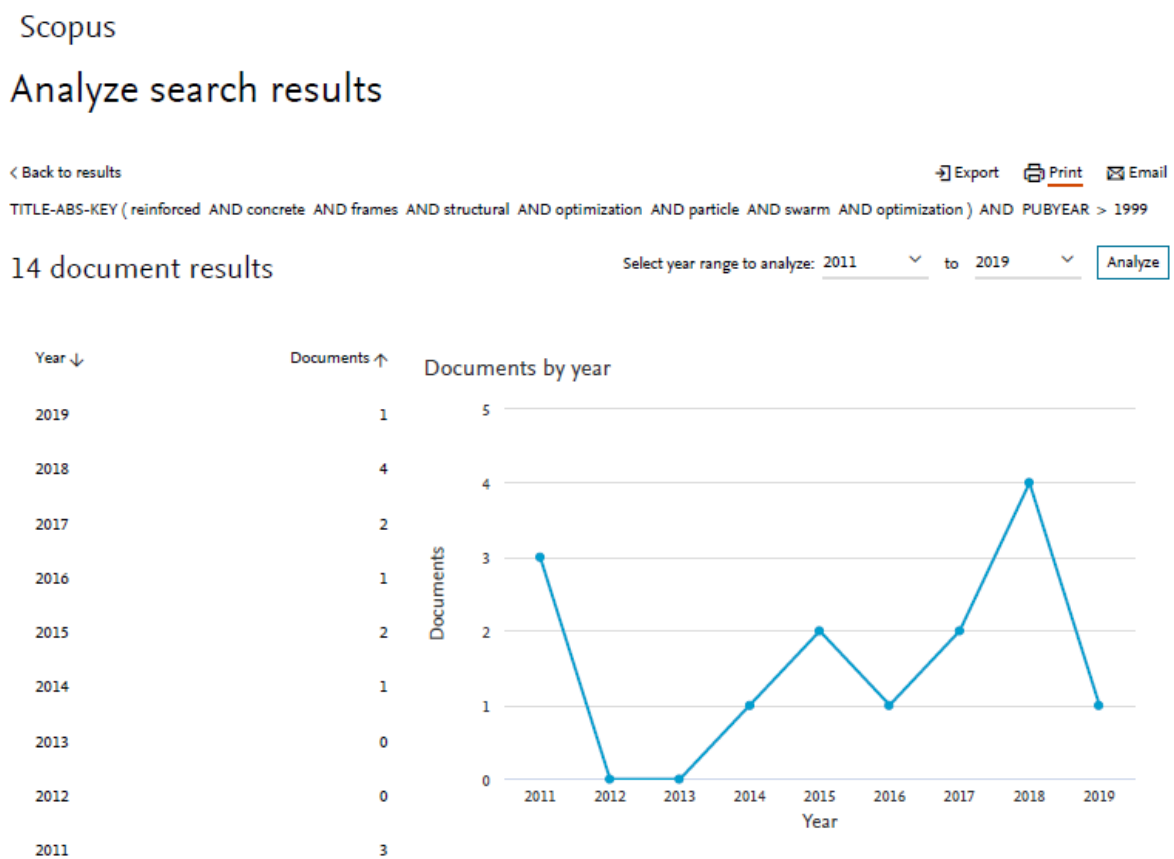


Figura 8. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando optimización por enjambre de partículas [5].

¹¹ Kar.

La figura 8 indica que en los últimos años el uso de la optimización por enjambre de partículas en las estructuras aporticadas de concreto ha aumentado, lo cual indica el interés de este algoritmo por parte de los investigadores del área.

2.3.4. Búsqueda de la armonía (HS): Es un algoritmo que ha tenido aplicaciones en el área de la ingeniería estructural, ingeniería geotécnica, ingeniería de recursos hídricos, ingeniería ambiental e ingeniería de tránsito. En la figura 9 se observa los resultados de la búsqueda realizada en SCOPUS relacionada con el uso de la búsqueda de la armonía en la optimización de estructuras aporticadas de concreto reforzado.

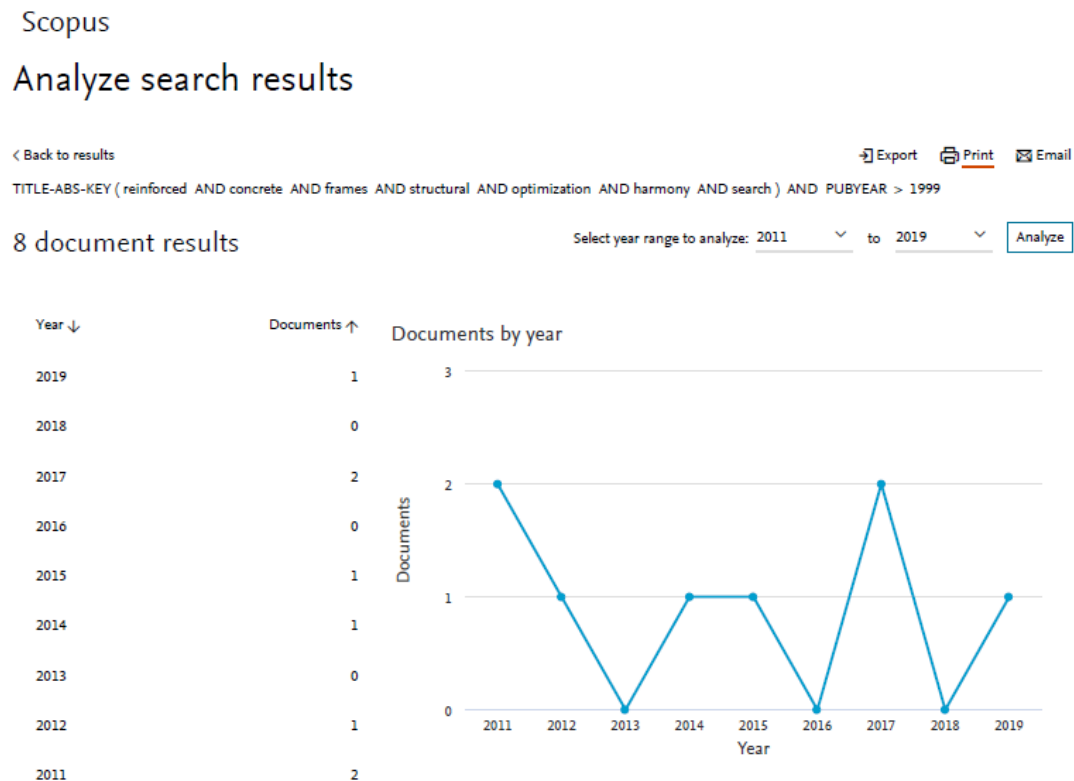


Figura 9. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando optimización por búsqueda de la armonía [5].

Al observar los datos obtenidos en la búsqueda realizada en SCOPUS se puede analizar que es un algoritmo relativamente nuevo, ya que su uso en los problemas de optimización de estructuras aporticadas, según los registros de la base de datos, inició desde el año 2011, fecha a partir de la cual su uso se ha mantenido constante.

2.3.5. Optimización por colonia de hormigas (ACO): Las aplicaciones en ingeniería civil incluyen problemas de ruteo de vehículos, problemas de tráfico, problemas de determinación de la superficie de deslizamiento crítico no circular en el análisis de estabilidad de taludes y problemas de optimización estructural multiobjetivo ¹². En la figura 10 se observa los resultados de la búsqueda realizada en SCOPUS relacionada con el uso de colonia de hormigas en la optimización de estructuras aporticadas de concreto reforzado.

¹² Kar.

Scopus

Analyze search results

[Back to results](#)

[Export](#) [Print](#) [Email](#)

TITLE-ABS-KEY (reinforced AND concrete AND frames AND structural AND optimization AND ant AND colony AND optimization) AND PUBYEAR > 1999

6 document results

Select year range to analyze: 2010 to 2017 [Analyze](#)

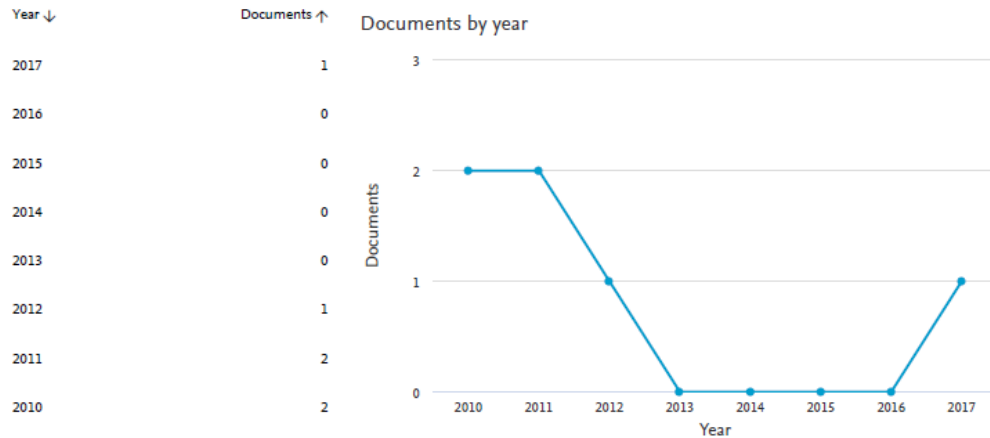


Figura 10. Resultados de búsqueda en Scopus de la optimización de estructuras aporticadas empleando optimización por colonia de hormigas [5].

Al analizar los datos obtenidos en la búsqueda realizada en SCOPUS se puede observar que la aplicación del algoritmo de optimización por colonia de hormigas en los problemas de estructuras aporticadas tiene un uso relativamente reciente a partir del año 2010, y su uso no ha sido frecuente en los últimos años.

2.3.6. Algoritmos Big Bang Big Crunch: Se ha usado en la optimización de armaduras, pórticos, estimación de paramentos de estructuras y muros de contención. En la figura 11 se observa los resultados de la búsqueda realizada en SCOPUS relacionada con el algoritmo Big Bang Big Crunch en la optimización de estructuras aporticadas de concreto reforzado ¹³¹⁴.



Figura 11. Resultados de búsqueda en Scopus de la optimización de estructuras aporticadas empleando Big Bang Big Crunch [5].

Al observar los datos obtenidos en la búsqueda realizada en SCOPUS se puede determinar que es un algoritmo relativamente nuevo, ya que su uso en los problemas de optimización de estructuras aporticadas, según los registros de la

¹³ Nigdeli, Bekda, and Yang, VII.

¹⁴ Kar.

base de datos, inició desde el año 2011, fecha a partir de la cual su uso ha sido poco frecuente.

2.3.7. El algoritmo búsqueda de sistema cargado CSS: se ha aplicado en problemas de ingeniería civil como la detección de daño en estructuras reticulares, optimización del costo en vigas, diseño óptimo de estructuras, amortiguadores de masa sintonizados y amortiguadores de masa sintonizados semiactivos ¹⁵¹⁶. En la figura 12 se aprecian las palabras claves usadas y los resultados en la búsqueda realizada en SCOPUS.

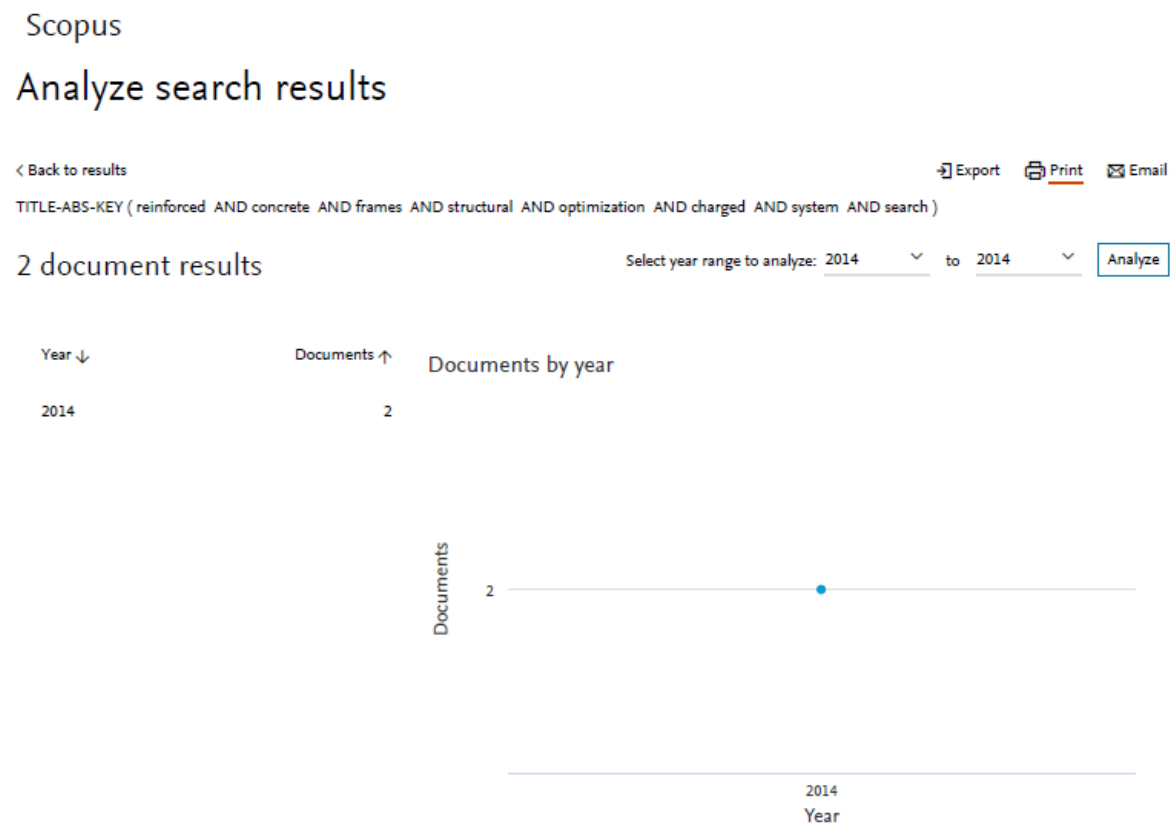


Figura 12. Resultados de búsqueda en Scopus de la optimización de estructuras aporticadas empleando búsqueda del sistema cargado [5].

¹⁵ Nigdeli, Bekda, and Yang, VII.

¹⁶ Kar.

Al observar los datos obtenidos en SCOPUS se puede determinar que el uso de este algoritmo en la optimización estructural inicio hace poco tiempo (año 2014) y ha sido poco usado en la optimización de estructuras aporticadas de concreto.

2.3.8. Algoritmo de la luciérnaga (FA): Ha tenido aplicaciones de optimización en el área de la ingeniería estructural (diseño de torres, losas de acero continuamente fundidas y armaduras) e ingeniería de transporte (planificación de rutas) ¹⁷. En la figura 13 se observa los resultados de la búsqueda realizada en SCOPUS relacionada con el algoritmo de la luciérnaga en la optimización de estructuras aporticadas de concreto reforzado.

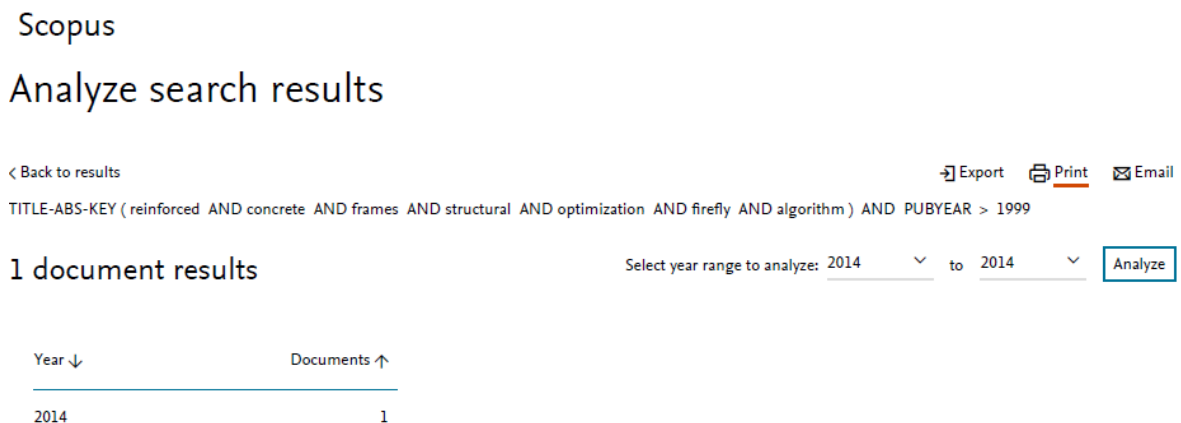


Figura 13. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo de la luciérnaga [5].

Al analizar los datos obtenidos en SCOPUS se puede determinar que el uso de este algoritmo en la optimización estructural inicio hace muy poco tiempo (año

¹⁷ Nigdeli, Bekda, and Yang, VII.

2014) y ha sido especialmente poco usado en la optimización de estructuras aporticadas de concreto.

2.3.9. Algoritmo murciélago (BA): Se ha usados en diversos problemas de optimización en ingeniería estructural como la colocación óptima de muros de cortante de placas de acero, diseño óptimo de armaduras y diseño de vigas de concreto reforzado. También se ha usado en problemas de ruteo de vehículos y en sistemas de operación de reservorios (yacimientos) ¹⁸. En la figura 14 se aprecian las palabras claves usadas y los resultados en la búsqueda realizada en SCOPUS.

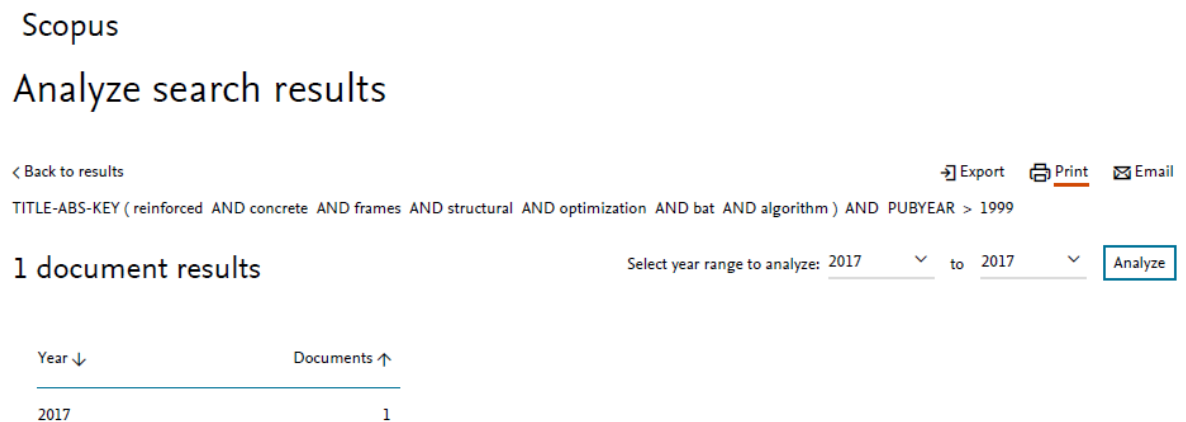


Figura 14. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo murciélago [5].

Al analizar los datos obtenidos en la búsqueda, se puede determinar que el uso de este algoritmo en la optimización de estructura aporticadas es nuevo ya que se observa solo una publicación en el año 2017.

¹⁸ Nigdeli, Bekda, and Yang, vii.

2.3.10. El algoritmo optimización de rayos RO: se ha empleado en ingeniería estructural (optimización de tamaño y forma de armaduras) e ingeniería de transporte (diseño de capas granulares para vías férreas) ¹⁹²⁰. En la figura 15 se observa los resultados de la búsqueda realizada en SCOPUS relacionada con el algoritmo de optimización de rayos en la optimización de estructuras aporticadas de concreto reforzado.

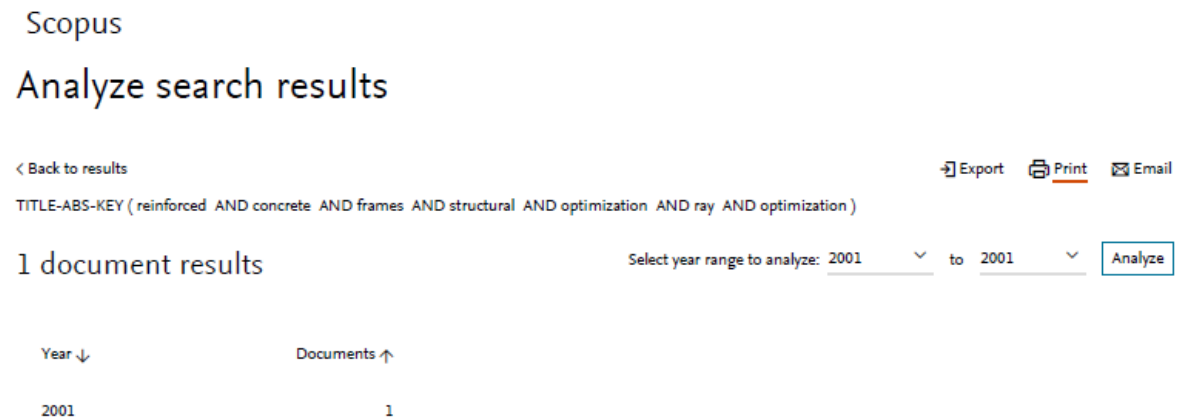


Figura 15. Resultados de búsqueda en Scopus: optimización de estructuras aporticadas empleando el algoritmo optimización de rayos [5].

Al analizar los datos obtenidos en la búsqueda, se puede determinar que es poco usado en la optimización de estructuras aporticadas de concreto ya que presenta una sola publicación en el año 2001.

2.3.11. Búsqueda Cuckoo (CS): Se han desarrollado metodologías basadas en el CS para el diseño óptimo de pórticos de acero, armaduras. De igual manera se ha utilizado para resolver el problema del vendedor viajero ²¹. Este algoritmo no ha

¹⁹ Nigdeli, Bekda, and Yang, VII.

²⁰ Kar.

²¹ Nigdeli, Bekda, and Yang, VII.

sido empleado en la optimización de pórticos de concreto, ya que al realizar la búsqueda en SCOPUS no se encontró resultado sobre este.

