

**Estudio comparativo sobre la aplicación de los algoritmos *PSO*
y *SQP* en la optimización de pórticos planos de concreto
armado.**

Trabajo de grado para optar al título de Magister en Ingeniería Civil

Estudiante:

Ing. Víctor Antonio Lapadula Sequera.

Director:

Ing. Jesús Daniel Villalba Morales, D.Sc.



PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA CIVIL
MAESTRÍA EN INGENIERÍA CIVIL
ÉNFASIS EN ESTRUCTURAS
Bogotá D.C., 2014

APROBACIÓN

El Trabajo de grado con título “Estudio comparativo sobre la aplicación de los algoritmos PSO y SQP en la optimización de pórticos planos de concreto armado.”, desarrollado por el estudiante Víctor Antonio Lapadula Sequera, en cumplimiento de uno de los requisitos depuestos por la Pontificia Universidad Javeriana, Facultad de Ingeniería, Departamento de Ingeniería civil, para optar el Título de Magister en ingeniería Civil, fue aprobado por:

Ing. Jesús Daniel Villalba Morales
Director de trabajo de grado

Ing. Federico Alejandro Nuñez Moreno
Jurado 1

Ing. Oscar Alfredo Ardila Giraldo
Jurado 2

Estudio comparativo sobre la aplicación de los algoritmos *PSO* y *SQP* en la optimización de pórticos planos de concreto armado.

Víctor Antonio Lapadula Sequera.

La Pontificia Universidad Javeriana, no es responsable por los conceptos emitidos por los autores-investigadores del presente trabajo, por lo cual son responsabilidad absoluta de sus autores y no comprometen la idoneidad de la institución ni de sus valores.

RESUMEN

Esta investigación presenta la comparación de las técnicas *SQP* (*Sequential Quadratic Programming*) y *PSO* (*Particle Swarm optimization*) al ser aplicadas al problema de optimización de pórticos planos de concreto armado. El *SQP* y el *PSO* son algoritmos de optimización provenientes de filosofías diferentes. El primero es una técnica clásica basada en cálculo diferencial, mientras que el segundo es una meta-heurística inspirada en comportamientos observados en la naturaleza. Como función objetivo de la optimización se utilizó el costo inicial de construcción del pórtico. El desempeño de los algoritmos se midió en función de la reducción del costo de construcción en las estructuras analizadas y evaluando cuál de los dos alcanzaba diseños con menor costo inicial sin violar las restricciones normativas. Se utilizaron 3 pórticos de 4, 8 y 12 pisos para evaluar el desempeño de ambos algoritmos. Las restricciones se tomaron de la Norma Sismo-Resistente Colombiana (NSR-10) para pórticos con capacidad de disipación moderada de energía (DMO). A diferencia de otras metodologías de optimización, se plantearon 6 variables de diseño para las columnas y 11 para las vigas, con el fin evaluar el desempeño de los algoritmos ante problemas con múltiples variables, obteniéndose buenos resultados. Se encontró que el *PSO* logró mejorar los resultados arrojados por el *SQP* en un 2.2% para el pórtico 4 pisos, 13.3% para el pórtico de 8 pisos y 16 % para el pórtico 12 pisos. En estos resultados se identificó la tendencia que tiene el *SQP* a quedar atrapado en mínimos locales lo cual es una desventaja. También, el algoritmo *PSO* utilizado logró reducciones de la función objetivo mayores al 30% para los tres pórticos, lo cual mejora los resultados obtenidos por otras investigaciones. Se concluye que de los dos algoritmos, el *PSO* tuvo mejor desempeño, recomendándose su uso para problemas de optimización con funciones objetivo más complejas en el área de diseño estructural.

Palabras Clave: Optimización estructural, Pórticos de concreto, PSO, SQP.

TABLA DE CONTENIDO

	Pág.
1. Introducción.....	10
1.1 Planteamiento del problema y justificación	10
1.2 Objetivos.....	11
1.2.1 Objetivo General:	11
1.2.1 Objetivos Específicos:	12
1.3 Síntesis.....	12
2. Marco Teórico.....	13
2.1 Diseño por Optimización.....	13
2.1.1 Función objetivo	13
2.1.2 Variables del problema y el espacio de soluciones	13
2.1.3 Restricciones del problema de optimización	15
2.2 Particle Swarm Optimization (PSO).....	15
2.3 Sequential Quadratic Programming (SQP)	17
2.4 Pórticos Planos.....	18
2.5 Método de la resistencia última en el diseño de concreto armado.....	19
2.6 Diseño a flexión de vigas de concreto armado	19
2.7 Diseño de elementos a flexo-compresión	21
3. Metodología de optimización.....	23
3.1 Planteamiento de la Metodología de Optimización.....	23
3.1.1 Variables del problema	24
3.1.2 Función Objetivo	26
3.1.2 Restricciones del problema.....	27
3.2 Algoritmo Particle Swarm Optimization (PSO).....	30
3.3 Algoritmo Sequential Quadratic Programming (SQP)	35
3.4 Análisis de las estructuras.....	36
4. Ejemplos Numéricos.....	38
4.1 Ejemplos Numéricos utilizados	38
4.1 Efecto de la función objetivo penalizada en la búsqueda del <i>PSO</i>	40

4.2 Criterio de parada del <i>PSO</i>	42
4.2 Comparación del desempeño del <i>PSO</i> al generar la primera población dentro y fuera del espacio factible	44
4.3 Comparación de los resultados encontrados por el <i>PSO</i> y el <i>SQP</i>	45
4.3.1 Pórtico 1:	45
4.3.2 Pórtico 2:	53
4.3.3 Pórtico 3:	59
4.4 Discusión General.....	64
5. Conclusiones	67
6. Referencias Bibliográficas.....	68

ÍNDICE DE FIGURAS

	Pág.
Figura 1. Representación de un problema restringido.	14
Figura 2. Representación del espacio de búsqueda de un problema con mínimos locales.....	14
Figura 3. Distribución de fuerzas en la viga en el momento de la falla a flexión	20
Figura 4. Condición de fuerza equivalente al momento.	21
Figura 5. Distribución de esfuerzos y deformaciones en el momento de la falla de la columna.	21
Figura 6. Esquema general de la metodología de optimización.....	23
Figura 7. Distribución del acero longitudinal en vigas.	24
Figura 8. Diagrama de Interacción Axial-Momento (P-M) para columnas.....	29
Figura 9. Estados de explotación y exploración del PSO.....	31
Figura 10. Estados de convergencia y jumping-out del PSO.	32
Figura 11. Comportamiento del factor evolutivo en una corrida de PSO.	32
Figura 12. Esquema para determinar los parámetros del algoritmo PSO en cada iteración.	34
Figura 13. Esquema de pasos ejecutados por el algoritmo SQP.....	36
Figura 14. Pórtico 1: 4 pisos y 3 luces.	38
Figura 15. Pórtico 2: 8 pisos y 3 luces.	39
Figura 16. Pórtico 3: 12 pisos y 3 luces.....	39
Figura 17. Convergencia del algoritmo de PSO para el pórtico 1.	41
Figura 18. Convergencia del algoritmo de PSO para el pórtico 2.	41
Figura 19. Convergencia del algoritmo de PSO para el pórtico 3.	41
Figura 20. Convergencia del algoritmo de PSO para el pórtico 1 variando el criterio de parada	42
Figura 21. Convergencia del algoritmo de PSO para el pórtico 2 variando el criterio de parada. ...	42
Figura 22. Convergencia del algoritmo de PSO para el pórtico 3 variando el criterio de parada. ...	43
Figura 23. Comparación de convergencia de los algoritmos para el pórtico 1	48
Figura 24. Relación capacidad/demanda en las columnas pórtico 1.	49
Figura 25. Relación capacidad/demanda para momento superior izquierdo en las vigas pórtico 1.49	49
Figura 26. Relación capacidad/demanda para momento inferior central en las vigas pórtico 1.	50
Figura 27. Relación capacidad/demanda para momento superior derecho en las vigas pórtico 1.	50
Figura 28. Relación capacidad/demanda para cortante nodo izquierdo en las vigas pórtico 1.....	51
Figura 29. Relación capacidad/demanda para cortante nodo derecho en las vigas pórtico 1.	51
Figura 30. Comparación de convergencia de los algoritmos para el pórtico 2	55
Figura 31. Relación capacidad/demanda para columnas del pórtico 2.	55
Figura 32. Relación capacidad/demanda para momento inferior central pórtico 2.	56
Figura 33. Relación capacidad/demanda para momento superior izquierdo pórtico 2.	56
Figura 34. Relación capacidad/demanda para momento superior derecho pórtico 2.	57
Figura 35. Relación capacidad/demanda para cortante nodo izquierdo pórtico 2.	57
Figura 36. Relación capacidad/demanda para cortante nodo derecho pórtico 2.	58
Figura 37. Relación capacidad/demanda para las columnas pórtico 3.....	61
Figura 38. Relación capacidad/demanda para momento inferior central pórtico 3.	62
Figura 39. Relación capacidad/demanda para momento superior izquierdo pórtico 3.	62
Figura 40. Relación capacidad/demanda para momento superior derecho pórtico 3.	62