2.4. LISTA DE SOFTWARE COMERCIAL

A continuación, se muestra una lista con diferentes programas de uso comercial empleados en el área de ingeniería civil y mecánica, los cuales contienen un módulo de optimización estructural (especialmente topológica), que usan principalmente algoritmos clásicos para el diseño conceptual de estructuras:

- COMSOL Multiphysics (<https://www.comsol.com/>) Incluye métodos clásicos de optimización basados en gradiente y métodos libres de gradiente: Los métodos basados en gradiente incluyen: adjoint method of the SNOPT optimizer, second gradient-based algorithm, Method of Moving Asymptotes (MMA). Los métodos libres de gradiente incluyen: Bound Optimization by Quadratic Approximation (BOBYQA), Constraint Optimization by Linear Approximation (COBYLA), Nelder-Mead, coordinate search method, Monte Carlo method.
- ANSYS (<https://www.ansys.com/>): Diseño óptimo basado en la técnica de ajuste de curvas (curve-fitting technique) para aproximar la respuesta utilizando varios vectores de diseño de prueba.
- MSC/Nastran (<https://www.mscsoftware.com/product/msc-nastran>): Optimización estructural basada en estática, frecuencias naturales. análisis de pandeo, conceptos de aproximación y análisis de sensibilidad

En general, el uso de algoritmos de optimización metaheurísticos en software de uso común en ingeniería civil (ej. SAP2000, ETABS) aún no se encuentra extendido, aunque existen diferentes programas y extensiones (algunos de uso propietario y otros de uso libre), en los cuales pueden encontrarse algunas librerías que incluyen algoritmos de optimización metaheurísticos de gran popularidad como GA, SA, GP, PSO. De igual manera pueden encontrarse repositorios web donde pueden descargarse y usarse de forma libre (respetando los derechos de autor) una gran variedad de algoritmos metaheurísticos uniobjetivo y multiobjetivo. A continuación, se muestra una lista con software que usa algoritmos metaheurísticos y repositorios web donde pueden encontrarse los códigos de los algoritmos, los cuales pueden ser adaptados a diversos problemas de optimización.

Software:

- MATLAB (Propietario): <https://www.mathworks.com/products/matlab.html>
- ParadisEO (Libre): <http://paradiseo.gforge.inria.fr/>
- Open BEAGLE (Libre): <https://github.com/chgagne/beagle>
- ECJ (Libre): <https://cs.gmu.edu/~eclab/projects/ecj/>
- ACO Iridia (Libre): <http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/>
- METSlib (Libre): <https://projects.coin-or.org/metslib>

Repositorios y sitios web:

- Evolutionary Multiobjective Optimization:
<http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOOsoftware.html>
- Particle Swarm Optimization:

<http://www.particleswarm.info/Programs.html>

- CMA-ES: Modern evolution strategies

http://cma.gforge.inria.fr/cmaes_sourcecode_page.html

Los algoritmos más utilizados en la optimización de estructural aporticadas son los algoritmos genéticos, el algoritmo por enjambre de partículas y el algoritmo de recocido simulado. Algoritmos metaheurísticos como el algoritmo de la luciérnaga, o el algoritmos murciélago tienen aplicaciones hasta ahora recientes, lo cual indica la novedad de estos métodos en esta área de investigación.

3. METODOS EMPLEADOS EN OPTIMIZACIÓN.

Al iniciar la optimización de una estructura se presenta un primer problema que consiste en seleccionar el método de optimización (métodos exactos o los aproximados). Ambos tipos de métodos proporcionan una herramienta adecuada para la optimización de estructuras. Es importante recalcar que los exactos son los más utilizados, ya que se basan en programación lineal iterativa, que es relativamente sencilla. En contraste, los métodos heurísticos utilizan técnicas derivadas de la inteligencia artificial, como lo son los algoritmos genéticos, la cristalización simulada, las colonias de hormigas, la búsqueda tabú, etc. Su principal diferencia radica en que, en los métodos exactos, los resultados obtenidos siempre son iguales (cada vez que se ejecute el algoritmo) y se garantiza la funcionalidad, mientras que en los aproximados (o heurísticos) generan respuestas adecuadas en un tiempo apropiado aunque no se garantiza que sea el óptimo global ²².

Los métodos heurísticos han sido utilizados ampliamente en la ingeniería estructural. Los primeros trabajos aplicados a esta área fueron llevados a cabo por Jenkinsy de Rajeev y Krishnamoorthy en 1991 los cuales utilizaron algoritmos genéticos en la optimización de peso de estructuras metálicas. Posteriormente en 1997 se realizan las primeras las primeras intervenciones en estructuras de concreto, registrado en el trabajo de Coello et al, quienes optimizaron las vigas de hormigón armado mediante algoritmos genéticos. Además, en 1991-1992 Rajeev y Krishnamoorthy aplicaron algoritmos genéticos en la construcción de pórticos de hormigón armado. Más adelante, en 2003 Hrstka al. y Leps y Sejnoha realizaron la

²² El-Ghazali Talbi, *Metaheuristics*, 1st edn (Hoboken: John Wiley & Sons, Inc., 2009).

optimización de vigas de hormigón armado. Lee y Ahn ²³ y Camp ²⁴ aplicaron algoritmos genéricos a los pórticos de edificación de hormigón armado y finalmente la optimización en la construcción de edificaciones macizas de Sahab ²⁵.

3.1. METODOS EXACTOS.

La mayoría funciona como un algoritmo de búsqueda por árboles, realizan sus proyecciones en un espacio factible y resuelve el problema dividiéndolo en problemas menos complejos éstos métodos son viables a instancias de problemas muy pequeños con una convergencia lenta. A continuación, se profundizará cada uno de los algoritmos mencionados anteriormente ²⁶.

²³ C. Lee and J. Ahn, 'Flexural Design of Reinforced Concrete Frames by Genetic Algorithm', *Journal of Structural Engineering*, 129.6 (2003), 762–74 <[https://doi.org/10.1061/\(ASCE\)0733-9445\(2003\)129:6\(762\)](https://doi.org/10.1061/(ASCE)0733-9445(2003)129:6(762))>.

²⁴ Charles V. Camp, Shahram Pezeshk, and Håkan Hansson, 'Flexural Design of Reinforced Concrete Frames Using a Genetic Algorithm', *Journal of Structural Engineering*, 129.1 (2003), 105–15 <[https://doi.org/10.1061/\(ASCE\)0733-9445\(2003\)129:1\(105\)](https://doi.org/10.1061/(ASCE)0733-9445(2003)129:1(105))>.

²⁵ Ignacio Payá and others, 'Optimización Heurística de Pórticos de Edificación de Hormigón Armado', *Métodos Numéricos Para Cálculo y Diseño En Ingeniería: Revista Internacional*, 22.3 (2006), 241–60.

²⁶ Jesica de Armas Adrian, 'Problemas de Corte: Metodos Exactos y Aproximados Para Formulaciones Mon y Multi-Objetivo (Tesis Doctoral)' (Universidad de la Laguna, 2011).

3.1.1. La programación dinámica. Su funcionamiento está basado en la división repetitiva del problema en estructuras más sencillas. Toma como fundamento el principio de Bellman, el cual indica que: “la subpolítica de una política óptima es en sí misma óptima”. En efecto, este método se basa en que las soluciones parciales pueden construir una solución general óptima. Además, el tiempo de resolución puede verse reducido mediante la eliminación de decisiones parciales que no conducen hacia la solución óptima ²⁷.

²⁷ Richard Bellman, *Dynamic Programming*, 2nd edn (United States: Dover Publications Inc., 2003).

3.1.2. El algoritmo de ramificación y límite. De manera general, este algoritmo enumera todas las soluciones posibles del problema tratado. Posteriormente, se construye un árbol dinámico cuyo nodo principal es el nodo raíz es el problema que se está resolviendo y todo el espacio de búsqueda que se asocia a él. En contraste, los nodos de las hojas son las soluciones parciales y los nodos internos son los subproblemas que conforman la solución total. La eliminación de los nodos de las hojas se realiza mediante una función límite que selecciona solo las soluciones más óptimas. Este algoritmo continua con la búsqueda incluso después de encontrar un camino a una solución actualizando la mejor opción de este modo descarta soluciones parciales ²⁸.

3.1.3. Programación de restricciones. Utiliza al igual que el algoritmo anterior la metodología del árbol además de implicaciones lógicas. El modelamiento de los problemas se realiza mediante dos conjuntos asociados, uno de variables y el otro de restricciones. Las variables pueden tomar valores reales, por lo general, números finitos enteros, mientras que las restricciones pueden ser símbolos o expresiones matemáticas.

3.2. ALGORITMOS APROXIMADOS.

Los algoritmos de aproximación, como los heurísticos, encuentran soluciones a funciones objetivo en un tiempo razonable.

Por lo general, las heurísticas son utilizadas para resolver problemas de gran extensión y complejidad. Su campo de acción es muy amplio y tienen costo computacional razonable (Pueden ejecutarse con las herramientas

²⁸ de Armas Adrian.

computacionales disponibles en la actualidad). De acuerdo al grado de especificación los algoritmos heurísticos se clasifican en dos: metaheurísticas y heurísticas específicas. Las metaheurísticas son utilizadas con propósito general, es decir, pueden ser una herramienta para una gran gama de problemas de optimización, a diferencia de las heurísticas específicas que son utilizados para problemas específicos. Ahora bien, las metaheurísticas pueden definirse como técnicas de nivel superior que sirven para diseñar heurísticas subyacentes para trabajar con problemas específicos ²⁹.

A pesar de que la inteligencia artificial apareció a mediados de los años 50, la aplicación de redes neuronales, la programación automática y la solución de problemas con restricciones, solo fue implementada en ingeniería de estructuras desde a finales de los años ochenta ³⁰. Debido a que en el diseño de estructuras de hormigón es necesario la selección de distintas variables con restricciones, la utilización de metaheurísticas es viable para solucionar estos inconvenientes. En contraste, el predimensionamiento de estructuras tradicionalmente se realiza mediante métodos de análisis de esfuerzos y de verificación de estados, además de la comparación con obras realizadas anteriormente, basándose en gran medida de la experiencia del ingeniero. Todo esto genera un proceso que en gran medida es empírico, dejando a un lado pensamientos objetivos a la hora de seleccionar los materiales y sus dimensiones. Esto da como resultado una estructura segura, pero no se garantiza que la disposición de los materiales utilizados sea la óptima ³¹.

²⁹ Talbi.

³⁰ Nigdeli, Bekda, and Yang, VII.

³¹ Payá and others.

Finalmente, la síntesis de los métodos utilizados para la solución de problemas de optimización se muestra en la figura 16.

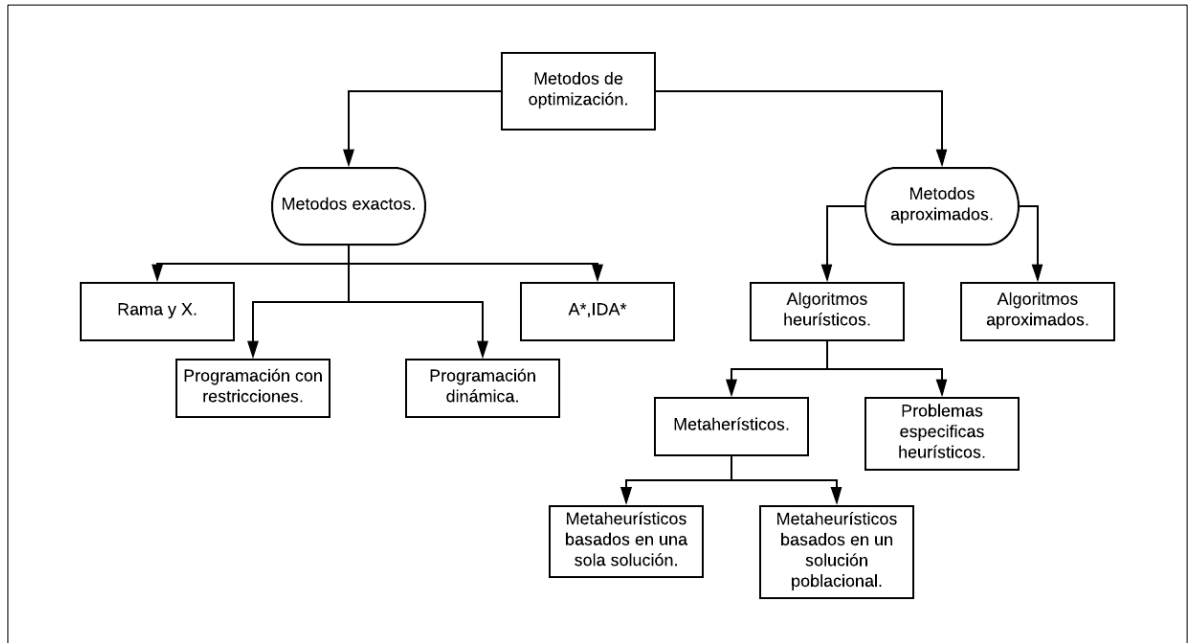


Figura 16. Métodos de optimización [6].

3.3. METAHEURÍSTICA.

La metaheurística resuelve instancias de problemas que se consideran difíciles en general, explorando el espacio de búsqueda de soluciones generalmente grande de estas instancias. Estos algoritmos logran esto al reducir el tamaño efectivo del espacio y al explorar ese espacio de manera eficiente. Las metaheurísticas tienen tres propósitos principales: resolver problemas más rápido, resolver problemas grandes y obtener algoritmos sólidos. El diseño de una metaheurística debe tener en cuenta distintos criterios contradictorios como: la diversificación, que consiste

en la exploración del espacio de búsqueda y la intensificación, que explora las mejores soluciones encontradas. Partiendo de esto se definen las regiones prometedoras que están conformadas por las buenas soluciones. Es decir, en la exploración se ahonda más en las soluciones en busca de encontrar mejores; mientras que la diversificación aquellas que no han sido exploradas. Véase la figura 17.

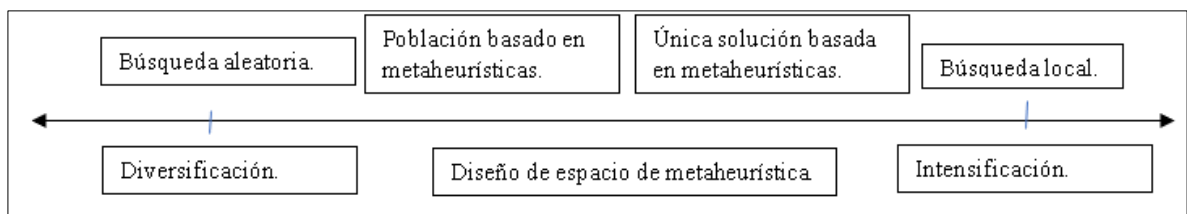


Figura 17. Se evidencian los dos criterios contradictorios, exploración (diversificación) contra la explotación (intensificación). Es razonable relacionar las soluciones únicas con la exploración, mientras que las basadas en una población están más orientadas a l

En la figura 18. Se muestra la genealogía de las metaheurísticas más representativas. Polya introdujo el concepto heurístico para resolver problemas de optimización en 1945 ³². Las referencias originales de las siguientes metaheurísticas representadas en la figura se basan en su aplicación a la optimización y / o problemas de aprendizaje automático: ACO (optimización de colonias de hormigas) ³³, AIS (sistemas inmunitarios artificiales) ³⁴, BC (colonia de

³² G. Polya, *How to Solve It: A New Aspect of Mathematical Method, Optimización Heuristic*, 2nd edn (New York: Penguin Science, 1990).

³³ M. Dorigo, 'Optimization, Learning and Natural Algorithms (PhD Thesis)' (Politecnico di Milano, 1992).

³⁴ J.Doyne Farmer, Norman H Packard, and Alan S Perelson, 'The Immune System, Adaptation, and Machine Learning', *Physica D: Nonlinear Phenomena*, 22.1–3 (1986), 187–204 <[https://doi.org/10.1016/0167-2789\(86\)90240-X](https://doi.org/10.1016/0167-2789(86)90240-X)>.

abejas)³⁵, CA (algoritmos culturales)³⁶, CEA (algoritmos coevolutivos)³⁷, CMA-ES (matriz de covarianza estrategia de evolución de la adaptación)³⁸, DE (evolución diferencial)³⁹, EDA (estimación de algoritmos de distribución)⁴⁰, EP (programación evolutiva)⁴¹, ES (estrategias de evolución)⁴², GA (algoritmos genéticos)⁴³, GDA (gran diluvio)⁴⁴, GLS (búsqueda local guiada)⁴⁵, GP (programación genética)⁴⁶, GRASP (procedimiento de búsqueda codicia adaptativa)⁴⁷, ILS (búsqueda local iterada)⁴⁸, NM (método ruidoso)⁴⁹, PSO

³⁵ Y. Yonezawa and T. Kikuchi, 'Ecological Algorithm for Optimal Ordering Used by Collective Honey Bee Behavior', in *MHS'96 Proceedings of the Seventh International Symposium on Micro Machine and Human Science* (IEEE, 1996), pp. 249–56 <<https://doi.org/10.1109/MHS.1996.563432>>.

³⁶ Robert G. Reynolds, 'An Introduction to Cultural Algorithms', in *Evolutionary Programming --- Proceedings of the Third Annual Conference*, ed. by Anthony Fogel, V. Sebald, and J. Lawrence, 1st edn (San Diego, CA, USA: World Scientific Press, 1994), pp. 131–39.

³⁷ W. Daniel Hillis, 'Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure', *Physica D: Nonlinear Phenomena*, 42.1–3 (1990), 228–34 <[https://doi.org/10.1016/0167-2789\(90\)90076-2](https://doi.org/10.1016/0167-2789(90)90076-2)>.

³⁸ N. Hansen and A. Ostermeier, 'Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation', in *Proceedings of IEEE International Conference on Evolutionary Computation* (IEEE, 1996), pp. 312–17 <<https://doi.org/10.1109/ICEC.1996.542381>>.

³⁹ R. Storn and K. Price, *Differential Evolution-A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces* (Berkeley, 1995).

⁴⁰ Shumeet Bajula, *Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning* (Pittsburgh, 1994).

⁴¹ Lawrence J. Fogel, Alvin J. Owens, and Michael J. Walsh, *Artificial Intelligence through Simulated Evolution*, 1st edn (Oxford: John Wiley & Sons, Inc., 1966).

⁴² H-P. Schwefel, *Kybernestische Evolution Als Strategie Der Experimente Forschung in Der Stromungstechnik* (Berlin, 1965).

⁴³ J.H. Holland, *Adaption in Natural and Artificial Systems*, 1st edn (Ann Arbor: University of Michigan Press, 1975).

⁴⁴ Gunter Dueck, 'New Optimization Heuristics', *Journal of Computational Physics*, 104.1 (1993), 86–92 <<https://doi.org/10.1006/jcph.1993.1010>>.

⁴⁵ C. Voudouris, 'Guided Local Search — an Illustrative Example in Function Optimisation', *BT Technology Journal*, 16.3 (1995), 46–50.

⁴⁶ John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, 1st edn (Cambridge: MIT Press, 1992).

⁴⁷ Thomas A Feo and Mauricio G.C Resende, 'A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem', *Operations Research Letters*, 8.2 (1989), 67–71 <[https://doi.org/10.1016/0167-6377\(89\)90002-3](https://doi.org/10.1016/0167-6377(89)90002-3)>.

⁴⁸ Olivier Martin, Steve W Otto, and Edward W Felten, 'Large-Step Markov Chains for the TSP Incorporating Local Search Heuristics', *Operations Research Letters*, 11.4 (1992), 219–24 <[https://doi.org/10.1016/0167-6377\(92\)90028-2](https://doi.org/10.1016/0167-6377(92)90028-2)>.

(optimización de enjambre de partículas)⁵⁰, SA (recocido simulado)⁵¹, SM (método de suavizado)⁵², SS (búsqueda de dispersión)⁵³, TA (aceptación de umbral)⁵⁴, TS (búsqueda tabú)⁵⁵ y VNS (búsqueda de vecindario variable)⁵⁶.

⁴⁹ Irène Charon and Olivier Hudry, 'The Noising Method: A New Method for Combinatorial Optimization', *Operations Research Letters*, 14.3 (1993), 133–37 <[https://doi.org/10.1016/0167-6377\(93\)90023-A](https://doi.org/10.1016/0167-6377(93)90023-A)>.

⁵⁰ J. Kennedy and R. Eberhart, 'Particle Swarm Optimization', in *Proceedings of ICNN'95 - International Conference on Neural Networks* (IEEE, 1995), IV, 1942–48 <<https://doi.org/10.1109/ICNN.1995.488968>>.

⁵¹ S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, 'Optimization by Simulated Annealing', *Science*, 220.4598 (1983), 671–80 <<https://doi.org/10.1126/science.220.4598.671>>.

⁵² Fred Glover and Claude McMillan, 'The General Employee Scheduling Problem. An Integration of MS and AI', *Computers & Operations Research*, 13.5 (1986), 563–73 <[https://doi.org/10.1016/0305-0548\(86\)90050-X](https://doi.org/10.1016/0305-0548(86)90050-X)>.

⁵³ Fred Glover, 'HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS', *Decision Sciences*, 8.1 (1977), 156–66 <<https://doi.org/10.1111/j.1540-5915.1977.tb01074.x>>.

⁵⁴ Gunter Dueck and Tobias Scheuer, 'Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing', *Journal of Computational Physics*, 90.1 (1990), 161–75 <[https://doi.org/10.1016/0021-9991\(90\)90201-B](https://doi.org/10.1016/0021-9991(90)90201-B)>.

⁵⁵ P. Hanse and Mladnovic M., 'Variable Neighborhood Decomposition Search', *Journal of Heuristics*, 7 (2001), 330–50.

⁵⁶ N. Mladenović and P. Hansen, 'Variable Neighborhood Search', *Computers & Operations Research*, 24.11 (1997), 1097–1100 <[https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)>.