Figura 41. Relación capacidad/demanda para cortante en el nodo izquierdo pórtico 3..... 63
Figura 42. Relación capacidad/demanda para cortante en el nodo derecho pórtico 3. 63

ÍNDICE DE TABLAS

Tabla 1. Rangos de valores para las variables.	25
Tabla 2. Cambio en los valores de c_1 y c_2 según el estado evolutivo.	33
Tabla 3. Resultados del PSO para el Pórtico 1 generando la población inicial desde puntos factibles.	44
Tabla 4. Resultados del PSO para el Pórtico 1 generando la población inicial desde puntos no factibles.	44
Tabla 5. Resultados del SQP para el Pórtico 1 en 30 corridas.	45
Tabla 6. Resultados del PSO para el Pórtico 1 en 30 corridas.	46
Tabla 7. Dimensiones de columna y área de acero a colocar para el diseño óptimo del PSO.	52
Tabla 8. Dimensiones de columna y área de acero a colocar para el diseño óptimo del SQP.	52
Tabla 9. Dimensiones de vigas y áreas de acero a colocar para el diseño óptimo del PSO.	52
Tabla 10. Dimensiones de vigas y áreas de acero a colocar para el diseño óptimo del SQP.	52
Tabla 11. Resultados del SQP para el pórtico 2.	53
Tabla 12. Resultados del PSO para el pórtico 2.	54
Tabla 13. Dimensiones de columna y área de acero a colocar para el diseño óptimo del PSO.	58
Tabla 14. Dimensiones de columna y área de acero a colocar para el diseño óptimo del SQP.	58
Tabla 15. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del PSO.	59
Tabla 16. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del SQP.	59
Tabla 17. Resultados del SQP para el pórtico 3.	59
Tabla 18. Resultados del PSO para el pórtico 3.	60
Tabla 19. Dimensiones de columna y área de acero a colocar para el diseño óptimo del PSO.	63
Tabla 20. Dimensiones de columna y área de acero a colocar para el diseño óptimo del SQP.	64
Tabla 21. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del PSO.	64
Tabla 22. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del SQP.	64
Tabla 23. Resultados reportados por Kaveh & Sabzi, (2011) para el pórtico 1.	66
Tabla 24. Resultados reportados por Kaveh & Sabzi, (2011) para el pórtico 2.	66
Tabla 25. Resultados reportados por Kaveh & Sabzi, (2011) para el pórtico 3.	66

1. Introducción

1.1 Planteamiento del problema y justificación

El proceso de diseño de cualquier sistema consiste en hallar los valores de las variables involucradas con el fin de cumplir unos requisitos de funcionalidad, estética, resistencia y economía. El diseño se puede abordar desde dos enfoques: el enfoque tradicional en el cual el diseñador cambia las variables en cada iteración (guiado por lo general por su experiencia), y (ii) el diseño por optimización (Arora, 2012). En este último, el diseño se transforma en un proceso de ir cambiando los valores de las variables siguiendo algún criterio, hasta encontrar un punto óptimo donde se hace máxima o mínima una función objetivo, que sirve de comparación entre diseños. La optimización tiene la ventaja de que el procedimiento de cambiar valores de las variables lo hace un algoritmo matemático, lo cual permite comparar muchas alternativas en corto tiempo. En cambio en el diseño tradicional, el diseñador podrá comparar 2 o 3 alternativas, y sin tener la certeza de que el diseño escogido sea el mejor (Flores & Rodríguez, 2010).

El diseño estructural busca determinar las propiedades de los elementos (Secciones transversales, materiales, etc.) con el fin de que cumplan unos requisitos de funcionalidad y resistencia. En el caso de edificios de concreto armado, a diferencia de los de acero estructural donde se tienen bases de datos de secciones, el rango de combinaciones entre las dimensiones de concreto y el área de acero que se pueden escoger para que un elemento cumpla con requisitos, por ejemplo, de resistencia es muy amplio, por lo que es difícil que un diseñador pueda evaluar todos los posibles diseños para un elemento. Si bien para una estructura sencilla el ingeniero con experiencia puede plantear soluciones competitivas, esto se hace más difícil cuando la complejidad del problema aumenta por lo que es importante contar con una herramienta de optimización confiable.

Una parte importante en el proceso de optimización es el algoritmo numérico. En la actualidad tales algoritmos se pueden clasificar en forma general, en técnicas clásicas y meta-heurísticas. A las primeras se les da ese nombre porque se basan en diferenciación, es decir, en determinar los gradientes tanto de la función objetivo como de las restricciones del problema en cada iteración (Nocedal & Wright, 2006). Las meta-heurística son técnicas que se han venido desarrollando en los últimos 20 años (Talbi, 2009), y en general, se basan en procesos observados en la naturaleza.

Dentro de las técnicas clásicas se encuentran el *SQP* (*Sequential Quadratic Programming*), el cual es uno de los algoritmos más efectivos para ser aplicado a problemas de optimización restringida, con funciones objetivo no lineales (Nocedal & Wright, 2006). Este método fue aplicado por Flores & Rodríguez, (2010) en la optimización de secciones de pórticos de concreto armado, demostrándose su aplicabilidad a este problema. Sin embargo, en el trabajo mencionado cada elemento fue tratado como un problema independiente, ignorando las condiciones que debe cumplir un pórtico como sistema (derivadas, relaciones de áreas de aceros y tamaños entre secciones, etc.).

Por otro lado, una de las técnicas meta-heurísticas, es el *PSO* (*Particle Swarm Optimization*), el cual está inspirado en la inteligencia de enjambres. En su algoritmo, se imita el comportamiento social observado en grupos de animales, como peces y aves, por ejemplo, para buscar alimento. En investigaciones anteriores se ha mostrado la aplicabilidad del *PSO* en el problema de optimización de pórticos de concreto (Kaveh & Sabzi, 2011) y (Gharehbaghi & Fadaee, 2012).

En las investigaciones mencionadas, la función objetivo utilizada para guiar la optimización siempre fue el costo asociado a la construcción del pórtico. Ahora bien, el costo total de un pórtico no es nada más la inversión inicial sino también se debe tomar en cuenta el costo a largo plazo (costo por ciclo de vida), en el cual se incluye la inversión requerida para reparar la estructura en caso de que ocurra, por ejemplo, un evento sísmico probable. Incluir esta parte en el problema de optimización, no es sencillo, y requiere que el algoritmo de optimización tenga un buen desempeño.

Por lo antes mencionado, en este proyecto se evaluará el desempeño de las técnicas *PSO* y *SQP* al ser aplicados al problema de optimización de pórticos planos de concreto armado, usando como función objetivo el costo inicial del pórtico, para así determinar cuál de los dos enfoques puede ser aplicado más adelante en problemas que incluyan, tanto el costo inicial como el costo a largo plazo (optimización multi-objetivo).

1.2 Objetivos

1.2.1 Objetivo General:

Comparar los valores de costo mínimo y tiempo computacional, obtenidos por los algoritmos *PSO* y *SQP* al aplicarlos al problema de optimización de pórticos planos de concreto armado.

1.2.1 Objetivos Específicos:

- Definir las características del problema de optimización de pórticos planos de concreto armado.
- Implementar los algoritmos de *PSO* y *SQP*.
- Integrar la formulación del problema y los algoritmos de optimización con un módulo de análisis estructural de pórticos planos de concreto armado.
- Determinar las soluciones de costo mínimo de pórticos planos de concreto reforzado a partir de los algoritmos *PSO* y *SQP*.

1.3 Síntesis

En el Capítulo 2 se presentan los fundamentos teóricos en los que se basa esta investigación, dándose una descripción del proceso de optimización y sus partes, y se plantean los conceptos que hay detrás de los algoritmos *PSO* y *SQP*. También se explican algunos aspectos del análisis y diseño de pórticos de concreto armado. En el Capítulo 3, se describe las partes de la metodología de optimización planteada, como lo son, la función objetivo, las variables consideradas, las restricciones y los algoritmos utilizados. En el Capítulo 4, se muestran los resultados de aplicar las técnicas *PSO* y *SQP* a tres pórticos de concreto, y se comparan estos resultados con los obtenidos en otras investigaciones y finalmente en el Capítulo 5 se presentan las conclusiones obtenidas tras el desarrollo de esta investigación.

2. Marco Teórico

2.1 Diseño por Optimización

El diseño por optimización consiste en la obtención de la solución de un problema a través de la aplicación de una formulación matemática que permite evaluar y comparar diversas soluciones con el fin de hallar una respuesta óptima. “*El proceso de optimización permite la toma de decisiones, ya que en él se van descartando soluciones hasta reducirlas a una sola, la mejor*” (Wilde & Biegthler, 1976, p. 19).

Un problema de diseño, visto como uno de optimización, requiere la definición de la función objetivo, las variables involucradas en el problema y las restricciones de diseño. A continuación, se describen estos aspectos con más detalle.