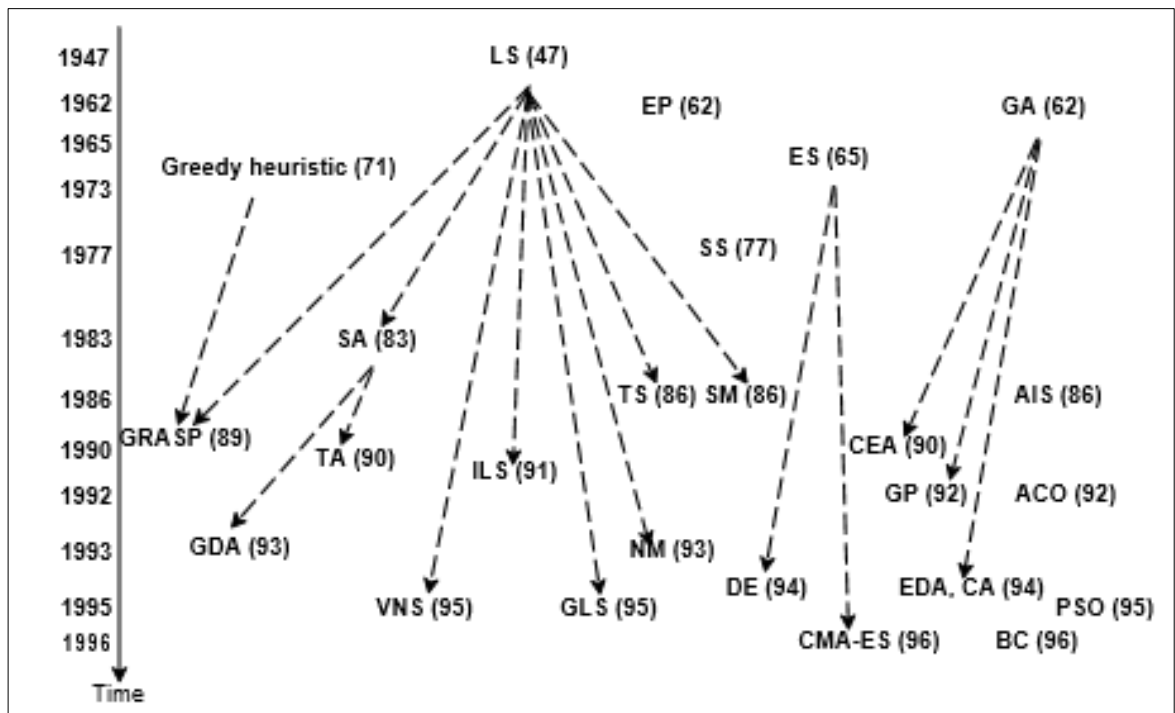


Figura 18. Genealogia de las metaheurísticas [3].

4. ALGORITMOS METAHEURISTICOS Y DISEÑO DE ESTRUCTURAS APORTICAS DE CONCRETO REFORZADO.

En los algoritmos Metaheurísticos, la exploración uniforme del espacio de búsqueda es de gran importancia para obtener soluciones satisfactorias. De aquí dará la importancia de la búsqueda aleatoria, ya que en este proceso es donde se seleccionan las respuestas buenas que conforman el espacio de búsqueda. Debido a que la selección se realiza en cada iteración, no es necesario utilizar memoria partiendo de que en cada repetición se selecciona la mejor opción vecina, por lo que en cada iteración se seleccionara una solución más favorable. El algoritmo de búsqueda que se implementa utiliza criterios de selección que se establecen en el diseño de la metaheurística. De manera que podemos nombrar algunas características generales de este tipo de técnicas. Se definen de manera general como reglas de alto nivel que utilizan algoritmos de búsqueda para problemas de optimización complejos. Tienen como objetivo fundamental realizar una inspección del espacio de búsqueda para encontrar soluciones casi óptimas. Se rigen por un comportamiento no determinista, estocástico. Presentan herramientas que evitan la incorporación de soluciones poco favorables al espacio de búsqueda. Se pueden obtener metaheurísticas a partir de heurísticas específicas⁵⁷.

Finalmente, en la figura 9 se puede observar un esquema básico de un algoritmo metaheurístico. De manera general, se entiende que este tipo de algoritmo

⁵⁷ Gustavo R. Zavala and others, 'A Survey of Multi-Objective Metaheuristics Applied to Structural Optimization', *Structural and Multidisciplinary Optimization*, 49.4 (2014), 537–58 <<https://doi.org/10.1007/s00158-013-0996-4>>.

aproximado utiliza estrategias de alto nivel para explorar eficientemente espacios de búsqueda.

```
1:  $A(0) \leftarrow \text{GenerarSolucionesIniciales}()$   
2:  $t \leftarrow 0$   
3: Evaluar ( $A(0)$ )  
4: While not CriteriodeParada() do  
5:  $S(t) \leftarrow \text{Generación}(A(t))$   
6: Evaluar( $S(t)$ )  
7:  $A(t+1) \leftarrow \text{Actualizar}(A(t), S(t))$   
8:  $t \leftarrow t + 1$   
9: end while
```

Figura 19. Modelo genérico de algoritmo metaheurístico [31].

Obsérvese en la figura 19, que se parte de un conjunto A de soluciones iniciales, el cual será modificado en cada una de las iteraciones creando un conjunto S de soluciones “buenas” hasta que se cumpla el criterio de parada.

A continuación, se presenta una clasificación de las metaheurísticas de acuerdo a la fuente de inspiración.

4.1. CLASIFICACIÓN DE LAS METAHEURÍSTICAS.

4.1.1. Basadas en la naturaleza. La creación de algoritmos de optimización mediante la imitación de distintos procesos naturales es lo que ha dado lugar a técnicas como los algoritmos evolutivos y sistemas inmunes artificiales de la biología, hormigas, colonias de abejas, enjambres, etc. lo que dio lugar a distintas metaheurísticas basadas en los procesos naturales. Así como las metaheurísticas que no están inspiradas en la naturaleza tabú la cual parte de una solución inicial la cual va variando hasta encontrar la respuesta óptima [38]. Obsérvese la tabla 2.

Tabla 2. Algoritmos metaheurísticos basados en la naturaleza [3].

Inspirados en la naturaleza.	No inspirados en la naturaleza.
• Recocido simulado (SA). • Inteligencia de enjambre. -Optimización de enjambre de partículas (PSO). -Optimización de colonias de hormigas (ACO). • Computación evolutiva (EC). -Algoritmos Genéticos (GA). -Estrategia de evolución (ES). -Programación Genética (GP). • Estimación de distribución.	• Búsqueda Tabú (TS). • búsqueda de vecindario variable(VNS). • Búsqueda de dispersión (SS). • Búsqueda Local Iterada (ILS). • Búsqueda Local Guiada (GLS). • Reconexión de la ruta (PR). • Procedimiento codicioso de búsqueda adaptativa aleatoria (GRASP).

-Algoritmo (EDA).	
-------------------	--

4.1.2. Sistemas con memoria y sin memoria. Existen distintos algoritmos metaheurísticos que almacenan información durante el proceso de búsqueda como el tabú. En contraste, existen otros algoritmos que no necesitan memoria para desarrollar su búsqueda, por ejemplo, la búsqueda local grasp. y el recocido simulado que consiste en buscar el equilibrio termodinámico variando la temperatura [1] ⁵⁸.

4.1.3. Deterministas y estocásticos. La gran diferencia data en las reglas utilizadas en las metaheurísticas. En las metaheurísticas deterministas se toman decisiones que son relativamente predecibles, como se presenta en la búsqueda total y en el tabú ya que son algoritmos que almacenan información en cada iteración. En contraste los algoritmos estocásticos o no deterministas utilizan métodos aleatorios que no permiten una predicción en la selección de respuestas, por ejemplo los algoritmos genéticos con los cuales se pueden obtener diferentes respuestas ⁵⁹.

⁵⁸ Warren Hare, Julie Nutini, and Solomon Tesfamariam, 'A Survey of Non-Gradient Optimization Methods in Structural Engineering', *Advances in Engineering Software*, 59 (2013), 19–28 <<https://doi.org/10.1016/j.advengsoft.2013.03.001>>.

⁵⁹ Luis Humberto Niño Álvarez and Eder Jair Durán Cáceres, 'Optimización Del Dimensionamiento Estructural de Un Portico Plano de Concreto Reforzado, Destinado a Almacenamiento de Archivos (Trabajo de Grado)' (Universidad Industrial de Santander- Escuela de Ingeniería Civil, 2016).

4.1.4. Búsqueda basada en población y búsqueda basada en una sola solución. Se parte de que en la búsqueda basada en una sola solución (búsqueda local, recocido simulado, tabú), la solución inicial es modificada para explotar el espacio de búsqueda variando la respuesta inicial en cada iteración. En contraste, la búsqueda basada en una población (enjambre de partículas, algoritmos genéticos) genera un conjunto de soluciones que pueden o no ser “buenas”. En efecto, los algoritmos basados en una única solución tienen una inclinación por la intensificación de espacios locales, a diferencia de los algoritmos poblacionales que utilizan la diversificación en el espacio de búsqueda mejorando la búsqueda de un óptimo global ⁶⁰.

4.1.5. Iterativo y codicioso. Su principal disimilitud son las condiciones de la solución inicial. En los algoritmos iterativos se parte de una solución completa es decir de un conjunto de soluciones y en cada iteración se transforma este conjunto con operadores de bajo nivel como los algoritmos genéticos en el cual se puede definir el número de individuos del conjunto de soluciones según el tipo de problema. En los codiciosos el conjunto de soluciones es vacío y en cada paso se adiciona una variable de operación hasta obtener una solución completa. Por lo general los algoritmos metaheurísticos son iterativos ⁶¹.

4.2. CONCEPTOS COMUNES PARA LAS METAHEURÍSTICAS.

Al iniciar el proceso de diseño de una metaheurística es necesario plantear dos incógnitas que son: el tipo de representación que tendrán las soluciones y la

⁶⁰ Lee and Ahn.

⁶¹ Gustavo Zavala and others, ‘Structural Design Using Multi-Objective Metaheuristics. Comparative Study and Application to a Real-World Problem’, *Structural and Multidisciplinary Optimization*, 53.3 (2016), 545–66 <<https://doi.org/10.1007/s00158-015-1291-3>>.

función objetivo a la que se quiere llegar. En esta subsección se profundizará en el planteamiento de estos interrogantes.

4.2.1. Representación. Se considera como un paso fundamental en el proceso de diseño de la metaheurística definir la representación de las soluciones que maneja el algoritmo, ya que esto puede interferir con la eficiencia y eficacia del mismo. La representación o codificación debe ser compatible con los operadores de búsqueda que se quieren utilizar, es decir, la codificación de una solución puede ser útil para el método de vecindario, pero no para el recombinatorio. Es por esto que se recomienda definir antes la representación de la solución y posteriormente seleccionar los métodos de evaluación y los operadores de búsqueda. A pesar de que un problema puede tener distintas representaciones para sus soluciones, siempre deben mantener las siguientes características ⁶².

- Integridad, se debe garantizar que todas las posibles soluciones estén representadas con la misma codificación.
- Conexión: Esta característica debe garantizar que los operadores de búsqueda tengan diversidad, es decir, que puedan evaluar a cada una de las soluciones que conformen el conjunto de soluciones inicial.
- Eficiencia: Algunos operadores de búsqueda modifican las soluciones, por lo que deben ser fáciles de modificar, esto con el objetivo de disminuir las complejidades de tiempo y espacio.

⁶² Franklin and Truman Kevin.

4.2.2. Función objetivo. La función objetivo es la formulación de lo que se quiere optimizar. Mediante la evaluación de esta función se encuentra el valor idóneo de calidad o cantidad que se quiere encontrar. Debido a que la función $F: S \rightarrow R$, se obtiene como resultado un número real absoluto que puede tomar cualquier solución. Como se planteó anteriormente las soluciones contenidas en el conjunto R pueden estar sujetas a modificaciones d , $d: R \rightarrow S$, para generar funciones que puedan ser evaluadas, es decir las d codifican las soluciones. La función objetivo es de máxima importancia, si esta es mal definida las soluciones “buenas” no conducirán a respuestas favorables en el problema de optimización. Se puede definir como la guía de las d para encontrar soluciones viables en el conjunto R .⁶³ Existen distintos conceptos necesarios para llevar a cabo la formulación de funciones objetivo, por ejemplo:

- **Funciones objetivo autosuficientes:** Cuando la formulación de la función objetiva es sencilla, es decir, cuando la fórmula guía (con la que se plantea el problema) es igual que la función objetivo se dice que la función es autosuficiente. Esto se presenta generalmente en problemas de optimización continuos (lineales y no lineales).
- **Objetivo orientador:** Este concepto indica que la función objetivo debe cumplir con el mayor número de condiciones o cláusulas necesarias para que la guía de la búsqueda sea más eficiente. Por lo tanto, durante la formulación de la función objetivo se debe buscar maximizar el cumplimiento de las restricciones planteadas por el problema.

⁶³ Franklin and Truman Kevin.

- Decodificación de representación: Se debe definir de manera adecuada la representación de las soluciones, esto se relaciona de manera directa con la formulación de la función objetivo. Esto se debe a la relación que existe entre una representación adecuada para obtener soluciones factibles. En efecto, es necesario que la solución objetivo pueda ser codificada fácilmente para garantizar la eficacia de la metaheurística, un ejemplo de esto es la representación binaria ⁶⁴.
- Optimización interactiva. Se da cuando el usuario hace parte activa de los procesos iterativos de la metaheurística. Por lo general, se diseñan metaheurísticas interactivas cuando el usuario aporta términos importantes a la hora de realizar la búsqueda, esta interacción se realiza mediante la representación gráfica de las soluciones, esto en busca de que se dé una convergencia en un menor tiempo. En otros casos es necesaria debido a que la evaluación de la función objetivo no se puede realizar cualitativamente, por ejemplo, el sabor del café, por lo tanto, el usuario realiza manualmente la evaluación de la solución obtenida ⁶⁵.
- Funciones objetivas y competitivas: En ocasiones el valor absoluto obtenido no puede converger con todas las soluciones, esto se presenta comúnmente en la teoría de juegos, los sistemas de clasificación de aprendizaje o la coevaluación cooperativa o competitiva donde la estrategia A es mejor que la B, la B mejor que la C y la C mejor que la A. Para estos casos los investigadores presentan dos tipos de soluciones, usar funciones objetivo relativas o competitivas. En la función relativa se aplica la verdad o la falsedad parcial, es decir, la función objetivo es solo asocia un rango parcial de las soluciones iniciales. Respecto a

⁶⁴ Niño Alvarez and Durán Cáceres.

⁶⁵ Talbi.

la función competitiva se lleva a cabo una competencia sobre una población de soluciones. Existen tres tipos de competencia: torneo, bipartita y torneo. En la competencia bipartita se comparan dos posibles soluciones, mientras que en la completa se comprara la población completa, de esta manera se determina la más viable. Este tipo de funciones objetivos se utilizan comúnmente en algoritmos metaheurísticos basados en la población, ya que permiten la evolución de la población sin necesidad de encontrar un valor absoluto para todas las soluciones ⁶⁶.

- Metodologías de superficie de respuesta: Se utiliza para encontrar la función objetivo mediante la aproximación polinomial utilizando técnicas como los mínimos cuadrados de Box y Wilson y el diseño de experimentos (DOE) de Taguchi.

Finalmente, se pueden utilizar otro tipo de técnicas para formular la función objetivo como lo son los modelos de Kriging, diseño y análisis de experimentos informáticos, procesos gaussianos y aprendizaje automático como SVM (máquinas de vectores de soporte) utilizados ampliamente en procesos industriales ⁶⁷.

4.3. DESCRIPCIÓN DE LA METAHEURÍSTICAS MÁS UTILIZADAS.

En esta subsección se describen algunas de las metaheurísticas más utilizadas en problemas de optimización.

⁶⁶ Duysinx and Tossings.

⁶⁷ Gecheng Chen and Zhiqiang Ge, 'SVM-Tree and SVM-Forest Algorithms for Imbalanced Fault Classification in Industrial Processes', *IFAC Journal of Systems and Control*, 8 (2019), 100052 <<https://doi.org/10.1016/j.ifacsc.2019.100052>>.

4.3.1. Recocido simulado (SA). Esta técnica imita el proceso de recocido de los metales, que básicamente es aumentar la temperatura de un metal hasta un punto de alta energía, esto permite que las moléculas que conforman el material se desplacen libremente, posteriormente se disminuye la temperatura gradualmente hasta que el metal alcance un estado cristalino, un punto de mínimo absoluto de energía.

Durante este proceso, al aumentar la temperatura las moléculas que conforman el material se desplazan de manera aleatoria y a medida que las temperaturas bajas se reacomodan, y con suerte este estado final tendrá un estado de menor energía que el inicial. Al realizar la analogía con las metaheurísticas se supone que las moléculas iniciales son las soluciones en un mínimo local, que mediante el aumento de energía formara un conjunto nuevo de soluciones aleatorias. Las soluciones serán aprobadas a medida que un índice (temperatura) disminuya a medida que aumenten las iteraciones. Esto permite aceptar solo las soluciones con un valor objetivo más alto, lo que evitara parar en mínimos locales ⁶⁸.

Aun así, el proceso se describe de manera general en 3 pasos, determinación de temperatura, transformación de las soluciones o moléculas y búsqueda del equilibrio termodinámico. Se parte de una estructura o colocación dada, la cual sufre transformaciones (intercambiar componentes), posteriormente se realizan iteraciones, siempre buscando el equilibrio termodinámico. Todo esto con una temperatura constante. Además, se debe observar una disminución en la energía (función objetivo). De acuerdo a las condiciones que se indique variara la temperatura cuando la energía aumente después de un número suficiente de iteraciones. Estas condiciones se definen empíricamente por el diseñador, que indirectamente definen el criterio de parada del programa.

⁶⁸ Hare, Nutini, and Tesfamariam.

La eficiencia de este tipo de técnica depende en gran medida de la rapidez con que se baje la temperatura, ya que puede pasar de ser recocido a temple. Véase la figura 20.

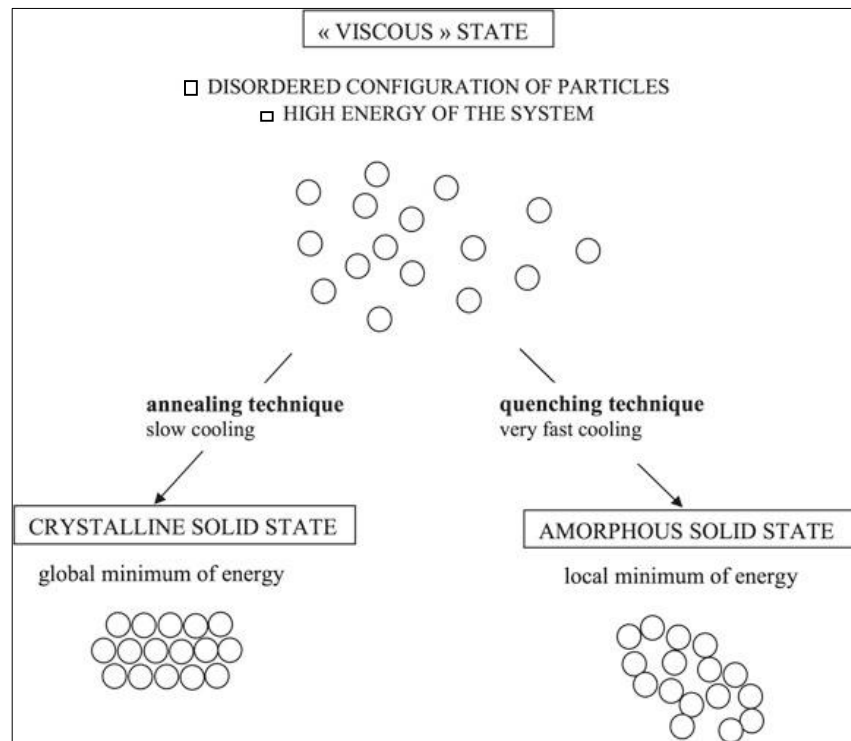
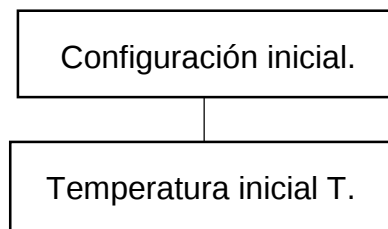


Figura 20. Comparación entre la técnica de recocido (annealing) y templeado (quenching) [1].

En la figura anterior se observa que si el descenso de temperatura se da de manera rápida puede obtenerse un mínimo local de energía, por lo que el número de iteraciones deben ser las suficientes para evitar estos mínimos locales.

En el flujograma mostrado en la figura 21 el algoritmo comienza con un estado elegido al azar y una temperatura inicial. Luego, en cada iteración, se genera un

nuevo candidato a través de una función de vecindad teniendo en cuenta la temperatura del sistema, el candidato anterior y las características intrínsecas del espacio de solución. Cada nuevo estado propuesto es aceptado o rechazado de acuerdo con un criterio, siempre se acepta una propuesta mejor; una peor solo se acepta con una probabilidad que depende de la temperatura del sistema y de la transición energética entre el estado anterior y el actual. Luego, la temperatura se reduce mediante una función decreciente y se realiza una comprobación para ver si el cambio promedio en la función objetivo después de las iteraciones de está por debajo de un umbral. En caso de estancamiento, el algoritmo desencadena una búsqueda local con el candidato actual como suposición inicial. En cualquier caso, si no se cumple el criterio de detención, se genera un nuevo candidato, reiniciando efectivamente el ciclo de recocido ⁶⁹.



⁶⁹ Carlos Loucera, Andres Iglesias, and Gálvez Akemi, 'Simulated Annealing and Natural Neighbor for Rational Bézier Surface Reconstruction from Scattered Data Points', in *International Conference on Harmony Search Algorithm* (Bilbao: Javier del Ser, 2017), pp. 354–63.