2.1.1 Función objetivo

La función objetivo es el criterio utilizado durante el proceso de optimización para hacer la comparación entre diseños. Por lo general, la optimización se plantea en términos de minimizar la función objetivo, la cual puede ser tiempo de ejecución, costo de elaboración, etc. Así, el mejor diseño será, por ejemplo, en la construcción, el que tenga menor inversión inicial, o en la planeación de viajes, la ruta por la cual se llegue más rápido a un punto determinado.

2.1.2 Variables del problema y el espacio de soluciones

Las variables de diseño son las características del modelo que representa el problema real, las cuales al cambiar, modifican el valor de la función objetivo (Payá et al, 2006). Cuando a cada variable se le da un valor determinado, se forma una solución para el problema. Así, las muchas soluciones que puede tener un problema forman el espacio de soluciones o de búsqueda. Cuando en el problema de optimización se incluyen restricciones, el espacio de búsqueda queda dividido en el espacio factible y el no factible. En el primero se encuentran todas aquellas soluciones que cumplen con las restricciones del problema, y en el segundo todas aquellas que dejan de ser validas, ya que violan al menos uno de límites impuestos.

Lo anterior se ejemplifica en la Figura 1. Supongamos que un problema está representado por la ecuación de una parábola $y = x^2 - 20$, y que los valores de x están limitados al rango $-15 \leq x \leq 15$. Entonces, mediante la restricción anterior, queda definido el espacio factible. El mínimo de la función está en $x = 0$, y este punto a la vez cumple con la restricción del problema. Cualquier punto $x > 15$ ó $x < -15$, estará fuera del espacio factible.

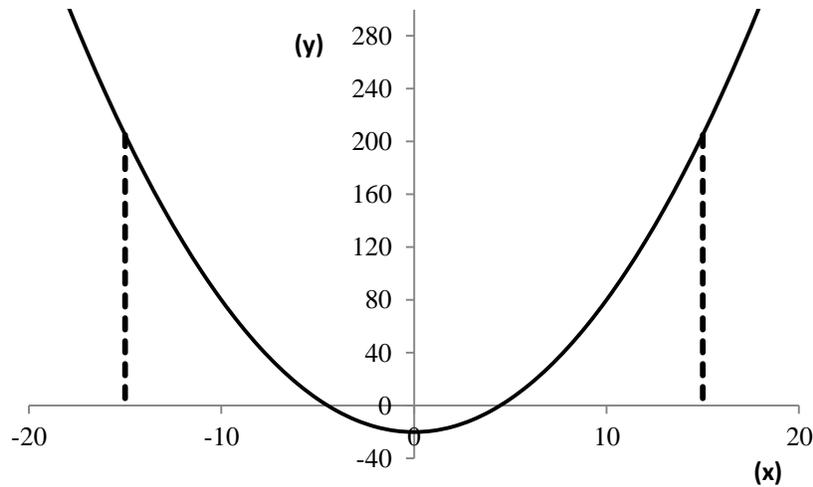


Figura 1. Representación de un problema restringido.

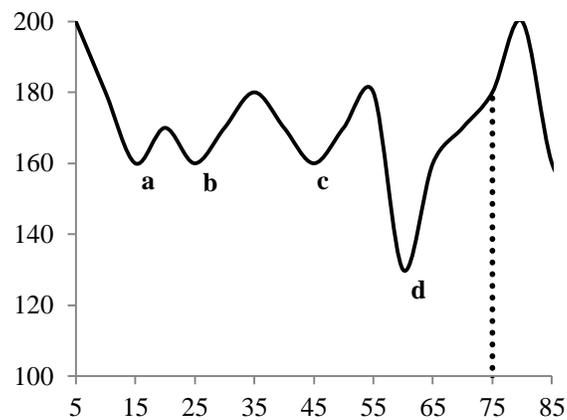


Figura 2. Representación del espacio de búsqueda de un problema con mínimos locales.

Algunos problemas de optimización presentan espacios de búsqueda complejos, con múltiples mínimos locales (puntos **a**, **b** y **c** en la Figura 2). Estos espacios de búsqueda son difíciles de explorar para algunos algoritmos (Belegundu & Chandrupatla, 2011). Para el problema representado en la Figura 2, el mínimo

está en el punto \mathbf{d} , y también representa un problema de optimización restringida con $0 \leq x \leq 75$.