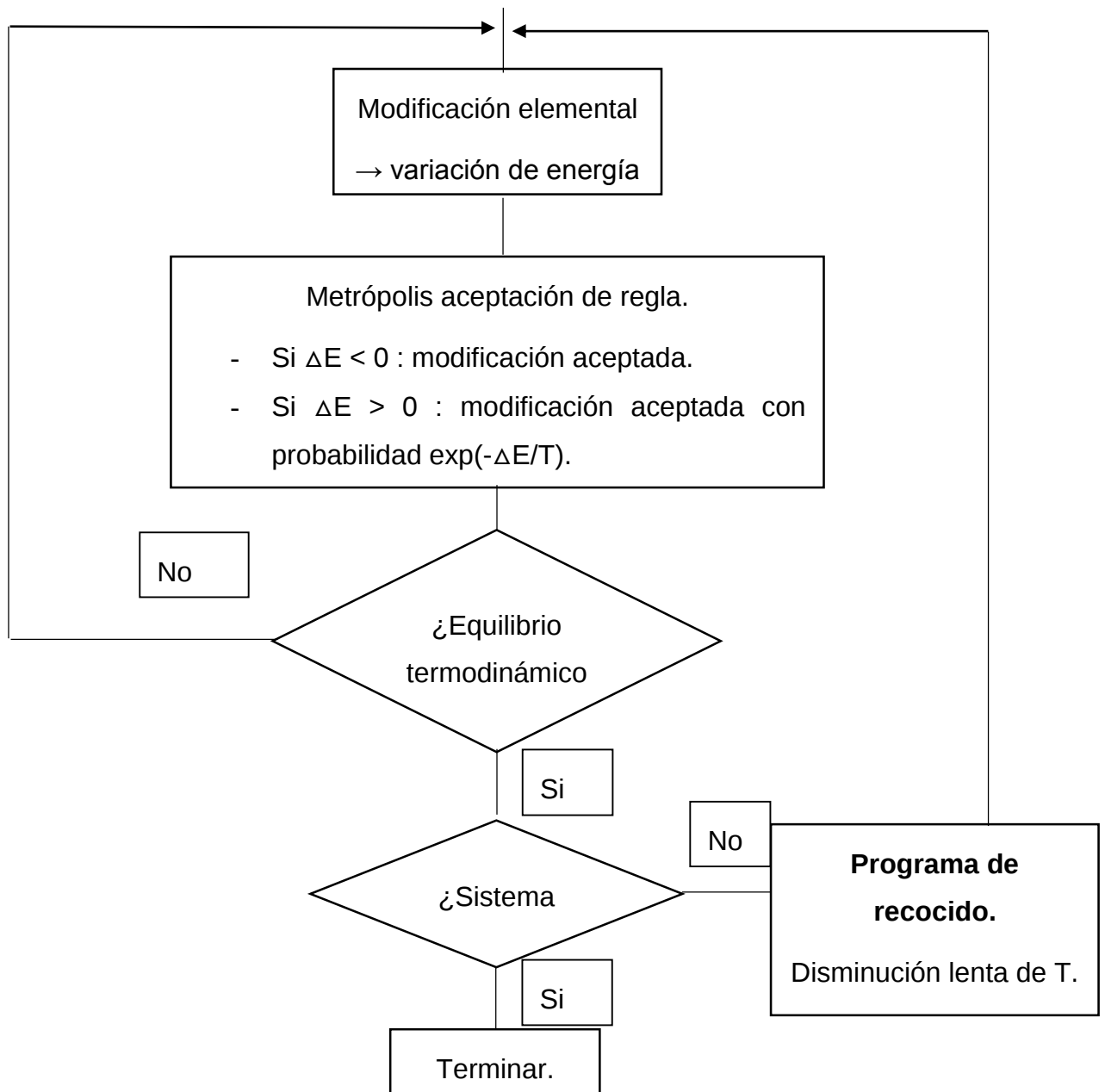


Figura 21. Diagrama de flujo básico técnica de recocido [1].

4.3.2. El método de búsqueda tabú (TS). Fue utilizado formalmente por Glover en 1986. Su funcionamiento trata de emular el mecanismo de la memoria humana. Por lo tanto, este método utiliza la memoria como factor fundamental en su operación. A diferencia del algoritmo expuesto anteriormente la introducción de condiciones por parte del usuario se dificulta debido a que la memoria introduce múltiples grados de libertad. Aun así, el algoritmo funciona de manera similar al recocido, inicia con una configuración actual S , que será modificada a medida que pasan las iteraciones para transformarse en una nueva solución t . Este proceso se lleva a cabo en 2 etapas ⁷⁰.

La primera consiste en la creación del conjunto de configuraciones iniciales S , donde se debe procurar que esté compuesto por un número finito para que puede ser alcanzado por un movimiento elemental. Cuando el conjunto es demasiado grande se puede presentar una lista de candidatos o bien formar un subconjunto $V(S)$ de vecinos de tamaño fijo. De esta manera se evalúa la función objetivo con cada uno de los componentes de $V(S)$ ⁷¹.

Para seleccionar la configuración t es necesario evaluar la función objetivo f en todas las configuraciones de $V(S)$, la que tome el valor mínimo será la seleccionada, aún si t es peor que S , es decir aún si $f(t) > f(S)$, esto contribuye a que el algoritmo no localice mínimos locales. El problema data en que existe el riesgo de que en cada iteración se regrese a una solución seleccionada anteriormente. Para evitar este problema es necesario que en cada repetición se actualice una lista de movimientos prohibidos, la “lista tabú”. Este conjunto m almacena los últimos $f(t)$ de cada iteración limitando la aparición de este en una

⁷⁰ Ré-Mi Hage and others, ‘Optimized Tabu Search Estimation of Wear Characteristics and Cutting Forces in Compact Core Drilling of Basalt Rock Using PCD Tool Inserts’, *Computers & Industrial Engineering*, 2019 <<https://doi.org/10.1016/j.cie.2019.07.049>>.

⁷¹ M. P. Saka, O. Hasançebi, and Z. W. Geem, ‘Metaheuristics in Structural Optimization and Discussions on Harmony Search Algorithm’, *Swarm and Evolutionary Computation*, 28 (2016), 88–97 <<https://doi.org/10.1016/j.swevo.2016.01.005>>.

iteración siguiente. A menudo también se implementan procesos de intensificación y exploración con el objetivo de actualizar la lista durante un lapso de tiempo mayor. Esto es posible gracias a que ya no se actualiza de acuerdo a la última iteración, sino que tiene en cuenta la frecuencia de selección de la configuración. Aunque esto da gran ventaja la complejidad que se agrega es notable. A continuación, se observa el modelo de un “tabú simple”, véase la figura 22. En la figura 23, se observa el flujograma del mismo ⁷².

⁷² Saka, Hasançebi, and Geem.

```

1: procedimiento BusquedaTabú
2: Inicio.
3: Seleccionar un inicial  $x \in K$ .
4:  $T := \frac{1}{4}$ ;  $K := 0$ ;
5: Inicio
6: if  $S(x) - T$  is empty
7:   stop;
8: else
9:   Set  $k := k+1$ ;
10:  Select  $s_k \in S(x)-T \square s_k(x) = \text{ÓPTIMO}(S(x))$ ;
11:   $s := s_k$ ;
12: end
13: Let  $x := s$ ;
14: if  $f(x) < f(x^*)$ 
15:   $x^* := x$ ;
16: end
17: revisar condiciones de parada;
18: actualizar  $T$ ;
19: end
20: end

```

Figura 22. Estructura básica de algoritmo “tabú simple” [33].

Se encontró que la búsqueda tabú funciona de una mejor manera, cuando existe aumento en el espacio variable, ya que encuentra diseños cortos y factibles en todos los casos. Es importante recalcar que este método ha sido utilizado para

optimizar del diseño de estructuras de acero, el peso estructural de los marcos y para evaluar el rendimiento sísmico de estructuras de marcos optimizados ⁷³.

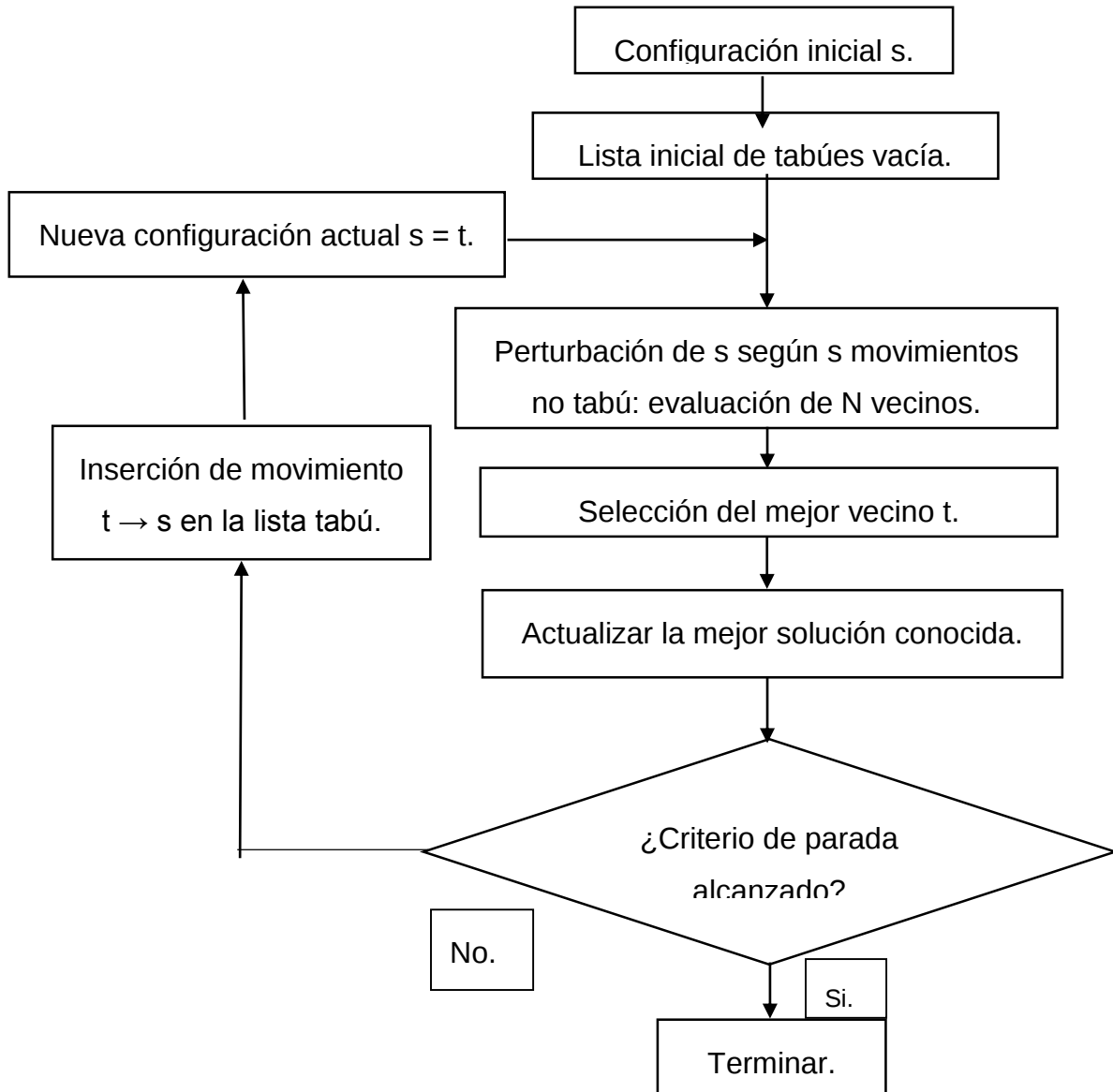


Figura 23. Diagrama de flujo de algoritmo tabú simple [1].

⁷³ Hage and others.

4.3.3. Algoritmos genéticos (GA) y algoritmos evolutivos (EA). Los algoritmos genéticos son una técnica de optimización heurística, que utiliza fundamentos matemáticos para imitar la selección natural desarrollada por John Holland. Su funcionamiento se basa en la transformación de un conjunto (población) de objetos matemáticos moldeados por los cromosomas. En los algoritmos genéticos se selecciona una población de acuerdo al principio darwiniano de reproducción y supervivencia del más apto, después de realizarse una recombinación.

Para entender de mejor manera el concepto de algoritmo genético es necesario tener en cuenta las siguientes definiciones. El cromosoma es una serie de datos que contienen instrucciones que también son llamados genes. Estos datos pueden ser almacenados como una cadena en código binario o bien trabajar con matrices. Un gen es una sección del cromosoma que codifica un parámetro en específico. El alelo es el valor que toma el gen. Suponiendo que un gen está compuesto por una cadena binaria se dice que puede tener dos alelos 0 o 1, Un esquema es un patrón que puede tomar los genes que conforman un cromosoma. Por lo que un cromosoma hecho por una cadena de genes binarios tendrá la longitud de la cadena por valores formados por {0,1}. En la figura 24 se observa el modelo básico de un algoritmo genético simple ⁷⁴.

⁷⁴ C. A. Coello Coello, A. D. Christiansen, and F. Santos Hernández, 'A Simple Genetic Algorithm for the Design of Reinforced Concrete Beams', *Engineering with Computers*, 13.4 (1997), 185–96 <<https://doi.org/10.1007/BF01200046>>.

```
1: procedimiento EstrategiaEvolutiva
2: Inicializar población de padres B.
3: while (no terminación de la condición) do
4:   for n= 1:k do begin
5:     Cn= Reproducción (αB);
6:     Sn= s.recombinación(Cn);
7:     Yn=y.recombinación(Cn);
8:     S=¼s.mutación(Sn);
9:   end
10:  actualizar B;
11:  Realizar selección y actualizar población de
    padres B
12:end
13: end
```

Figura 24. Modelo básico algoritmo genético simple [30].

Respecto a los métodos evolutivos, era poco común su uso debido a que el coste computacional era alto. Pero debido a los avances en las ciencias de la tecnología

las investigaciones realizadas con este tipo de técnicas están tomando un interés especial. El funcionamiento de un algoritmo genético se puede describir de la siguiente manera: Se inicia con un conjunto de puntos en el espacio seleccionados aleatoriamente; cada punto que conforma el conjunto tiene un desempeño frente a la adaptación necesaria de la función objetivo. Suponiendo que se busca una minimización de la función objetivo f , partiendo de esto el valor de x que constituya el $f(x)$ de menor valor será un sobreviviente. Este conjunto de puntos que conforman la población evolucionara gradualmente con un tamaño constante. Esto para que el rendimiento general de la población mejore. Estos métodos se basan fundamentalmente por la creación de una población que representa posibles soluciones que se combinan y compiten entre sí, buscando el individuo más apto. Los resultados obtenidos por este algoritmo presentan un error relativo inferior al 10^{-5} respecto a los valores experimentales, validando los algoritmos evolutivos como una buena opción para problemas de ajuste computacionales costosos ⁷⁵.

La evolución de una población a la siguiente se lleva a cabo en cuatro fases importantes: fase de selección, fase de variación o reproducción, una fase de valoración de desempeño y una fase de sustitución. Como se mencionó anteriormente la población inicial se selecciona arbitrariamente procurando escoger los individuos más potentes. Estos individuos estarán disponibles para la variación, esta etapa consiste en aplicar operadores de variación a las copias seleccionadas para generar nuevos individuos. En la figura 25 y 26 se pueden observar algunos de los operadores más utilizados ⁷⁶.

⁷⁵ A. M. Hernández-Díaz, J. M. Cecilia, and M. D. García-Román, 'Ajuste de Modelos de Cortante En Hormigón Estructural Mediante Algoritmos Evolutivos', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 33.1–2 (2017), 52–57 <<https://doi.org/10.1016/j.rimni.2015.10.001>>.

⁷⁶ Kar.

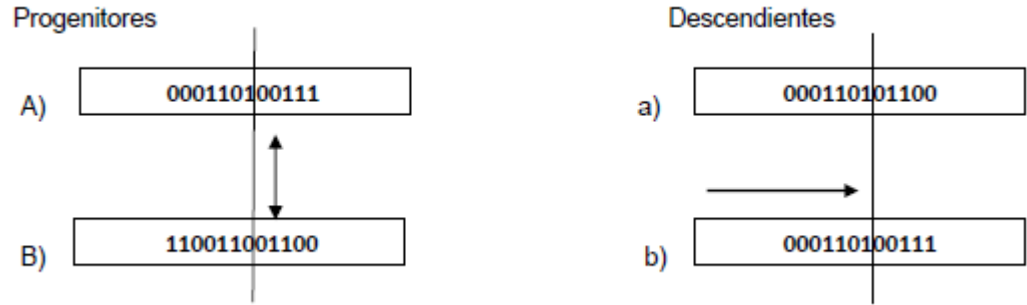


Figura 25. Diagrama de cruzamiento de único punto [36].

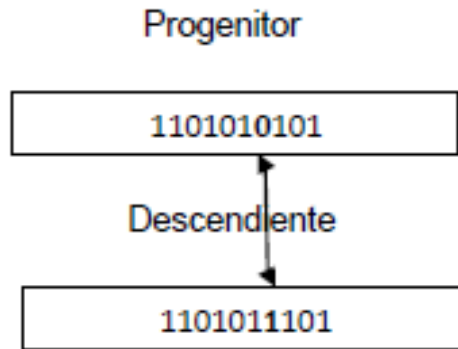


Figura 26. Diagrama de ejemplo de mutación [36].

Los cruces consisten en producir dos descendientes a partir de dos progenitores, en este caso solo se intercambia un segmento de la cadena de datos. En contraste, la mutación un nuevo individuo a partir de la variación de bit que confirma la cadena de datos. Las estructuras de estos operadores dependen del tipo de representación que tengan los individuos. Una vez terminada la etapa de reproducción, se procede con la evaluación, mediante la evaluación de estos objetivos en la función objetivo. Finalmente se seleccionan los individuos que conformaran la nueva población, a menudo se reemplazan los individuos con

menos potencial. El algoritmo toma como criterio de parada el número de generaciones que se determinan inicialmente. Finalmente se obtiene una población con individuos óptimos, debido a esto los algoritmos evolutivos son muy útiles para los problemas con varios óptimos globales. Esto puede generar problemas cuando los óptimos cumplen funciones completamente opuestas. En la figura 27 se observa un esquema que resume el funcionamiento de un algoritmo evolutivo básico.

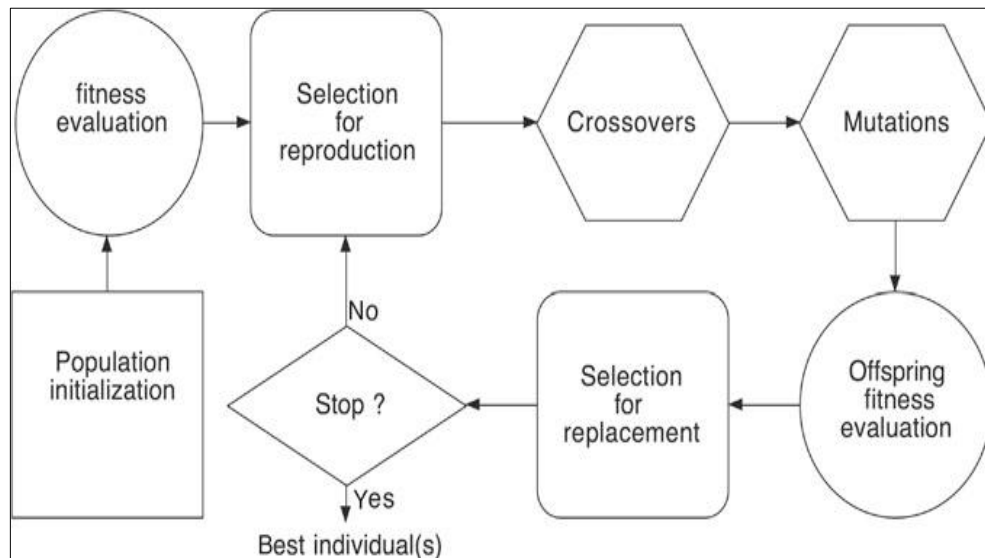


Figura 27. Esquema representativo del funcionamiento de un algoritmo evolutivo básico [35].

Este tipo de algoritmos son ampliamente utilizados en muchas áreas de la ingeniería. Ejemplo de esto es que son aplicando en la ingeniería estructural, como en confiabilidad estructural, diseño de puentes, estructuras, mantenimiento y reparación, diseño de puentes de vida de placa de acero soldados, zonificación sísmica, diseño sísmico de sistemas de la línea de vida, optimización de la estructura de armadura, el tamaño, la forma y la topología de las estructuras

esqueléticas y la optimización del diseño de las estructuras de acero, edificios de losas planas de hormigón, postes de telecomunicación de acero y amortiguadores viscosos.

```
1:  $P(0) \leftarrow \text{GeneranpoblaciónInicial}()$ 
2:  $t \leftarrow 0$ 
3:  $\text{Evaluar}(P(0))$ 
4: while not  $\text{criteriodeparada}()$  do
5:  $P'(t) \leftarrow \text{selección}(P(t))$ 
6:  $P''(t) \leftarrow \text{Reproducción}(P'(t))$ 
7:  $\text{Evaluar}(P''(t))$ 
8:  $P(t+1) \leftarrow \text{Reemplazar}(P(t), P''(t))$ 
9:  $t \leftarrow t+1$ 
10: end while
```

Figura 28. Algoritmo evolutivo más utilizado en las metaheurísticas [30].

En la figura 28, se observa el algoritmo evolutivo más utilizado en las metaheurísticas. Las soluciones candidatas se consideran individuos, que están compuestas por los cromosomas, cada uno tendrá un indicador de desempeño inicial. El conjunto de individuos o soluciones se denomina población. Como se describió anteriormente, algunos de los individuos serán seleccionados para la reproducción mediante la selección natural. Estos individuos seleccionados generan algunos hijos que reemplazarán a los individuos menos aptos. En cada

iteración se evalúan las soluciones para observar la aptitud de la población. El proceso termina cuando se cumple una condición específica o se llevan a cabo un número determinado de generaciones ⁷⁷.

Este tipo de algoritmos presentan algunas fortalezas y debilidades. Su principal desventaja es que no se ha realizado un modelamiento matemático que logre la convergencia de este tipo de algoritmos. Debido a esto la población de soluciones obtenidas no garantiza que sean globalmente óptimas. Aun así, si se concluye que los individuos de la población final son “buenos” o tienen un rendimiento alto.

En perspectiva, los algoritmos evolutivos proporcionan soluciones de alta calidad, en un corto lapso de tiempo, son efectivos especialmente en problemas de gran escala.

4.3.4. Algoritmos de colonias de hormigas (ACO). Durante los últimos años los algoritmos metaheurísticos han utilizado la imitación de organizaciones naturales como lo pueden ser las colonias de hormigas, enjambres de partículas, saltos de rana y colonias de abejas artificiales. Estos algoritmos son utilizados frecuentemente en los modelos de ingeniería estructuras debido a que utilizan reglas simples que logran un comportamiento de sistema inteligente ⁷⁸.

La primera aplicación de colonias artificiales de hormigas para la resolución de funciones continuas la realizaron Bilchev y Parmee ⁷⁹. Estos trabajos solo

⁷⁷ Hare, Nutini, and Tesfamariam.

⁷⁸ Hare, Nutini, and Tesfamariam.

⁷⁹ G. Bilchev and I. C. Parmee, 'The Ant Colony Metaphor for Searching Continuous Design Spaces', 1995, pp. 25–39 <https://doi.org/10.1007/3-540-60469-3_22>.

utilizaron procedimientos de búsqueda local como parte del algoritmo ACO. Posteriormente, Wodrich y Bilchev ⁸¹ extendieron estas estrategias para la resolución de problemas de optimización global mediante la inclusión del concepto de hormigas «locales y globales». Las hormigas «globales» usan mecanismos de caminata aleatoria y difusión de rastro de feromonas, mientras que las hormigas «locales» tienen la capacidad de percibir el rastro de feromonas que han depositado las hormigas «globales» ⁸².

Se presenta un caso particular en las colonias de hormigas, cuando ellas van por comida fuera del hormiguero se observa una situación particular. Todas las hormigas toman el mismo camino, es importante recalcar que el camino siempre es el más corto posible y que las hormigas se comunican a través del entorno, un modo conocido como “estigmergia”. Esto es posible debido a que las hormigas secretan feromonas que sus compañeras logran identificar y seguir. Esto resulta una habilidad colectiva para encontrar la ruta más corta cuando la ruta original es obstaculizada. Este algoritmo se resume en tres procesos construcción de la solución basada en “decisiones tipo hormiga”, actualización de feromonas (intensificación y evaporación) y acciones colectivas ⁸³.

En la figura 29 se observan los pasos que realizan las hormigas. Las hormigas siguen un camino entre el hormiguero y una fuente de alimento, cuando el camino

⁸⁰ Y.J. Feng, L. Yu, and G.L. Zhang, ‘Ant Colony Pattern Search Algorithms for Unconstrained and Bound Constrained Optimization’, *Applied Mathematics and Computation*, 191.1 (2007), 42–56 <<https://doi.org/10.1016/j.amc.2006.09.142>>.

⁸¹ M. Wodrich and C. Bilchev, ‘Cooperative Distributed Search: The Ant’s Way’, 2000.

⁸² J. A. Fernández-Vargas and A. Bonilla-Petriciolet, ‘Desarrollo de Un Algoritmo de Optimización Global En Colonias de Hormigas Con Selección de Región Factible Para Espacios Continuos’, *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 30.3 (2014), 178–87 <<https://doi.org/10.1016/j.rimni.2013.06.006>>.

⁸³ Fernández-Vargas and Bonilla-Petriciolet.

es obstaculizado las hormigas toman para la derecha e izquierda con las mismas posibilidades, en el camino más corto se secreta la feromona en menor tiempo comparado con el camino largo.

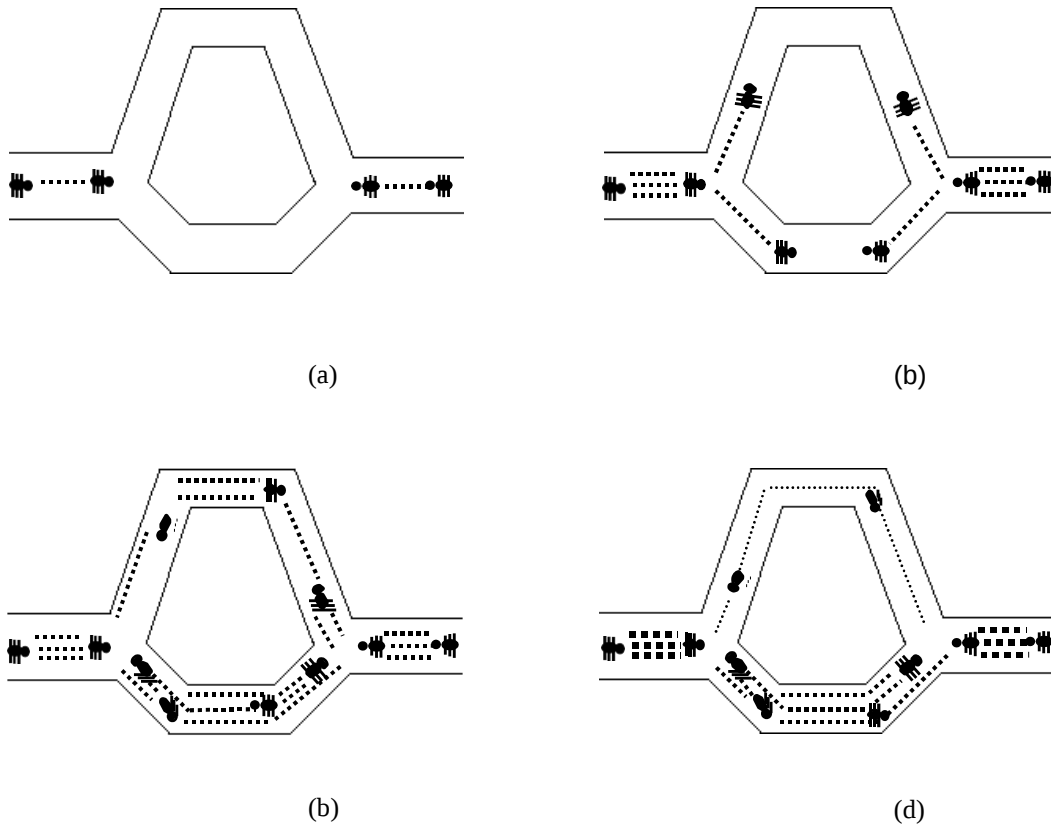


Figura 29. Comportamiento de las hormigas Reales [50].

Este tipo de algoritmo tiene algunas características particulares como los son un alto paralelismo intrínseco, flexibilidad (definiéndose como la capacidad de adaptarse rápidamente a un nuevo entorno), robustez (capacidad de funcionar, aunque algunos elementos falten o fallen), descentralización (una colonia no obedece una autoridad centralizada) y autoorganización (una colonia encuentra

colectivamente las soluciones a problemas nuevos). Estos algoritmos son muy útiles para problemas que deben tener una alta tolerancia a fallas.

De manera general este algoritmo utiliza a los individuos como soluciones y mediante la interacción con el entorno y entre sus individuos se obtiene una optimización numérica. En efecto, se utiliza la imitación del mecanismo que utilizan las colonias de hormigas para encontrar el camino más corto entre su nido y la comida, para encontrar la solución a un problema de optimización ⁸⁴.

Para entender de mejor manera los algoritmos de colonias de hormigas es necesario tener en cuenta distintos conceptos. Matemáticamente una gráfica es definida como un conjunto de nodos (o vértices) denominados i . Las líneas que interconectan estos nodos entre si se denomina (i, j) , que también son llamados bordes. Finalmente, el algoritmo busca la ruta más corta para llegar de un nodo determinado a otro. En el tiempo $t=0$ se inicializa el movimiento de las hormigas a través de los nodos $i(t)$ y los bordes (i, j) . Las hormigas se mueven de un lugar a otro basados en una función de probabilidad de dos “medidas de conveniencia”, intensidades de trayectoria de feromonas y visibilidad. Un camino con alta intensidad de feromonas indica que varias hormigas han recorrido esa trayectoria lo que indica que es un camino favorable. Respecto a la visibilidad, indica el grado de proximidad a las trayectorias más favorables. Este ciclo se repite hasta que cada una de las hormigas siga el mismo camino. En figura 30 se observa la estructura básica de este tipo de algoritmo ⁸⁵.

⁸⁴ Saka, Hasançebi, and Geem.

⁸⁵ Hare, Nutini, and Tesfamariam.

```
1: procedimiento ColoniadeHormigas
2: begin
3: Para cada borde (i, j), inicialice  $S_u(t)$ ; Colocar m
   hormigas en n nodos.
4: while not (no condición de terminación) do
   begin
5: for s=1, ..., n 1
6:     for k=1,..., m
7: Mover  $k^{th}$  hormiga a nodo j usando función de
   probabilidad  $p^k$ .
8: end end
9: Mueve cada hormiga al nodo inicial
   correspondiente; Calcule la longitud de cada
   recorrido del recorrido,  $L_k$ ;
10: Actualizar camino más corto;
11: Actualizar ruta de feromonas;
12: end end
```

Figura 30. Estructura general de un algoritmo de colonia de hormigas [33].

4.3.5. Algoritmo de optimización de enjambre de partículas (PSO). Este algoritmo basa su funcionamiento en la imitación de un colectivo de organismos que realizan procesos naturales tales como la escolarización de peces, la caza de aves o la coordinación de un enjambre de insectos, todos siguiendo un objetivo que se logra mediante la retroalimentación de los miembros. Es necesario definir la palabra enjambre, que, para este caso, es un grupo de individuos descentralizados y homogéneos que interactúan localmente entre ellos y el entorno, siempre buscando un objetivo común, es decir, una función objetivo. Este tipo de método es sumamente útil para problemas en los cuales existen múltiples parámetros relacionados de manera no lineal, donde la función objetivo es discontinua, no diferencial. Dependiendo del organismo que se trata de imitar, el número de pasos iterativos del algoritmo variara.

El fundamento es relativamente sencillo, cada uno de los individuos puede detectar en cualquier momento una solución potencial para el problema, mediante un indicador de desempeño o idoneidad clasifica la solución frente al resto de individuos del enjambre. Cada individuo enviará una señal a los demás individuos que será ajustada dependiendo de la idoneidad de la solución encontrada ⁸⁶. Cuando las partículas (individuos) encuentran una solución potencialmente optima, basadas en algoritmos de aprendizaje y cumpliendo distintas condiciones, inicia una fluencia inicial para guiar gradualmente a todas las partículas hacia una solución óptima global ⁸⁷⁸⁸.

Esta herramienta utiliza distintos mecanismos basados en números reales para evitar caer en mínimos locales, lo que proporciona una optimización menos

⁸⁶ Amir Hossein Gandom and others, *Metaheuristic Applications in Structures and Infrastructures*, 1st edn (London: Elsevier, 2013) <<https://doi.org/10.1016/C2011-0-08778-1>>.

⁸⁷ X.S. Yang and others, *Swarm Intelligence and Bio-Inspired Computation*, ed. by 1 (Elsevier, 2013) <<https://doi.org/10.1016/C2012-0-02754-8>>.

⁸⁸ Zhi-Hui Zhan and others, 'Orthogonal Learning Particle Swarm Optimization', *IEEE Transactions on Evolutionary Computation*, 15.6 (2011), 832–47 <<https://doi.org/10.1109/TEVC.2010.2052054>>.

compleja y más sencilla de implementar. Distintos mecanismos, como los vuelos de Levy, son utilizados para seguir el comportamiento de las partículas para capturar y estimar la posible dirección de convergencia del problema ⁸⁹.

Además, investigadores como Hong en 2009 y Liu, Wang en 2005 introdujeron la teoría de enjambres caóticos para mejorar el rendimiento de los algoritmos de enjambre de partículas, mediante los factores de peso de inercia adaptativa para un equilibrio eficiente. Es decir, introdujeron las propiedades de exploración y explotación ⁹⁰.

4.4. MULTIOBJETIVO EN LAS METAHEURÍSTICAS.

El principal dilema en la ingeniería de estructuras a la hora de definir objetivos es su contrariedad. Esto se presenta debido a que los dos principales objetivos son la optimización de materiales y la garantía de seguridad en el diseño. Aun así, al aumentar una la otra se ve afectada. Durante el diseño de un puente se busca optimizar materiales y garantizar un alto grado de seguridad. Estos son objetivos en conflicto ya que al aumentar la seguridad aumentan los costos. En respuesta es este tipo de problemas se crean las metaheurísticas con múltiples objetivos. Las metaheurísticas de un solo objetivo buscan una solución x que optimice de mejor manera la función objetivo f , mientras que en los problemas de múltiples soluciones se busca un conjunto de soluciones que se compensen de la mejor manera posible. Este tipo de respuestas se llaman respuestas no dominadas o codependientes ⁹¹.

⁸⁹ Michael F. Shlesinger and Joseph Klafter, 'Lévy Walks Versus Lévy Flights', in *On Growth and Form* (Dordrecht: Springer Netherlands, 1986), pp. 279–83 <https://doi.org/10.1007/978-94-009-5165-5_29>.

⁹⁰ Gandom and others.

⁹¹ Gustavo Zavala and others.

Existen dos enfoques para trabajar con los problemas multiobjetivo como posteriori y priori. En el tratamiento priori el problema se transforma en un problema de un solo objetivo mediante ponderaciones que definen el peso o la importancia de cada objetivo. Respecto al enfoque posteriori se evalúa el rendimiento de las soluciones obtenidas en la metaheurística ⁹².

Aunque este tipo de algoritmos son útiles en varios campos de la ingeniería son poco utilizados. Recientemente se realizó una encuesta ⁹³, que expuso que los algoritmos multiobjetivo más utilizados son: algoritmos evolutivos de objetivos múltiples (MOEA), el algoritmo genético de clasificación no dominado-II (NSGA-II) y el algoritmo evolutivo 2 de Pareto de fuerza (SPEA2) . En efecto, aunque el NSGA-II y SPEA2 se utilizan desde el año 2000 el uso de este tipo de algoritmos es limitado. Aunque se han propuesto distintos metaheurísticas multiobjetivo evolutivas y no evolutivas no se han implementado en problemas de ingeniería estructural.

En contraste, los algoritmos multiobjetivo proporcionan un conjunto de soluciones Pareto que proporcionan distintas ventajas, por ejemplo, presentan unas múltiples soluciones óptimas que se pueden adaptar a distintos problemas en específico.

Para entender de mejor manera los algoritmos multiobjetivo es necesario definir algunos conceptos como lo son: el problema de optimización de objetivos múltiples, La optimización de Pareto, el dominio de Pareto, el conjunto óptimo de Pareto y el frente de Pareto. La definición matemática formal de un problema multiobjetivo se puede observar en la figura 31.

⁹² Gaurav Dhiman and Vijay Kumar, 'Multi-Objective Spotted Hyena Optimizer: A Multi-Objective Optimization Algorithm for Engineering Problems', *Knowledge-Based Systems*, 150 (2018), 175–97 <<https://doi.org/10.1016/j.knosys.2018.03.011>>.

⁹³ Gustavo R. Zavala and others.

MOP encuentra un vector $x^*=[x_1^*,x_2^*,\dots,x_n^*]$ que satisfaga las restricciones de desigualdad $g_i(x)>0$, $i=1,2,\dots,m$, las p restricciones de igualdad $h_i(x)=0$, $i=1,2,\dots,p$, y minimizar el vector función $f(x)= [f_1(x), f_2(x),\dots,f_k(x)]^T$, donde $x=[x_1,x_2,\dots,x_n]^T$ es el vector de las variables de decisión.

Figura 31. Definición matemática formal de un problema multiobjetivo [30].

Logrando identificar que todos los vectores que satisfacen las restricciones iniciales conforman el conjunto K , en donde todos los elementos x del conjunto K son soluciones factibles. En efecto para definir los vectores que conforman el conjunto K , es necesario introducir el concepto de optimalidad de Pareto. Véase figura 32.

La optimalidad de Pareto, A un punto $x^* \in k$ es una optimalidad de Pareto si para cada $x \in k$ y $I=\{1,2,\dots, k\}$ ya sea, $p(f_i(x) =f_i(x^*))$ o hay al menos una $i \in I$ tal que $f_i(x) >f_i(x^*)$.

Figura 32. Definición matemática de optimalidad de Pareto [30].

De donde se concluye que x^* es una solución que pertenece al conjunto K , solo si x es un vector que no empeora los objetivos entre sí. Partiendo de esto es necesario establecer el concepto de dominio de Pareto, observado en la figura 33.

El dominio de Pareto, A un vector $u=(u_1, \dots, u_k)$ se dice que domina $v=(v_1, \dots, v_k)$ (denotado $u < v$) si y solo si u es parcialmente menos que v , i.e., $\forall i \in \{1, \dots, k\}, u_i < v_i$ y $\exists i \in \{1, \dots, k\}: u_i < v_i$.

Figura 33. Definición matemática del dominio de Pareto [56].

En efecto, un punto óptimo de Pareto es aquel que pertenece al conjunto K y no es codependiente o dominado por otro punto del conjunto. El dominio de Pareto relaciona dos soluciones que pueden representarse de manera binaria con 1 o 0.

Después de aplicar el operador vuelve una respuesta en donde se indica que 1 significa que hay dominación y 0 que no ⁹⁴.

La optimización de una metaheurística multiobjetivo estará basada en la búsqueda del conjunto de soluciones no dominadas entre sí, que también se conoce como conjunto de Pareto, el cual está definido en la figura 34.

Conjunto óptimo de Pareto para una dada MOP $f(x)$, es definido como $P^* = [x \in \Omega] \neg \exists x' \in \Omega, f(x') < f(x)$.

Figura 34. Definición matemática del conjunto de Pareto [30].

Una vez definido el conjunto de Pareto es prudente definir los vectores que conformara el conjunto de Pareto. Es importante recalcar que cada componente

⁹⁴ Gustavo Zavala and others.

del conjunto debe tener correspondencia con la función objetivo, esto es lo que se llama frente de Pareto. Véase la figura 35.

El frente de Pareto para un dado MOP $f(x)$ y es el conjunto optimo de Pareto P^* , es definido como $PF^*=\{f(x)|x \in P^*\}$.

Figura 35. Definición matemática frente de Pareto [56].

Es de particular importancia indicar que este análisis es a posteriori, por lo que se obtiene primero el conjunto de soluciones y posteriormente se evalúan en la función objetivo. Partiendo de los resultados los especialistas (ingenieros civiles) seleccionan la mejor respuesta de acuerdo a sus preferencias de diseño.

El tamaño del conjunto, es decir, el número de soluciones Pareto no está preestablecido. Es decir, no se recomienda que sea mayor a 500 soluciones. Aun así tener un menor número de soluciones no es un indicador de mayor o menor eficiencia, debido a que el tamaño del conjunto de soluciones depende del problema de optimización. A pesar de que el conjunto Pareto variara dependiendo del problema, existen condiciones que se buscan en este tipo de optimizaciones. La primera es la convergencia, es decir, buscamos que todas las soluciones sean aproximadas al frente de Pareto. La segunda es la diversidad, lo que indica que se quieren obtener soluciones que estén a lo largo del conjunto de Pareto, no todas ubicadas en un punto del frente de Pareto. Cuando gráficamente se observan lugares que no contienen vectores soluciones se dice que las indicaciones iniciales no cumplen con el principio de diversidad. Aunque el algoritmo cumpla

con su función no es prudente que queden regiones sin explorar, ya que podría contener soluciones de Pareto más óptimas ⁹⁵.

4.5. ESTRUCTURAS APORTICADAS DE CONCRETO.

4.5.1. Definición: Son estructuras tipo línea (una dimensión es significativamente mayor que las demás) a menudo se usan en edificaciones. Están compuestas por la unión de vigas y columnas conectadas de forma fija entre sí, formando una estructura plana o tridimensional. La carga en los pórticos causa flexión de sus elementos estructurales, y como se considera que las uniones son rígidas, este tipo de estructuras son generalmente indeterminadas, desde el punto de vista del análisis estructural. La resistencia de un pórtico se deriva de las interacciones de momento entre las vigas y columnas en las uniones ⁹⁶.

4.5.2. Métodos de análisis estructural: El análisis estructural es una parte integral de cualquier proyecto de ingeniería estructural, y es el proceso mediante el cual se predice el desempeño de una estructura dada bajo unas condiciones de carga establecidas. Las características de desempeño que son de interés para el diseño estructural son: (a) fuerzas internas (axial, cortante y momento flector), (b) deflexiones y (c) reacciones en los soportes. De esta manera el análisis estructural involucra el cálculo de estas cantidades causadas por unas cargas dadas y/u otros efectos externos (desplazamientos de los soportes y cambios de temperatura) ⁹⁷.

⁹⁵ Gustavo Zavala and others.

⁹⁶ R.C. Hibbeler, *Análisis Estructural*, ed. by Pearson Educación de México, 8 th ed. (México: Pearson, 2012).

⁹⁷ A. Kassimali, *Matrix Analysis of Structures*, 2nd ed. (Stamford: Cengage Learning, 2012).

Los métodos existentes para el análisis estructural lineal de estructuras tipo línea compuestas por elementos prismáticos (uniformes) son ⁹⁸⁹⁹:

a) Métodos aproximados:

- ☐ Análisis para cargas verticales
- ☐ Análisis para cargas laterales: Método del Portal
- ☐ Análisis para cargas laterales: Método del voladizo

b) Métodos clásicos

- ☐ Método de deformaciones consistentes (la fuerza o de la flexibilidad)
- ☐ Método de pendiente-deflexión
- ☐ Método de distribución de momentos o método de Cross

c) Métodos matriciales

- ☐ Método de la matriz de la flexibilidad
- ☐ Método de la rigidez

d) Método de elementos finitos (originados como una extensión de los métodos matriciales para su aplicación a estructuras tipo superficie como placas y cascarones)

Los métodos clásicos y aproximados generalmente fueron diseñados para analizar cierto tipo de estructuras, realizando los cálculos de forma manual, considerando ciertas suposiciones para reducir la cantidad de cálculos y requiere una comprensión por parte del analista del comportamiento estructural. En contraste

⁹⁸ Hibbeler.

⁹⁹ Kassimali.

los métodos matriciales fueron específicamente desarrollados para su implementación en computadores, ya que son sistemáticos (pueden programarse) y generales (el mismo formato del procedimiento analítico puede usarse en diversos tipos de estructuras reticulares) ¹⁰⁰.