2.1.3 Restricciones del problema de optimización

Las restricciones de un problema de optimización son funciones que, al igual que la función objetivo, dependen de la solución x . La notación más común para identificarlas es:

$$\begin{aligned} h_i(x) &= 0 && \text{con } i \text{ desde } 1 \text{ hasta } k \\ g_i(x) &\leq 0 && \text{con } i \text{ desde } 1 \text{ hasta } m \end{aligned}$$

Siendo k el número de restricciones de igualdad y m el número de restricciones de desigualdad en el problema. Una restricción puede ser activa, e-activa, inactiva o violada dependiendo de su valor en el punto solución x . Las restricciones de igualdad pueden ser solo activas cuando se cumple que $h_i(x) = 0$, o inactivas cuando $h_i(x) \neq 0$. Por otro lado, las restricciones de desigualdad son activas cuando se cumple que $g_i(x) = 0$, e-activas cuando $g_i(x) = e$, siendo e un número pequeño cercano a cero, inactivas cuando $g_i(x) < 0$, o violadas cuando $g_i(x) > 0$ (Ruszczynski, 2006). Como se verá en la metodología, todas las restricciones del problema planteado son de desigualdad.

2.2 Particle Swarm Optimization (PSO)

El *PSO* es un método de optimización que tiene su origen en los intentos por simular el movimiento de ciertos grupos de animales como aves y peces para buscar alimento. Fue presentado por primera vez por Kennedy & Eberhart, en 1995, y a partir de esa fecha se han publicado diferentes modificaciones al algoritmo original. En forma general, un enjambre (*swarm*), está constituido por un número N de partículas o individuos, los cuales se encuentran moviéndose en el espacio de búsqueda. Cada posición de una partícula representa una posible solución al problema. En cada iteración, las partículas se van moviendo de una posición a otra, es decir, se genera un cambio en la solución que cada una representa. Las modificaciones en la posición de la partícula i , durante la iteración k , se hacen en función de la mejor posición encontrada tanto por todo el enjambre como por la partícula hasta esa iteración.

De esta forma, se obtiene que el vector solución para cada una de las partículas se actualiza haciendo:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (1)$$

$$v_i^{k+1} = w * v_i^k + C_1 * r_1 * (P_i^k - x_i^k) + C_2 * r_2 * (G^k - x_i^k) \quad (2)$$

Donde x_i^{k+1} representa la nueva posición de la partícula i en la iteración $k+1$, v_i^{k+1} es el cambio de una posición a otra (también se le llama velocidad), w se conoce como factor inercial, C_1 es el coeficiente cognitivo, C_2 es el coeficiente social, y r_1 y r_2 son dos números aleatorios que se generan uniformemente entre 0 y 1, P_i^k es la mejor solución encontrada por la partícula i hasta la iteración k y G^k es la mejor solución encontrada por todo el enjambre (Talbi, 2009). También el algoritmo de *PSO* se puede trabajar de forma local, dividiendo el enjambre de partículas en grupos que realizan la búsqueda de forma independiente entre sí. De esta forma se tienen varios G^k , uno por cada grupo de partículas (Parsopoulos & Vrahatis, 2007).

En las ecuaciones (1) y (2), se identifican tres parámetros dentro del algoritmo, los cuales son el peso inercial w , el coeficiente cognitivo C_1 y el social C_2 . El primero busca controlar la influencia de la velocidad anterior v_i^k en la nueva velocidad v_i^{k+1} . Los coeficientes cognitivo y social, representan la importancia que le da cada partícula tanto a la mejor solución encontrada por ella P_i^k , como a la mejor encontrada por todo el enjambre G_i^k . La diferencia entre las versiones de *PSO* está en la forma en cómo se modifican estos parámetros durante las iteraciones. En algunos trabajos se dejan como constantes (Gharehbaghi & Fadaee, 2012), mientras que en otros se van modificando según las condiciones de la búsqueda (Zhan et al, 2009), (Ratnaweera & Halgamuge, 2004), (Tsoulos & Stavrakoudis, 2010), (Shi & Eberhart, 1998), (Arumugam & Rao, 2008). A estos últimos se les conoce como *PSO* auto-adaptativos, ya que son capaces de “adaptar” por sí mismos los parámetros del algoritmo en cada iteración, de acuerdo a lo que sea más conveniente para la convergencia. Un criterio de adaptación es por ejemplo variar de forma lineal los parámetros C_1 y C_2 a medida que avanzan las iteraciones (Ratnaweera & Halgamuge, 2004). Lo anterior se hace siguiendo la idea de que en las primeras iteraciones se requiere mayor exploración del espacio de búsqueda y para esto, las partículas se deben mantener dispersas, por lo cual se le da un mayor valor a C_1 con respecto a C_2 . Por otro lado, en las últimas iteraciones se aumenta el valor de C_2 y se disminuye C_1 haciendo que las partículas tiendan hacia la mejor solución encontrada por el enjambre incrementándose la búsqueda en esa parte del espacio (explotación). En el enfoque mencionado, los valores de C_1 y C_2 son los

mismos en cada partícula, sin embargo, en 2009, Guo & Chen presentaron una variación del algoritmo de *PSO* donde a cada partícula modifica sus parámetros, dando también buenos resultados.