Como el análisis de estructuras a gran escala y altamente redundantes por medio de los métodos clásicos puede llevar bastante tiempo, los métodos matriciales se usan comúnmente. Actualmente, en la mayoría de las oficinas de diseño, el análisis de estructuras tipo lien (ej. armaduras y marcos) se realiza en computadores, usando software basado en los métodos matriciales. Sin embargo, los métodos clásicos son preferidos por muchos ingenieros para analizar estructuras pequeñas porque proporcionan una mejor visión del comportamiento estructural. Los métodos clásicos también pueden ser usados para los diseños preliminares, para verificar los resultados de análisis computacionales (basados en métodos matriciales) y también para obtener las relaciones de fuerza desplazamiento necesario en los métodos matriciales. Además, el estudio de los métodos clásicos se considera esencial para desarrollar una comprensión del comportamiento estructural ¹⁰¹.

De los métodos matriciales, el método de la matriz de flexibilidad es conveniente para analizar estructuras pequeñas con pocas redundantes. El método de la rigidez es más sistemático y puede implementarse más fácilmente en los computadores, por lo cual es preferible para el análisis de estructuras a gran escala y altamente redundantes. La mayoría del software comercial disponible para el análisis estructural se basa en este método [2].

¹⁰⁰ Kassimali.

¹⁰¹ Kassimali.

Cuando se diseña una estructura es importante considerar las incertidumbres de los materiales y las cargas. En el diseño de pórticos de concreto, generalmente se usa el método de diseño por factores de carga y resistencia LRFD (Load and Resistance Factor Design), en el cual la incertidumbre puede tomarse en cuenta usando la teoría de la probabilidad, separando la incertidumbre del material con la incertidumbre de las cargas. Por ejemplo, para tener en cuenta las incertidumbres asociadas con las cargas, el método utiliza factores de carga aplicados a las cargas o a las combinaciones de estas ¹⁰².

4.5.3. Enfoques de diseño: clásico y optimización: Cuando se diseña una estructura tipo pórtico usando el método LRFD, la diferencias entre los enfoques de diseño clásico y el enfoque de diseño por optimización son las siguientes ¹⁰³:

El proceso de diseño estructural generalmente sigue los siguientes pasos:

- (1) Definir la función: En este paso se establece cuál es el uso de la estructura. Si se piensa en el diseño de un puente, se debe responder preguntas como, qué tan largo o ancho debe ser, cuántos carriles debe tener, cuáles son las cargas esperadas, etc.
- (2) Diseño conceptual: Aquí se establece el tipo de concepto de construcción que se debe usar. Para el caso del puente, cuando se va a construir es necesario decidir si se construirá un puente de celosía, un puente suspendido, un puente en arco, etc.
- (3) Optimización: En este paso, dentro del concepto seleccionado anteriormente y las restricciones de la función, se debe hacer que la estructura sea lo mejor posible. Para un puente podría ser minimizar el costo, lo que se puede lograr disminuyendo la cantidad de material.

¹⁰² Hibbeler.

¹⁰³ Christensen, CLIII.

(4) Detalles: Usualmente controlado por el mercado, factores sociales y estéticos. En el caso de un puente podría ser la selección un color interesante, detalles de iluminación, etc.

La forma tradicional o clásica y aún dominante de realizar el paso (3) es un proceso iterativo-intuitivo que se puede describir de la siguiente manera: (a) Un diseño específico es sugerido, (b) Los requisitos basados en la función son investigados (c) Si estos requisitos no se cumplen (ej. esfuerzos elevados), un nuevo diseño es sugerido, e incluso si se cumple con los requisitos, el diseño puede no ser el óptimo (ej. el puente puede ser demasiado pesado), así que aún se puede sugerir un nuevo diseño. (d) El nuevo diseño sugerido es devuelto al paso (b). De esta manera se forma un proceso, donde se prueba una serie de diseños hasta lograr convergencia en un diseño final aceptable ¹⁰⁴.

El enfoque basado en optimización matemática difiere conceptualmente del enfoque clásico iterativo- intuitivo. En el primero el problema de diseño se formula como un problema de optimización matemático, donde los requisitos debidos a la función, actúan como restricciones, y el concepto “lo mejor posible” se da en una forma matemática precisa (función objetivo). De esta manera, en el paso (3), el proceso de diseño es mucho más automático que en el proceso clásico intuitivo-iterativo. Este proceso de optimización matemática, aplicado a estructuras, es lo que se denomina optimización estructural ¹⁰⁵.

¹⁰⁴ Christensen, CLIII.

¹⁰⁵ Christensen, CLIII.

5. METAHEURÍSTICAS EN EL DISEÑO ESTRUCTURAL DE PORTICOS DE CONCRETO.

La idea central es plantear aquí los problemas de diseño estructural como un problema que puede ser abordado por algoritmos de optimización aproximados. Básicamente utiliza la toma de decisiones para seleccionar la mejor acción entre varias alternativas. Es aquí donde distintos investigadores buscan integrar disciplinas como el modelado matemático, la simulación, el análisis estadístico y la optimización matemática para determinar las soluciones a los problemas de optimización. Fundamentalmente se busca maximizar o minimizar una función objetivo bajo ciertas condiciones o limitaciones del problema a solucionar. Para desarrollar la toma de decisiones se necesitan variables de decisión que son la cantidad de recursos que se utilizan para optimizar una función. Un problema de optimización se resuelve cuando se encuentra los valores de las variables de decisión que maximiza o minimiza la función objetivo cumpliendo todas las condiciones o limitaciones planteadas desde el inicio.

Cuando se habla de problemas de ingeniería estructural se buscan optimizar los diseños para mejorar el desempeño estructural y hacer un uso eficiente de los recursos (materiales y financieros). Las formulaciones de los algoritmos buscan encontrar un diseño óptimo y factible, pero es necesario tener en cuentas distintas condiciones como el grado de seguridad del diseño. Teniendo en cuenta lo planteado anteriormente se plantea el siguiente problema. Una estructura esta optimizada para un costo mínimo teniendo restricciones en la tensión, desplazamiento, pandeo y periodo de vibración. Aun así, se observa que el diseño tiene una respuesta baja ante pruebas de resistencia sísmica. Esto se debe a que la energía inicial producida por el terremoto no fue incluida en el problema de

optimización. Aquí data la importancia en la definición de las funciones objetivo, en este caso deberían ser varias para obtener un diseño razonable ¹⁰⁶.

5.1. FUNCIONES OBJETIVO EN PROBLEMAS DE DISEÑO ESTRUCTURAL.

En este tipo de casos las funciones objetivo principales son el costo y el criterio, que están en función de distintas variables de diseño. Mediante la medida del mérito se obtiene un diseño óptimo. Esto varía dependiendo del número de soluciones posible para un diseño. Es decir, para una estructura existen diseños más económicos o con mayor grado de seguridad o con una mejor respuesta dinámica, etc. para llevar a cabo la mejor elección es necesario determinar los criterios más importantes. Alguna función objetivo puede buscar minimizar el peso estructural, que a su vez puede afectar la rigidez estructural, en efecto es necesario buscar una solución que compense de mejor manera los criterios iniciales. A continuación, se plantean algunas funciones objetivo comunes [32].

¹⁰⁶ Franklin and Truman Kevin.

5.1.1. Costo estructural. Esta función objetivo reúne criterios como el costo del material, fabricación e instalación. Generalmente, optimizar el precio de la estructura es el factor que inclina la balanza durante una contratación. Para plantear este problema de optimización se debe tener en cuenta los métodos con los que se proyectan el costo de una estructura: estimación del precio unitario, estimación del sistema, estimación del pie cuadrado y del pie cubico y la estimación del orden de magnitud. En el primer método es necesario tener los planos y especificaciones y aunque tiene tan solo un 5% de error es el menos adecuado a la hora de determinar el diseño por el tiempo y detalle que se necesita. En la estimación del sistema se agrupa la estructura en distintos componentes. En términos estructurales, esto se puede dividir en pisos, techos, marcos y paredes (concreto, acero, madera o una combinación de los tres). Tiene un valor de precisión del 10% y es viable cuando se busca que la función objetivo tenga en cuenta el costo de la estructura. Otro método viable para utilizar el costo estructural como función objetivo es el pie cuadrado y la estimación del pie cubico, presenta la ventaja de que existen tablas de los costos aproximados del pie cuadrado y cubico que dependen de los materiales a utilizar. El método de estimación del orden de la magnitud depende del tipo de edificio, el área y la ubicación, es poco recomendado debido a que no tiene una interacción real con la estructura. Por lo que los métodos anteriormente toman variables que determinan la optimización del costo estructural ¹⁰⁷.

En la figura 36 se observa una gráfica del tiempo de cálculo del costo contra la exactitud. Además, se muestran los valores de cada uno de los métodos para calcular un diseño de 2 millones.

¹⁰⁷ Saeed Gholizadeh and Vahid Aligholizadeh, 'Reliability-Based Optimum Seismic Design of RC Frames by a Metamodel and Metaheuristics', *Structural Design of Tall and Special Buildings*, 28.1 (2019), 1–19 <<https://doi.org/10.1002/tal.1552>>.

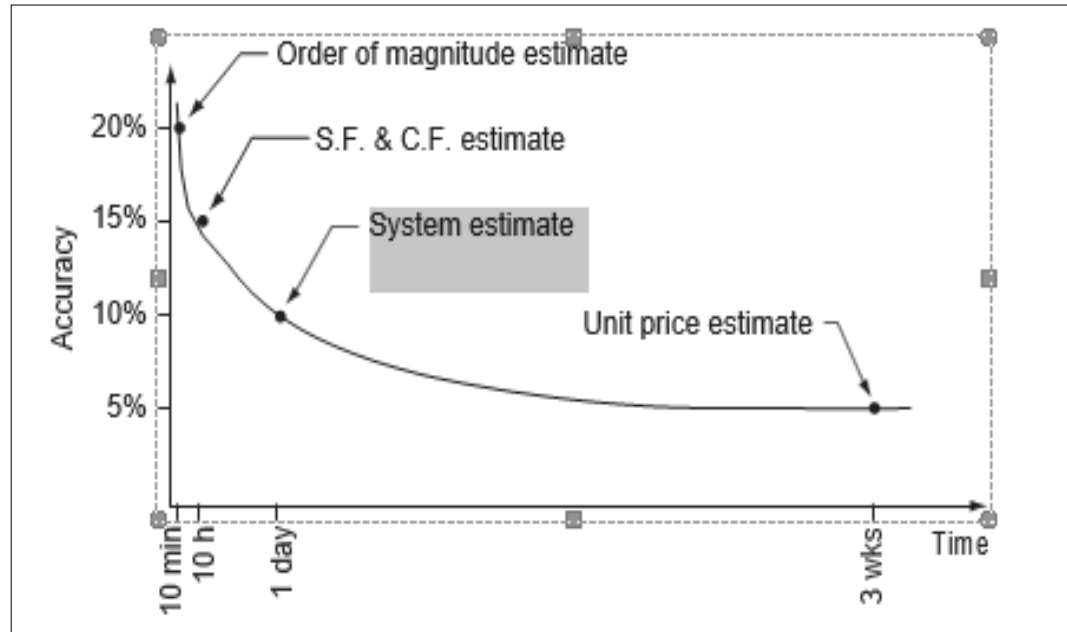


Figura 36. Tiempo y exactitud de las estimaciones para un edificio de \$ 2,000,000 [32].

El costo puede definirse por la fórmula 1.

$$f(i) = \sum_{n=1}^{\infty} C_{i1}V_i + \sum_{n=1}^{\infty} C_{i2}U_i \quad (1)$$

Donde $f(i)$ es el precio total de la estructura C_{i1} y C_{i2} es el costo de volumen, V_i es el volumen del miembro estructural, C_{i2} es el costo de instalación, y U_i es un miembro estructura instalado. Esta fórmula varía de acuerdo al método utilizado

para calcular el costo de la estructura. En el estudio presentado por ¹⁰⁸ optimizan el costo de una estructura de hormigón armado usando los algoritmos GB1 y GB2 (Global Best 1 y 2) obteniendo ahorros cercanos al 6 % cuando se compara con el diseño realizado por una oficina de proyectos experimentada.

5.1.2. Peso estructural. Al bajar el peso de la estructura, por lo general, disminuye el costo de la estructura. El peso de la estructura se puede describir por la fórmula 2.

$$W = \sum_{n=1}^{\infty} \rho_i A_i l_i \quad (2)$$

Donde ρ_i , A_i y l_i son la densidad del peso, el área transversal y la longitud del miembro i , respectivamente. Cuando la densidad ρ_i es igual para todos los miembros es posible seleccionar el volumen estructural como la función objetivo. En general esta función se comporta de manera similar a la de costo estructural cuando se utiliza el mismo material para el sistema estructural y cuando solo se considera el costo del miembro estructural ¹⁰⁹.

Se han usado una amplia variedad de algoritmos para optimizar el peso de diferentes estructuras. En ¹¹⁰ se usa el algoritmo de enjambre de partículas con toma de decisiones (DMPSO) en el diseño sísmico de pórticos de concreto

¹⁰⁸ A. Carbonell, V. Yepes, and F. González-Vidosa, 'Búsqueda Exhaustiva Por Entornos Aplicada Al Diseño Económico de Bóvedas de Hormigón Armado', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 27.3 (2011), 227–35 <<https://doi.org/10.1016/j.rimni.2011.07.003>>.

¹⁰⁹ Franklin and Truman Kevin.

¹¹⁰ S.H. Esfandiari, M.J. Urgessa, G.S.-Sheikholaferin, S. Manshadi Dehghan, 'Optimization of Reinforced Concrete Frames Subjected to Historical Time-History Loadings Using DMPSO Algorithm', *Structural and Multidisciplinary Optimization*, 58.5 (2018), 2119–34 <<https://doi.org/10.1007/s00158-018-2027-y>>.

reforzado usando cargas de tiempo historia optimizando el peso estructural .Los resultados muestran la funcionalidad de este algoritmo en este tipo de problemas dinámicos.

5.1.3. Energía de deformación. En una estructura determinada, cuando el trabajo interno (energía de deformación) realizado por las tensiones y las deformaciones tiene un valor mínimo se considera que es una estructura óptima. La energía de deformación está dada por la fórmula 3 ¹¹¹.

$$Es = \sum_{n=1}^{\infty} \alpha_i \epsilon_i V_i \quad (3)$$

Donde α_i , ϵ_i y V_i son la tensión y el volumen del miembro i .

Esta función objetivo se ha usado en una gran variedad de problemas de optimización estructural. En la publicación ¹¹² realizan la optimización topológica continua de estructuras bidimensionales usando el algoritmo ASAM (Algoritmo Simulated Annealing Modificado), obteniendo resultados satisfactorios y coherentes.

5.1.4. Energía potencial. Cuando se minimiza la energía potencial estructural pueden disminuir las fuerzas externas, aumentando así el grado de seguridad. Si las cargas dadas son $\bar{P} \quad p_1, p_2, \dots, p_n$ y la correlación de los desplazamientos correspondientes son $(O)= \delta_1, \delta_2, \dots, \delta_n$ la energia potencial se puede describir por la fórmula 4.

¹¹¹ Franklin and Truman Kevin.

¹¹² C. Millán Páramo and O. Begambre Carrillo, 'Solución de Problemas de Optimización Topológica Empleando El Algoritmo Simulated Annealing Modificado', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 32.2 (2016), 65–69 <<https://doi.org/10.1016/j.rimni.2014.11.005>>.

$$f4 = Ep = \{P\}^T \{\Delta\} = \sum_{n=1}^{\infty} p_i \delta_i \quad (4)$$

5.1.5. Desplazamientos. Los desplazamientos en puntos o regiones específicas de la estructura pueden seleccionarse como objetivos para disminuir los cambios estructurales cuando se sometan a diferentes situaciones de carga. Para esto se busca optimizar la fórmula 5, buscando minimizar o maximizar el desplazamiento según sea necesario.

$$f5 = \delta_1 \quad (5)$$

Por lo general es importante implementar esta función para garantizar un alto grado de seguridad en la estructura. Generalmente esta función se usa como una restricción como en el trabajo realizado por ¹¹³ donde se evaluaron estructuras planas aporticadas obteniendo resultados satisfactorios.

5.1.6. Energía sísmica. Durante un terremoto las estructuras están sometidas a excitaciones energéticas que ponen a prueba el diseño y sus consideraciones de respuesta dinámica y daño causado por cargas sísmicas. Para esto es necesario optimizar la respuesta dinámica mediante la limitación de criterios como la aceleración de la masa, el periodo fundamental y la energía de entrada del terremoto. La energía de entrada representa el trabajo realizado por la cizalladura que ocurre en la base de la estructura a medida que se mueve la cizalla a través del desplazamiento del suelo. El modelo matemático que representa este fenómeno se puede observar en por la fórmula 6 ¹¹⁴.

¹¹³ Niño Alvarez and Durán Cáceres.

¹¹⁴ Franklin and Truman Kevin.

$$E_i = \sum_{n=1}^{\infty} D_i^2 ((1/2) M S_{vi}^2) \quad (6)$$

Donde $D_i = \{\phi_i\} \{m\} / \sqrt{M}$ es el parámetro de energía de modo i , $\{\phi_i\}$ es el modo i normalizado y $\{m\} = M$ es la masa total de la estructura y S_{vi} es el espectro de respuesta de la velocidad en relación con el modo i .

5.1.7. Estructura de control. En el diseño del problema de optimización deben incluirse distintas variables como lo son la ganancia de realimentación y los parámetros de un sistema de control. Por lo general, las restricciones se llevan a cabo en el factor de amortiguamiento de bucle cerrado, las frecuencias, etc. Lo que se busca es minimizar el índice de rendimiento para reducir la energía de control o el tiempo efectivo de respuesta de amortiguamiento. En la fórmula 7, se define un controlador un índice de rendimiento (PI) que utiliza un regulado cuadrático lineal.

$$PI = \int (\{Z\}^T [Q] \{Z\} + \{U\}^T [R] \{U\}) dt \quad (7)$$

Donde $[Q]$ es la matriz de ponderación de estado semidefinida positiva, $[R]$ es la matriz de ponderación de control definida positiva, Z es el vector de desplazamiento de estado y u es un vector de entrada de control. Fundamentalmente se busca minimizar el índice de rendimiento cuadrático y comprobar la ecuación de estado del sistema, sin dejar a un lado la ley de control de retroalimentación del estado que esta definida en la fórmula 8.

$$\{u\} = -[R]^{-1} [B]^T [P] \{z\} = -[G] \{z\} \quad (8)$$

Donde $[G]$ es la matriz de ganancia de bucle cerrado, y $[P]$ es la matriz de Riccati. Por lo que la función objetivo a minimizar se observa en la fórmula 9.

$$f_7 = \{z_0\}^T [P] \{z_0\} \quad (9)$$

En este caso z_0 es el vector de estado inicial (vector de perturbación). Finalmente, la fórmula 10 muestra la función objetivo final, que define el tiempo de respuesta de amortiguamiento.

$$f_8 = \frac{\{z\}^T [P] \{z_0\}}{\{z\}^T [Q] \{z_0\}} \quad (10)$$

Donde $[Q]$ es la matriz de ponderación. Es importante recalcar que durante todo el procedimiento se deben tener en cuenta los requisitos especiales de control de ingeniería de terremoto. Para este tipo de problema es recomendado utilizar el algoritmo de control activo óptimo generalizado (GOAC). Además, como se mostró anteriormente los problemas de minimización se pueden trabajar igual a los de maximización ¹¹⁵.

5.2. PROBLEMAS DE OPTIMIZACIÓN DE DISEÑO ESTRUCTURAL

Determinar el momento exacto en que se utilizó por primera vez la optimización en el diseño de estructura es complicado. Aun así, en la antigüedad, durante la construcción de los acueductos romano se observaban distintos métodos para

¹¹⁵ Franklin and Truman Kevin.

disminuir el peso y la cantidad necesaria de material para realizar la construcción. En 2003, Ohsaki y Swan realizaron una revisión histórica sobre distintas técnicas de optimización aplicadas a estructuras discretas. En donde se describen tres periodos importantes. El primero, que transcurrió desde 1894 hasta 1904 con los estudios realizados por J. C. Maxwell y A. G. M. Michel, que fueron pioneros en el campo. El segundo se reestablece en los años sesenta y setenta, cuando el uso de equipos de cómputo aumentaba. Finalmente, el tercer periodo, durante los años ochenta y noventa, en donde se destaca el gran avance de las tecnologías informáticas. Se tiene como referencia que las primeras metaheurísticas utilizadas en diseño estructural fueron implementadas en los años 2000 por Kicinger. En contraste, los problemas de optimización multiobjetivo aparecieron desde los años ochenta con autores como Goldberg y Samtini. Finalmente se utilizan técnicas de único objetivo para resolver problemas mediante algoritmos genéticos y evolutivos en los años 2000 ¹¹⁶. Partiendo de lo anterior se encuentra que los problemas que se abordaban eran principalmente los de diseño de barras o elementos y el diseño topológico.

Los problemas de diseño de barras o elementos, principalmente es un proceso de optimización local que interviene en el tamaño y forma de los elementos estructurales. Se enfoca en encontrar las medidas de una forma geométrica específica transversal al eje principal de la estructura o determinando el valor de las variables que conducen a la geometría óptima. El diseño topológico busca encontrar la topología completa del diseño de una estructura. También es conocido como diseño óptimo topológico (TOD) y puede ser implementado en estructuras discretas o continuas. Utiliza parámetros como el número de

¹¹⁶ Gustavo R. Zavala and others.

elementos y los vínculos o continuidades entre ellos para calcular la distribución óptima de masas, en busca de la estabilidad estructural ¹¹⁷.

¹¹⁷ J. Martínez-Frutos and P. Martí, 'Diseño Óptimo Robusto Utilizando Modelos Kriging: Aplicación Al Diseño Óptimo Robusto de Estructuras Articuladas', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 30.2 (2014), 97–105 <<https://doi.org/10.1016/j.rimni.2013.01.003>>.

5.2.1. Diseño de barras o elementos. Existen distintos tipos de problemas relacionados al diseño de barras. Por lo general se utilizan en complicaciones con estructuras compuestas de barras planas y espaciales, como armazones y marcos. Las estructuras de armaduras se definen como estructuras compuestas por elementos de barra que se unen por uniones articuladas. Para calcular el equilibrio es necesario determinar el área transversal de las barras, todo esto sin información de la geometría de los elementos, dejando a un lado forma y tamaño. En contraste, una estructura de marco requiere, además de las áreas transversales, la inercia a la flexión, por lo que, si es necesario definir las formas y tamaños de los elementos a priori, y así poder determinar la estabilidad de la estructura. En efecto, los problemas de estructuras de marco son más complejos que los de barras. Cuando la estructura está en tres dimensiones la complejidad aumenta considerablemente ya que aparecen cuatro variables: el área, dos momentos de inercia y una inercia torsional.