En el presente trabajo se utilizó el algoritmo presentado por Zhan et al., (2009), en el cual los parámetros C_1 , C_2 y w son modificados en función del estado evolutivo del algoritmo. En el Capítulo 3 se da una mayor descripción de este método.

2.3 Sequential Quadratic Programming (SQP)

La programación cuadrática secuencial es una técnica de optimización clásica basada en el cálculo del gradiente de la función objetivo y de las restricciones. Se puede decir que la técnica se divide en dos sub-problemas de optimización, uno para determinar la dirección de búsqueda, y otro para calcular el paso de una iteración a otra (Arora, 2012). Su nombre se debe a que en cada iteración, se resuelve un problema cuadrático, el cual se establece en función del gradiente de la función objetivo y de las restricciones. El problema cuadrático se puede plantear como (Nocedal & Wright, 2006):

$$\begin{aligned} \text{minimizar: } & f_k + \nabla f_k^T * p + 0.5 * p^T * \nabla^2 f_k^T * p \quad (3) \\ \text{sujeto a: } & \nabla c_i^T * p + c_i \geq 0 \quad ; \quad \text{para } i = 1 \text{ hasta } m. \end{aligned}$$

Donde f_k es el valor de la función objetivo, ∇f_k^T es el vector de gradiente de la función objetivo, $\nabla^2 f_k^T$ es matriz Hessiana (Segunda derivadas) de la función objetivo, c_i y ∇c_i^T son el valor de la restricción i y su gradiente, todas evaluadas en el punto solución de la iteración actual x^k . p es un vector que representa la dirección de cambio en la solución de cada iteración, y se calcula a partir de la solución del problema de optimización planteado en (3).

Una vez que se determina la dirección de búsqueda, es necesario calcular cuánto se debe mover la solución sobre la dirección ya encontrada (tamaño del paso α_k). Para ello, el algoritmo itera hasta que consigue un valor de α_k que logra disminuir una función descendente que puede ser la misma función objetivo del problema (Arora, 2012). De esta forma, la nueva solución en la iteración $k+1$ se obtiene haciendo:

$$x^{k+1} = x^k + \alpha^k * d^k$$

2.4 Pórticos Planos

Un pórtico plano es un sistema estructural formado por vigas y columnas, sometido a cargas contenidas en el mismo plano del pórtico, simplificándose el análisis de los elementos. El plano considerado es el de simetría de los elementos con respecto a la sección transversal, por lo que no se consideran efectos por torsión. (Carballo et al, 2006). En el análisis estructural de este sistema se consideran 3 grados de libertad por nodo (contenidos todos en el plano del pórtico). Si se tienen dos nodos por elemento, las matrices de rigidez de estos quedan de 6x6. Para la estructura la matriz de rigidez queda de 3 x n, siendo n el número de nodos en el pórtico (Uribe E., 2000). Las matrices de rigidez utilizadas para el análisis de los pórticos son:

Para elementos viga:

$$\begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (4)$$

Para elementos columna:

$$\begin{bmatrix} \frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} & -\frac{12EI}{L^3} & 0 & -\frac{6EI}{L^2} \\ 0 & \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 \\ -\frac{6EI}{L^2} & 0 & \frac{4EI}{L} & \frac{6EI}{L^2} & 0 & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} & \frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} \\ 0 & -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 \\ -\frac{6EI}{L^2} & 0 & \frac{2EI}{L} & \frac{6EI}{L^2} & 0 & \frac{4EI}{L} \end{bmatrix} \quad (5)$$

Donde:

A , I , L y E representan el área e inercia de la sección transversal, longitud del elemento y módulo de elasticidad del material.

En el análisis de un pórtico se puede considerar que la relación existente entre fuerzas y desplazamientos es lineal y que las deformaciones de los elementos permanecen en el rango elástico (Tena, 2007).