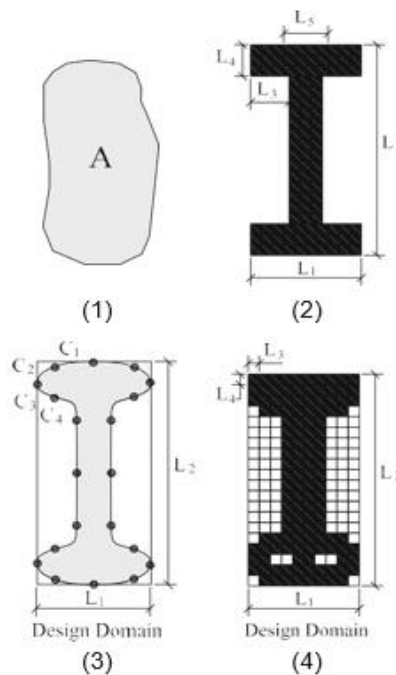


Figura 37. 1) Optimización de área. 2) Optimización de tamaño. 3) optimización de forma. 4) Optimización de topología. [37]

En todos los problemas se debe calcular la geometría cruzada de cada barra de la estructura, que define cuales variables de decisión son necesarias para plantear el problema. Se pueden distinguir cuatro problemas distintos. El primero es la optimización del área, en los cuales se toman las áreas de las barras sin considerar los tamaños ni formas como variables de decisión del problema. La optimización de tamaño es el segundo, la cual se enfoca en encontrar los valores de una estructura con forma predefinida. La optimización de forma, buscan encontrar la forma y geometría de una estructura que se encuentra delimitada por un marco, mediante nodos y trayectorias discretizadas, limitadas por el problema. Finalmente, la cuarta es la optimización topológica de la sección transversal, que utiliza una sección elemental conocida, se discretiza para que existan regiones triangular y cuadradas. El proceso de optimización se basa en la eliminación de piezas y reorganizar los elementos vinculados por sus vecinos. Pueden aparecer espacios en el interior. En la figura 37 se observan los subproblemas de manera gráfica.

5.2.2. Diseño topológico. Anteriormente se expuso que TOD tenía en cuenta la distribución de elementos internos, los enlaces y la geometría estructural para ser optimizada. En general, buscan encontrar la mejor distribución de las masas resistentes de material donde sea necesario, así como en el cálculo de su tamaño. Como en todo problema de optimización se trata de llegar a una función objetivo, teniendo en cuenta limitaciones que se establecen en el estado inicial del sistema. Se tiene conocimiento de tres categorías para este tipo de problemas. Los cuales se basan en el tipo de elemento utilizado o la forma en que se realizan las subdivisiones, es decir, las barras o elementos interconectados por nodos, lados o superficies este tipo de diseño es utilizado frecuentemente en estructuras metálicas y no de concreto.

El primer caso es sin optimización topológica o diseño topológico predefinido; en este tipo de casos la topología de la estructura es seleccionada por el ingeniero, por lo que será permanente durante todo el proceso de optimización, es decir, el número de elementos, enlaces, restricciones de movimientos, al igual que la sus relaciones entre sí serán constantes. Es considerado como una categoría porque permite la clasificación de los problemas TOD en los que solo se consideran los problemas de diseño de barras y estructuras continuas. Por lo general se presentan problemas de barras en dos o tres dimensiones donde una de las mediadas es equivalente a una de las otras dos (longitud en la sección transversal), en donde se utiliza una representación lineal para describir el eje longitudinal, cada elemento se intercomunica por nodos. Además, se dan problemas en los que la forma es constante y solo se optimiza el grosor de los elementos de la estructura o la idoneidad de las longitudes principales de la estructura ¹¹⁸¹¹⁹.

¹¹⁸ Gustavo Zavala and others.

¹¹⁹ Millán Páramo and Begambre Carrillo.

El segundo caso de diseño topológico es la optimización topológica discreta; este caso es similar al anterior pero la estructura no está definida a priori. Para encontrar el diseño se parte de un dominio de diseños o zona de estudio, es decir, en un cuerpo de alambre que contienen condiciones de contorno que están en dos o tres dimensiones. Por lo general se divide en cuadrantes para saber la ubicación de los vértices. El diseño se seleccionará teniendo en cuenta las uniones entre los elementos longitudinales que representan las barras.

Finalmente, tenemos el tercer caso, el cual llamaremos optimización topológica continua. Se utiliza para diseños de estructuras tipo placa o estructuras en dos o tres dimensiones. Se caracteriza porque puede tener el valor del grosor constante o no, además toma como variables de decisión los límites del espacio de exploración y las condiciones del entorno. Se toma como inicio una figura o esqueleto básico con medidas periféricas ya definidas y con exceso de material conocido como dominio de diseño. Independientemente del número de dimensiones el dominio de diseño está discretizado en pequeños elementos. El objetivo es la optimización de la resistencia y la distribución del material, da como resultado una figura nueva, descartando el material de cubierta en la resistencia del conjunto. Por lo general se obtienen figuras que adoptan formas de barras utilizando elementos discretos unidos entre sí. En los problemas de tres dimensiones la evolución variando los tamaños en las tres dimensiones, mediante el acople y desacople de elementos finitos volumétricos unidos a sus vecinos por superficies de contacto. Posteriormente se encontrarán las limitaciones y condiciones requeridas de forma y tamaño para cumplir con la estabilidad de la estructura.¹²⁰ En la figura 38 se observan la clasificación de diseño topológico.

¹²⁰ Gustavo R. Zavala and others.

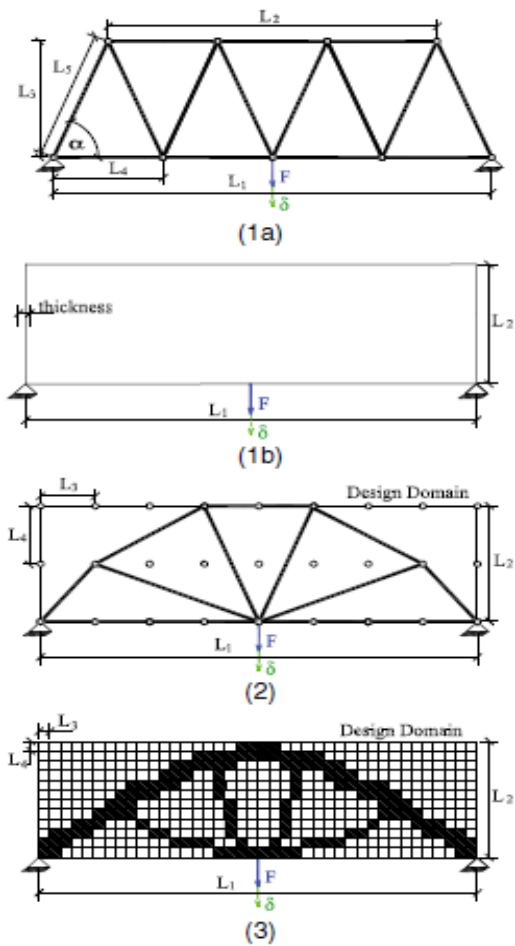


Figura 38. Clasificación del diseño topológico. 1) Sin optimización topológica con dos variantes a) barras b) Estructuras continuas. 2) Optimización topológica discreta. 3) Optimización topológica continua [30].

5.3. RESULTADOS DE APLICACIONES DE METAHEURISTICAS EN ESTRUCTURAS APORTICADAS DE CONCRETO.

A continuación, se presenta una tabla que resume investigaciones relevantes reportadas en la base SCOPUS¹²¹ para la optimización de estructuras aporticadas de concreto con algoritmos metaheurísticos.

Tabla 3. Resultados de aplicaciones de metaheurísticas en edificaciones aporticadas.

¹²¹ Elsevier B.V.

¹²² A. Kaveh and A. F. Behnam, 'Design Optimization of Reinforced Concrete 3D Structures Considering Frequency Constraints via a Charged System Search', *Scientia Iranica*, 20.3 (2013), 387–96 <<https://doi.org/10.1016/j.scient.2012.11.017>>.

¹²³ G A O Yun-long and Y A N Peng, *Harmony Search Algorithm*, 1st edn (Warsaw, 2016), xxxi <<https://doi.org/10.13195/j.kzyjc.2015.0352>>.

¹²⁴ Jiulin Bai and others, 'Seismic Optimization Design for Uniform Damage of Reinforced Concrete Moment-Resisting Frames Using Consecutive Modal Pushover Analysis', *Advances in Structural Engineering*, 19.8 (2016), 1313–27 <<https://doi.org/10.1177/1369433216642045>>.

¹²⁵ I. N. Serpik, I. V. Mironenko, and V. I. Averchenkov, 'Algorithm for Evolutionary Optimization of Reinforced Concrete Frames Subject to Nonlinear Material Deformation', *Procedia Engineering*, 150 (2016), 1311–16 <<https://doi.org/10.1016/j.proeng.2016.07.304>>.

¹²⁶ Panagiotis E. Mergos, 'Efficient Optimum Seismic Design of Reinforced Concrete Frames with Nonlinear Structural Analysis Procedures', *Structural and Multidisciplinary Optimization*, 58.6 (2018), 2565–81 <<https://doi.org/10.1007/s00158-018-2036-x>>.

¹²⁷ Neda Nabid, Iman Hajirasouliha, and Mihail Petkovski, 'Performance-Based Optimisation of RC Frames with Friction Wall Dampers Using a Low-Cost Optimisation Method', *Bulletin of Earthquake Engineering*, 16.10 (2018), 5017–40 <<https://doi.org/10.1007/s10518-018-0380-2>>.

¹²⁸ Sonia Chutani and Jagbir Singh, 'Optimal Design of RC Frames Using a Modified Hybrid PSOGSA Algorithm', *Archives of Civil Engineering*, 63.4 (2017), 123–34 <<https://doi.org/10.1515/ace-2017-0044>>.

¹²⁹ Hamid Mortezaie and Freydoon Rezaie, 'Optimization and Evaluation of Three-Dimensional High-Rise Concrete Structure Equipped by TMD with Considering the Soil-Structure Interaction Effect', *Journal of Vibroengineering*, 20.5 (2018), 2111–27 <<https://doi.org/10.21595/jve.2017.18628>>.

¹³⁰ Gholizadeh and Aligholizadeh.

Tabla 3. Resultados de aplicaciones de metaheurísticas en edificaciones aporticadas

N°	AUTOR	ARTICULO	ESTRUCTURA	OBJETIVO	RESTRICCIONES	ALGORITMO	NORMAS DE DISEÑO EMPLEADAS	TIPO DE ANALISIS EMPLEADO	LENGUAJE DE PROGRAMACIÓN O PROGRAMA USADO	CONCLUSIONES
1	A. Kaveh, & A.F. Behnam	Design optimization of reinforced concrete 3D structures considering frequency constraints via a charged system search [60].	Estructura de 7 pisos con 3 tramos en cada dirección horizontal	Peso de la estructura.	Pandeo de los miembros de compresión, deriva máxima permitida de la estructura y frecuencia natural de la estructura.	Busqueda de sistema cargados (CSS).	ACI 318-05 y las cargas se basan en ASCE7-05	Método de rigidez estándar.	elabs + lenguaje de programación no especificado	Los resultados obtenidos en cada una de las iteraciones del diseño de la estructura se transmiten a las nuevas soluciones de las iteraciones posteriores. Esto significa que se ha producido una explotación del algoritmo, incluso en las iteraciones finales, lo que lleva a mejores resultados para un diseño óptimo de la estructura. El valor inicial del objetivo fueron 750 Toneladas valor final 500. Ton % variación 33%
2	Gebrail Bekda & Sinan Melih Nigdeli	Modified Harmony Search for Optimization of Reinforced Concrete Frames [61].	Portico concreto reforzado de tres pisos y de tres tramos.	Costo de acero de refuerzo.	Pandeo de los elementos estructurales.	Busqueda de la armonía modificada.	ACI-318 (requisitos del código de construcción para la estructura de hormigón armado)	Análisis de tiempo historia	Lenguaje de programación no especificado.	El costo total del diseño óptimo es de \$ 724.54 . El valor para las iteraciones iniciales es de \$ 1148, casi un 58% más alto que el costo óptimo.
3	Jiulin Bai, Shuangshua ng Jin, Chuan Zhang & Jinping Ou.	Seismic optimization design for uniform damage of reinforced concrete moment-resisting frames using consecutive modal pushover analysis [62].	Prototipos de marcos resistentes al momento de concreto reforzado, con cinco y ocho pisos	Rendimiento sísmico.	Los refuerzos longitudinales de componente deben satisfacer las restricciones de refuerzo mínimas y máximas.	CMP (consecutive modal pushover analysis)	Código de diseño sísmico chino (GB 50011-2010)	Análisis de empuje modal consecutivo	OpenSees + lenguaje de programación no especificado	Para los dos porticos analizados se puede evidenciar que la variación de la deriva, desde el valor inicial al valor final, de la función objetivo de 11.7% y 17.2%, respectivamente para cada portico.
4	I.N. Serpika, I. V. Mironenko & V.I. Averchenkov ^b	Algorithm for Evolutionary Optimization of Reinforced Concrete Frames Subject to Nonlinear Material Deformation [63]	Portico sencillo	Costo de producción planificado del Portico	Resistencia, dureza y resistencia a la fisuración	Algoritmo Evolutivo	JSC NIC Construction, Moscow, 2012	Fuerza horizontal Equivalente	software BGITAFEM / OPTIMA	La optimización de un marco de un solo tramo ilustró una estabilidad y velocidad de convergencia suficientemente altas del esquema computacional presentado. El algoritmo propuesto se puede utilizar en los sistemas de diseño asistido por ordenador para las estructuras de construcción. El valor inicial del objetivo fue de 35 cms. y el valor final fue de 20 cms, donde la variación fue de 43%.

Tabla 3. Resultados de aplicaciones de metaheurísticas en edificaciones aporticadas (continuación)

N°	AUTOR	ARTICULO	ESTRUCTURA	OBJETIVO	RESTRICCIONES	ALGORITMO	NORMAS DE DISEÑO EMPLEADAS	TIPO DE ANALISIS EMPLEADO	LENGUAJE DE PROGRAMACIÓN O PROGRAMA USADO	CONCLUSIONES
5	M. J. Esfandiar ¹ , G. S. Urgessa, S. Sheikholaefi n & S. H. Dehghan Marshadi	Optimization of reinforced concrete frames subjected to historical time-history loadings using DMPPO algorithm [58].	Portico 2D de doce niveles	Costo total de la estructura	Las restricciones derivadas de las disposiciones de diseño ACI 318-11 (Comité ACI 318 2011) para problemas de optimización de hormigón armado son la resistencia, la capacidad de servicio y la ductilidad.	Optimización de enjambre de partículas usando toma de decisiones (DMPPO)	ACI-318	Análisis de tiempo historia	Lenguaje de programación no especificado.	El algoritmo Acelera y simplifica el proceso de optimización, proporciona una técnica ideal para modelar el diseño práctico contra la carga del historial de tiempo considerando las variaciones en las dimensiones transversales de los elementos de estructura de hormigón. Además de detallar las combinaciones y la ubicación de las barras de refuerzo para encontrar soluciones de diseño realistas que sean directamente construíbles. El valor inicial del objetivo fue \$ 60000 valor final \$ 36000, % variación 40%
6	Panagiotis E. Mergos	Efficient optimum seismic design of reinforced concrete frames with nonlinear structural analysis procedures [64].	Portico de tres niveles con tres luces, portico de seis niveles con dos luces.	Costo material de los marcos de concreto reforzado.	Las restricciones para cargas estáticas se basan en las disposiciones del Eurocódigo 2 (EC2) (CEN 2000). Las restricciones de diseño de EC2 están relacionadas con los	Algoritmo Genético	ACI-318-08	Análisis de historial de respuesta no lineal y de empuje	Matlab	Para el portico de tres niveles la metodología propuesta conduce a una solución óptima del 6% (4244 € en lugar de 4502 €) menos costosa que la metodología estándar. Para el portico de seis niveles la metodología propuesta arroja una solución óptima de 21% (5391 € en lugar de 6820). €) menos costo que el enfoque estándar.
7	Neda Nabid, Iman Hajirasouliha & Mhail Petkovski	Performance based optimisation of RC frames with friction wall dampers using a low cost optimisation method [65].	Cinco porticos RC diferentes, de calidad , con 3, 5, 10, 15 y 20 pisos con dispositivos de disipación de energía de fricción tipo pared en su rango medio	Deriva entre pisos y la capacidad de disipación de energía de los amortiguadores	La deriva máxima entre plantas [ASCE / SEI 41-13 (2014)]	Algoritmo Genético	ACI 318-14	Análisis modal	Lenguaje de programación no especificado.	En comparación con los amortiguadores de pared de fricción de diseño convencional con distribución uniforme de la carga de deslizamiento, los amortiguadores de diseño óptimo exhibieron índices de deriva de daños entre 43% y 75% más bajos e índices de daño global, respectivamente, cuando se sometieron a terremotos compatibles con espectro sintético . Para la misma fuerza de fricción total, el uso del método de optimización propuesto puede aumentar la capacidad de disipación de energía de los amortiguadores de pared de fricción hasta en un 46%.

Tabla 3. Resultados de aplicaciones de metaheurísticas en edificaciones aporticadas (continuación)

N°	AUTOR	ARTICULO	ESTRUCTURA	OBJETIVO	RESTRICCIONES	ALGORITMO	NORMAS DE DISEÑO EMPLEADAS	TIPO DE ANALISIS EMPLEADO	LENGUAJE DE PROGRAMACIÓN O PROGRAMA USADO	CONCLUSIONES
8	SONIA CHUTANI & JAGBIR SINGH	OPTIMAL DESIGN OF RC FRAMES USING A MODIFIED HYBRID PSO-GSA ALGORITHM [66].	Portico de cuatro niveles de tres luces.	El costo de los elementos estructurales de concreto reforzado incluye principalmente los costos de concreto, acero y encofrado.	Las restricciones de diseños se consideraron de acuerdo con las disposiciones de la India para el diseño de vigas RC (IS 456: 2000)	Algoritmo híbrido (PSO y GSA)	Código Indio IS: 456: 2000 (cuarta revisión)	Análisis de tiempo historia	Lenguaje de programación no especificado.	El algoritmo propuesto supera las limitaciones de dos algoritmos individuales existentes (PSO modificado y GSA modificado) en gasto computacional.
9	Hamid Montezai & Freydoon Rezaie.	Optimization and evaluation of three-dimensional high-rise concrete structure equipped by TMD with considering the soil-structure interaction effect [67].	Portico 3D de 20 pisos	El desplazamiento lateral máximo y la deriva entre pisos de la estructura.	Amortiguamiento.	Método de diseño de plástico basado en el rendimiento (PBDP)	ACI 318-14	Análisis de tiempo historia	OpenSees + lenguaje de programación no especificado	Al ajustar el amortiguamiento con una relación de masa de 1.67% y una relación de sintonía de 0.8 y 1 y una amortiguación de 14%, cero y 10% respectivamente para los terremotos se obtienen resultados óptimos para la estructura planteada.
10	Saeed Gholizadeh & Vahid Aligholizadeh	Reliability-based optimum seismic design of RC frames by a metamodel and metaheuristics [68]	Portico de nueve niveles de cuatro luces.	Costo estructural total	ACI 318-08	Algoritmo metaheurístico o caótico de optimización de cuerpos en colisión (CECBO)	ACI-318-08	Análisis de tiempo historia	Matlab	El algoritmo CECBO propuesto encuentra una mejor solución (4,8% menos en costo) en comparación con el ECBO.

La mayor parte de los estudios presentados anteriormente, incluyen las restricciones de diseño establecidas en las normas sismo resistente de diferentes regiones (ej. ACI-318 y Eurocode) para pórticos de concreto reforzado, lo cual facilita la aplicación directa de los resultados a ejemplos reales.

Se destaca que la función objetivo más utilizada en la optimización de diseño de pórticos es el costo de la estructura, considerando la importancia no solo del peso de los elementos estructurales sino también de los costos del proceso constructivo (ej. Costos de encofrado y figurado del acero). Además se observa que los algoritmos utilizados en los diferentes problemas resultan de la combinación o modificación de diferentes algoritmos originales para mejorar el proceso de optimización (convergencia y costo computacional).

6. CONCLUSIONES Y TRABAJOS FUTUROS.

El uso de las metaheurísticas en campos como el de la ingeniería de estructural se ha incrementado en los últimos años debido a que plantea soluciones de optimización confiables de alta calidad, a partir de un enfoque sencillo y eficaz para distintos problemas específicos. Se espera que la producción científica en esta área, especialmente en problemas de multiobjetivo aumente progresivamente, lo que permitirá dar solución a problemas reales de gran complejidad mediante distintos métodos aproximados que relacionan la inteligencia artificial y los algoritmos metaheurísticos.

De acuerdo a la figura dos, el uso de las metaheuristicas en la optimización de estructuras aporticadas de concreto se evidencia un aumento en las investigaciones publicadas en esta área del conocimiento. La cantidad de publicaciones que se presentaron en los años recientes (2017-2018) es aproximadamente cinco veces mayor a las publicaciones presentadas durante los años 2000 a 2005, lo que indica un creciente interés en este tema durante la última década.

Además, es un problema común de los algoritmos utilizados actualmente (que no implementa metaheurísticas) el estancamiento de las soluciones en mínimos locales, soluciones aceptables y factibles, pero no óptimas. Las metaheurísticas contribuyen en la eliminación de este problema ya que su funcionamiento es basado en un vecindario (recocido simulado y tabú), que en distintos estudios han demostrado ser eficaces frente al problema planteado. Todo esto es posible gracias al desarrollo de operadores que permiten la degradación del problema

para así evitar la selección de soluciones no óptimas. Finalmente contribuyen de manera indirecta a la convergencia de los problemas que es fundamental en los algoritmos aproximados. Ejemplo de estos operadores son la recombinación en los algoritmos genéticos y evolutivos que permiten explorar nuevos “valles” evitando caer en óptimos locales, generando un entorno con más posibilidades de encontrar un óptimo global. En efecto, la combinación de propiedades como la flexibilidad, versatilidad, diversificación y métodos eficaces frente a los problemas planteados en el diseño de estructuras de hormigón, impulsará inevitablemente las investigaciones en esta área en específico.

Los algoritmos más utilizados según la búsqueda realizada en SCOPUS y la información analizada en la tabla 3 son los Algoritmos Genéticos seguido por el algoritmo de optimización por enjambre de partículas y algunas aplicaciones del algoritmo de recocido simulado. Algoritmos relativamente recientes como el HS, FA y el CSS también se han utilizado en problema de optimización de estructuras aporticadas de concreto, aunque en menor medida en comparación con los algoritmos metaheurísticos, mencionados anteriormente.

Los algoritmos metaheurísticos, como los mencionados en este documento, que simplifican el proceso de optimización, serán herramientas de gran utilidad que servirán de apoyo para el diseño de estructuras. A partir de lo anterior, será de gran importancia integrar y enlazar estas herramientas de optimización con diferentes programas de acuerdo a las necesidades de los proyectos en el proceso de diseño estructural.

Muchos de los problemas de optimización de estructuras aporticadas de concreto reportados en la literatura especializada están codificados usando representación de punto flotante (es decir, se usan números reales para todas las variables de decisión). Sin embargo, no se encontró ningún estudio en el que se haya utilizado algoritmo de la evolución diferencial (DE) para resolver el problema de optimización estructural. Esto es sorprendente considerando que el algoritmo de la evolución diferencial es un enfoque muy poderoso para resolver problemas en los que todas las variables de decisión son números reales. Un trabajo futuro podría enfocarse en la implementación y aplicación de este algoritmo en problemas de optimización estructural de estructuras aporticadas de concreto.

En uno de los estudios consultados, se desarrolla una nueva estrategia computacionalmente eficiente para el diseño sísmico automatizado óptimo basado en el rendimiento y la deformación de marcos de hormigón armado utilizando procedimientos de análisis estáticos y / o dinámicos no lineales. La metodología propuesta se basa en un procedimiento iterativo simple, basado en la deformación, que diseña el refuerzo de acero de los marcos de concreto para cumplir con sus objetivos de rendimiento. El refuerzo de acero longitudinal se selecciona primero para satisfacer los estados límite de servicio y el refuerzo de acero transversal se elige adecuadamente para cumplir con los estados límite últimos.

Por lo tanto, esta estrategia de solución propuesta puede convertirse en una herramienta valiosa en el diseño sísmico automatizado, óptimo basado en la deformación y el rendimiento de marcos de hormigón armado con procedimientos de análisis estructural no lineal. Sin embargo, se requiere más investigación para extender y validar la metodología propuesta a marcos complejos de hormigón armado tridimensional con una gran cantidad de variables de diseño. Además, se

necesitan consideraciones adicionales para ampliar la metodología desarrollada para abordar otros tipos de estructuras de hormigón, como marcos rellenos, muros y sistemas duales.

BIBLIOGRAFIA

DE ARMAS ADRIAN, JESICA, 'Problemas de Corte: Metodos Exactos y Aproximados Para Formulaciones Mon y Multi-Objetivo (Tesis Doctoral)' (Universidad de la Laguna, 2011)

BAI, JIULIN, SHUANGSHUANG JIN, CHUAN ZHANG, AND JINPING OU, 'Seismic Optimization Design for Uniform Damage of Reinforced Concrete Moment-Resisting Frames Using Consecutive Modal Pushover Analysis', *Advances in Structural Engineering*, 19.8 (2016), 1313–27 <<https://doi.org/10.1177/1369433216642045>>

BAJULA, SHUMEET, *Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning* (Pittsburgh, 1994)

BELLMAN, RICHARD, *Dynamic Programming*, 2nd edn (United States: Dover Publications Inc., 2003)

BILCHEV, G., AND I. C. PARMEE, 'The Ant Colony Metaphor for Searching Continuous Design Spaces', 1995, pp. 25–39 <https://doi.org/10.1007/3-540-60469-3_22>

CAMP, CHARLES V., SHAHRAM PEZESHK, AND HÅKAN HANSSON, 'Flexural Design of Reinforced Concrete Frames Using a Genetic Algorithm', *Journal of Structural Engineering*, 129.1 (2003), 105–15
<[https://doi.org/10.1061/\(ASCE\)0733-9445\(2003\)129:1\(105\)](https://doi.org/10.1061/(ASCE)0733-9445(2003)129:1(105))>

CARBONELL, A., V. YEPES, AND F. GONZÁLEZ-VIDOSA, 'Búsqueda Exhaustiva Por Entornos Aplicada Al Diseño Económico de Bóvedas de Hormigón Armado', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 27.3 (2011), 227–35 <<https://doi.org/10.1016/j.rimni.2011.07.003>>

CHARON, IRÈNE, AND OLIVIER HUDRY, 'The Noising Method: A New Method for Combinatorial Optimization', *Operations Research Letters*, 14.3 (1993), 133–37
<[https://doi.org/10.1016/0167-6377\(93\)90023-A](https://doi.org/10.1016/0167-6377(93)90023-A)>

CHEN, GECHENG, AND ZHIQIANG GE, 'SVM-TREE AND SVM-FOREST Algorithms for Imbalanced Fault Classification in Industrial Processes', *IFAC Journal of Systems and Control*, 8 (2019), 100052
<<https://doi.org/10.1016/j.ifacsc.2019.100052>>

CHRISTENSEN, P., *An Introduction to Structural Optimization*, Solid Mechanics and Its Applications, 1st ED. (Dordrecht: Springer Netherlands, 2008), CLIII
<<https://doi.org/10.1007/978-1-4020-8666-3>>

CHUTANI, SONIA, AND JAGBIR SINGH, 'Optimal Design of RC Frames Using a Modified Hybrid PSOGSA Algorithm', *Archives of Civil Engineering*, 63.4 (2017), 123–34 <<https://doi.org/10.1515/ace-2017-0044>>

COELLO COELLO, C. A., A. D. CHRISTIANSEN, AND F. SANTOS HERNÁNDEZ, 'A Simple Genetic Algorithm for the Design of Reinforced Concrete Beams', *Engineering with Computers*, 13.4 (1997), 185–96 <<https://doi.org/10.1007/BF01200046>>

DHIMAN, GAURAV, AND VIJAY KUMAR, 'Multi-Objective Spotted Hyena Optimizer: A Multi-Objective Optimization Algorithm for Engineering Problems', *Knowledge-Based Systems*, 150 (2018), 175–97 <<https://doi.org/10.1016/j.knosys.2018.03.011>>

DORIGO, M., 'Optimization, Learning and Natural Algorithms (PhD Thesis)' (Politecnico di Milano, 1992)

DUECK, GUNTER, 'New Optimization Heuristics', *Journal of Computational Physics*, 104.1 (1993), 86–92 <<https://doi.org/10.1006/jcph.1993.1010>>

DUECK, GUNTER, AND TOBIAS SCHEUER, 'Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing', *Journal of Computational Physics*, 90.1 (1990), 161–75 <[https://doi.org/10.1016/0021-9991\(90\)90201-B](https://doi.org/10.1016/0021-9991(90)90201-B)>

DUYSINX, P, AND P TOSSINGS, 'Structural Optimization', *Structural Optimization*, 10.October (2011), 1–10

ELSEVIER B.V., 'Scopus', 2019 <<https://www.scopus.com/>> [accessed 8 March 2019]

ESFANDIARI, M.J. URGESSA, G.S.-SHEIKHOLAFERIN, S. MANSHADI DEHGHAN, S.H., 'Optimization of Reinforced Concrete Frames Subjected to Historical Time-History Loadings Using DMP SO Algorithm', *Structural and Multidisciplinary Optimization*, 58.5 (2018), 2119–34 <<https://doi.org/10.1007/s00158-018-2027-y>>

FARMER, J.DOYNE, NORMAN H PACKARD, AND ALAN S PERELSON, 'The Immune System, Adaptation, and Machine Learning', *Physica D: Nonlinear Phenomena*, 22.1–3 (1986), 187–204 <[https://doi.org/10.1016/0167-2789\(86\)90240-X](https://doi.org/10.1016/0167-2789(86)90240-X)>

FENG, Y.J., L. YU, AND G.L. ZHANG, 'Ant Colony Pattern Search Algorithms for Unconstrained and Bound Constrained Optimization', *Applied Mathematics and Computation*, 191.1 (2007), 42–56 <<https://doi.org/10.1016/j.amc.2006.09.142>>

FEO, THOMAS A, AND MAURICIO G.C RESENDE, 'A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem', *Operations Research Letters*, 8.2 (1989), 67–71 <[https://doi.org/10.1016/0167-6377\(89\)90002-3](https://doi.org/10.1016/0167-6377(89)90002-3)>

FERNÁNDEZ-VARGAS, J. A., AND A. BONILLA-PETRICIOLET, 'Desarrollo de Un Algoritmo de Optimización Global En Colonias de Hormigas Con Selección de Región Factible Para Espacios Continuos', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 30.3 (2014), 178–87 <<https://doi.org/10.1016/j.rimni.2013.06.006>>

FOGEL, LAWRENCE J., ALVIN J. OWENS, AND MICHAEL J. WALSH, *Artificial Intelligence through Simulated Evolution*, 1st edn (Oxford: John Wiley & Sons, Inc., 1966)

FRANKLIN, CHENG Y., AND Z. TRUMAN KEVIN, *Structural Optimization Dynamic and Seismic Applicatios*, ed. by Spon Press, 1st edn (New York, 2010)

Gandom, Amir Hossein, Xin-She Yang, Siamak Talatahari, and Amir Hossein Alavi, *Metaheuristic Applications in Structures and Infrastructures*, 1st edn (London: Elsevier, 2013) <<https://doi.org/10.1016/C2011-0-08778-1>>

GHOLIZADEH, SAEED, AND VAHID ALIGHOLIZADEH, 'Reliability-Based Optimum Seismic Design of RC Frames by a Metamodel and Metaheuristics', *Structural Design of Tall and Special Buildings*, 28.1 (2019), 1–19 <<https://doi.org/10.1002/tal.1552>>

GLOVER, FRED, 'HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS', *Decision Sciences*, 8.1 (1977), 156–66 <<https://doi.org/10.1111/j.1540-5915.1977.tb01074.x>>

GLOVER, FRED, AND CLAUDE MCMILLAN, 'The General Employee Scheduling Problem. An Integration of MS and AI', *Computers & Operations Research*, 13.5 (1986), 563–73 <[https://doi.org/10.1016/0305-0548\(86\)90050-X](https://doi.org/10.1016/0305-0548(86)90050-X)>

HAGE, RÉ-MI, ILIGE HAGE, CHADY GHNATIOS, I.S. JAWAHIR, AND RAMSEY HAMADE, 'Optimized Tabu Search Estimation of Wear Characteristics and Cutting Forces in Compact Core Drilling of Basalt Rock Using PCD Tool Inserts', *Computers & Industrial Engineering*, 2019 <<https://doi.org/10.1016/j.cie.2019.07.049>>

HANSE, P., AND MLADNOVIC M., 'Variable Neighborhood Decomposition Search', *Journal of Heuristics*, 7 (2001), 330–50

HANSEN, N., AND A. OSTERMEIER, 'Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation', in *Proceedings of IEEE International Conference on Evolutionary Computation* (IEEE, 1996), pp. 312–17 <<https://doi.org/10.1109/ICEC.1996.542381>>

HARE, WARREN, JULIE NUTINI, AND SOLOMON TESFAMARIAM, 'A Survey of Non-Gradient Optimization Methods in Structural Engineering', *Advances in Engineering Software*, 59 (2013), 19–28 <<https://doi.org/10.1016/j.advengsoft.2013.03.001>>

HERNÁNDEZ-DÍAZ, A. M., J. M. CECILIA, AND M. D. GARCÍA-ROMÁN, 'Ajuste de Modelos de Cortante En Hormigón Estructural Mediante Algoritmos Evolutivos',

Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria, 33.1–2 (2017), 52–57 <<https://doi.org/10.1016/j.rimni.2015.10.001>>

HIBBELER, R.C., *Analisis Estructural*, ed. by Pearson Educación de México, 8 th ed. (México: Pearson, 2012)

HILLIS, W.DANIEL, 'Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure', *Physica D: Nonlinear Phenomena*, 42.1–3 (1990), 228–34 <[https://doi.org/10.1016/0167-2789\(90\)90076-2](https://doi.org/10.1016/0167-2789(90)90076-2)>

HOLLAND, J.H., *Adaption in Natural and Artificial Systems*, 1st edn (Ann Arbor: University of Michigan Press, 1975)

KAR, ARPAN KUMAR, 'Bio Inspired Computing - A Review of Algorithms and Scope of Applications', *Expert Systems with Applications*, 59 (2016), 20–32 <<https://doi.org/10.1016/j.eswa.2016.04.018>>

KASSIMALI, A., *Matrix Analysis of Structures*, 2nd ed. (Stamford: Cengage Learning, 2012)

KAVEH, A., AND A. F. BEHNAM, 'Design Optimization of Reinforced Concrete 3D Structures Considering Frequency Constraints via a Charged System Search', *Scientia Iranica*, 20.3 (2013), 387–96 <<https://doi.org/10.1016/j.scient.2012.11.017>>

KENNEDY, J., AND R. EBERHART, 'Particle Swarm Optimization', in *Proceedings of ICNN'95 - International Conference on Neural Networks* (IEEE, 1995), IV, 1942–48 <<https://doi.org/10.1109/ICNN.1995.488968>>

KIRKPATRICK, S., C. D. GELATT, AND M. P. VECCHI, 'Optimization by Simulated Annealing', *Science*, 220.4598 (1983), 671–80 <<https://doi.org/10.1126/science.220.4598.671>>

KOZA, JOHN R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, 1st edn (Cambridge: MIT Press, 1992)

LEE, C., AND J. AHN, 'Flexural Design of Reinforced Concrete Frames by Genetic Algorithm', *Journal of Structural Engineering*, 129.6 (2003), 762–74 <[https://doi.org/10.1061/\(ASCE\)0733-9445\(2003\)129:6\(762\)](https://doi.org/10.1061/(ASCE)0733-9445(2003)129:6(762))>

LOUCERA, CARLOS, ANDRES IGLESIAS, AND GÁLVEZ AKEMI, 'Simulated Annealing and Natural Neighbor for Rational Bézier Surface Reconstruction from Scattered Data Points', in *International Conference on Harmony Search Algorithm* (Bilbao: Javier del Ser, 2017), pp. 354–63

MARTIN, OLIVIER, STEVE W OTTO, AND EDWARD W FELTEN, 'Large-Step Markov Chains for the TSP Incorporating Local Search Heuristics', *Operations Research Letters*, 11.4 (1992), 219–24 <[https://doi.org/10.1016/0167-6377\(92\)90028-2](https://doi.org/10.1016/0167-6377(92)90028-2)>

MARTÍNEZ-FRUTOS, J., AND P. MARTÍ, 'Diseño Óptimo Robusto Utilizando Modelos Kriging: Aplicación Al Diseño Óptimo Robusto de Estructuras Articuladas', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 30.2 (2014), 97–105 <<https://doi.org/10.1016/j.rimni.2013.01.003>>

MERGOS, PANAGIOTIS E., 'Efficient Optimum Seismic Design of Reinforced Concrete Frames with Nonlinear Structural Analysis Procedures', *Structural and Multidisciplinary Optimization*, 58.6 (2018), 2565–81 <<https://doi.org/10.1007/s00158-018-2036-x>>

MILLÁN PÁRAMO, C., AND O. BEGAMBRE CARRILLO, 'Solución de Problemas de Optimización Topológica Empleando El Algoritmo Simulated Annealing Modificado', *Revista Internacional de Metodos Numericos Para Calculo y Diseno En Ingenieria*, 32.2 (2016), 65–69 <<https://doi.org/10.1016/j.rimni.2014.11.005>>

MLADENović, N., AND P. HANSEN, 'Variable Neighborhood Search', *Computers & Operations Research*, 24.11 (1997), 1097–1100 <[https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)>

MORTEZAIE, HAMID, AND FREYDOON REZAIE, 'Optimization and Evaluation of Three-Dimensional High-Rise Concrete Structure Equipped by TMD with Considering the Soil-Structure Interaction Effect', *Journal of Vibroengineering*, 20.5 (2018), 2111–27 <<https://doi.org/10.21595/jve.2017.18628>>

NABID, NEDA, IMAN HAJIRASOULIHA, AND MIHAIL PETKOVSKI, 'Performance-Based Optimisation of RC Frames with Friction Wall Dampers Using a Low-Cost Optimisation Method', *Bulletin of Earthquake Engineering*, 16.10 (2018), 5017–40 <<https://doi.org/10.1007/s10518-018-0380-2>>

NIGDELI, SINAN MELIH, GEBRAIL BEKDA, AND XIN-SHE YANG, *Metaheuristics and Optimization in Civil Engineering*, 2016, vii <<https://doi.org/10.1007/978-3-319-26245-1>>

NIÑO ALVAREZ, LUIS HUMBERTO, AND EDER JAIR DURÁN CÁCERES, 'Optimización Del Dimensionamiento Estructural de Un Portico Plano de Concreto Reforzado, Destinado a Almacenamiento de Archivos (Trabajo de Grado)' (Universidad Industrial de Santander- Escuela de Ingenieria Civil, 2016)

PAYÁ, IGNACIO, VÍCTOR YEPES, JUAN CLEMENTE, FERNANDO GONZÁLEZ, IGNACIO PAYÁ ZAFORTEZA, AND VÍCTOR YEPES PIQUERAS, 'Optimización Heurística de Pórticos de Edificación de Hormigón Armado', *Métodos Numéricos Para Cálculo y Diseño En Ingeniería: Revista Internacional*, 22.3 (2006), 241–60

POLYA, G., *How to Solve It: A New Aspect of Mathematical Method, Optimización Heuristic*, 2nd edn (New York: Penguin Science, 1990)

REYNOLDS, ROBERT G., 'An Introduction to Cultural Algorithms', in *Evolutionary Programming --- Proceedings of the Third Annual Conference*, ed. by Anthony

FOGEL, V. SEBALD, AND J. LAWRENCE, 1st edn (San Diego, CA, USA: World Scientific Press, 1994), pp. 131–39

SAKA, M. P., O. HASANÇEBİ, AND Z. W. GEEM, ‘Metaheuristics in Structural Optimization and Discussions on Harmony Search Algorithm’, *Swarm and Evolutionary Computation*, 28 (2016), 88–97
<<https://doi.org/10.1016/j.swevo.2016.01.005>>

SCHWEFEL, H-P., *Kybernestische Evolution Als Strategie Der Experimente Forschung in Der Stromungstechnik* (Berlin, 1965)

SCIMAGO LAB, ‘SCIMAGO JOURNAL & COUNTRY RANK’, 2019
<<https://www.scimagojr.com/>> [accessed 20 July 2019]

SERPIK, I. N., I. V. MIRONENKO, AND V. I. AVERCHENKOV, ‘Algorithm for Evolutionary Optimization of Reinforced Concrete Frames Subject to Nonlinear Material Deformation’, *Procedia Engineering*, 150 (2016), 1311–16
<<https://doi.org/10.1016/j.proeng.2016.07.304>>

SHLESINGER, MICHAEL F., AND JOSEPH KLAFTER, ‘Lévy Walks Versus Lévy Flights’, in *On Growth and Form* (Dordrecht: Springer Netherlands, 1986), pp. 279–83
<https://doi.org/10.1007/978-94-009-5165-5_29>

SIARRY, PATRICK, *Metaheuristics*, ed. by Patrick Siarry, 1st edn (Cham: Springer International Publishing, 2016) <<https://doi.org/10.1007/978-3-319-45403-0>>

STORN, R., AND K. PRICE, *Differential Evolution-A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces* (Berkeley, 1995)

TALBI, EL-GHAZALI, *Metaheuristics*, 1st edn (Hoboken: John Wiley & Sons, Inc., 2009)

VOUDOURIS, C., 'Guided Local Search — an Illustrative Example in Function Optimisation', *BT Technology Journal*, 16.3 (1995), 46–50

WODRICH, M., AND C. BILCHEV, 'Cooperative Distributed Search: The Ant's Way', 2000

YANG, X.S., Z. CUI, R. XIAO, AND A.H. GANDOMI, *Swarm Intelligence and Bio-Inspired Computation*, ed. by 1 (Elsevier, 2013) <<https://doi.org/10.1016/C2012-0-02754-8>>

YONEZAWA, Y., AND T. KIKUCHI, 'Ecological Algorithm for Optimal Ordering Used by Collective Honey Bee Behavior', in *MHS'96 Proceedings of the Seventh International Symposium on Micro Machine and Human Science* (IEEE, 1996), pp. 249–56 <<https://doi.org/10.1109/MHS.1996.563432>>

YUN-LONG, G A O, AND Y A N PENG, *Harmony Search Algorithm*, 1st edn (Warsaw, 2016), xxxi <<https://doi.org/10.13195/j.kzyjc.2015.0352>>

ZAVALA, GUSTAVO, ANTONIO J. NEBRO, FRANCISCO LUNA, AND CARLOS A. COELLO COELLO, 'Structural Design Using Multi-Objective Metaheuristics. Comparative Study and Application to a Real-World Problem', *Structural and Multidisciplinary Optimization*, 53.3 (2016), 545–66 <<https://doi.org/10.1007/s00158-015-1291-3>>

ZAVALA, GUSTAVO R., ANTONIO J. NEBRO, FRANCISCO LUNA, AND CARLOS A. COELLO COELLO, 'A Survey of Multi-Objective Metaheuristics Applied to Structural Optimization', *Structural and Multidisciplinary Optimization*, 49.4 (2014), 537–58 <<https://doi.org/10.1007/s00158-013-0996-4>>

ZHAN, ZHI-HUI, JUN ZHANG, YUN LI, AND YU-HUI SHI, 'Orthogonal Learning Particle Swarm Optimization', *IEEE Transactions on Evolutionary Computation*, 15.6 (2011), 832–47 <<https://doi.org/10.1109/TEVC.2010.2052054>>