2.5 Método de la resistencia última en el diseño de concreto armado

Acorde a lo establecido por Segura, (2011) “ *por el método de la resistencia última, también llamado método de la resistencia, el diseñador podrá estudiar el comportamiento de la estructura en el instante de la falla; por tanto, si este instante se hace suficientemente mayor que el de su trabajo para las cargas que soporta normalmente, se podrá tener un diseño con factores de seguridad apropiados.*

Para los elementos sometidos a flexión, las normas de diseño de concreto (ACI 318-11) y la NSR-10 establecen que una sección puede estar controlada por compresión o tracción:

Sección controlada por compresión: se da cuando en el momento en el que el concreto alcanza su deformación unitaria última (igual a 0.003), la deformación unitaria en el acero a tensión es igual o menor a un límite de deformación unitaria que se considera igual a $\frac{f_y}{E}$ (deformación de fluencia del acero).

Sección controlada por tracción: ocurre cuando la deformación unitaria en el acero a tensión es igual o mayor a 0.005 justo cuando el concreto alcanza su deformación de falla igual 0.003 en la fibra extrema a compresión.

El segundo caso es preferible, ya que antes de que el elemento colapse se deben desarrollar deformaciones en el mismo, dando oportunidad a los ocupantes de un edificio de desalojar en caso de sismo. (Rochel, 2012).

2.6 Diseño a flexión de vigas de concreto armado

Con el fin de obtener una expresión que permita calcular el momento resistente de una viga, se plantea el esquema de la Figura 3 adaptado de (Nilson & Winter, 1994):

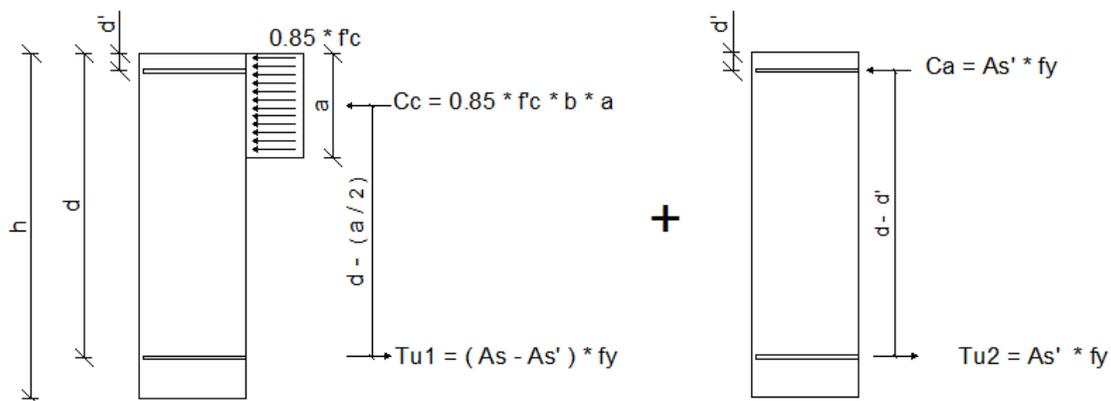


Figura 3. Distribución de fuerzas en la viga en el momento de la falla a flexión

Donde h es la altura de concreto en la viga, d la altura medida desde la fibra extrema a compresión y el centroide de las barras de acero en tracción, a altura del bloque de compresión equivalente, b base de la viga, As área de acero a tensión en la sección y As' área de acero a compresión en la sección.

En el diseño a flexión es posible considerar el esfuerzo que se produce en el acero a compresión, de forma tal que el momento resistente queda formado por el momento que se produce entre las fuerzas $Tu1$ y Cc , con un brazo igual a $d - a/2$. De forma tal que el momento resistente 1 es igual a:

$$Mn1 = (As - As') * fy * \left(d - \frac{a}{2} \right) \quad (6)$$

$(As - As')$ Representa la porción del acero en tensión que equilibra la compresión en el concreto. Por otro lado, el momento que se produce entre las fuerzas $Tu2$ y Ca es igual a:

$$Mn2 = As' * fy * (d - d') \quad (7)$$

Donde se considera que el acero a compresión As' está en fluencia en el momento de la falla.

Así, se obtiene que el momento resistente de la sección considerando ambos refuerzos (a tensión y compresión) es:

$$Mn = (As - As') * fy * \left(d - \frac{a}{2} \right) + As' * fy * (d - d') \quad (8)$$

Este momento se afecta por el factor de reducción de resistencia estipulado por la NSR 2010 que es igual a 0.9 para flexión.

2.7 Diseño de elementos a flexo-compresión

Una columna es el ejemplo más común de un elemento sometido a flexo-compresión. Así, una columna se ve sometida a una carga de compresión P_u y un momento M_u y esta condición de carga se suele sustituir por un estado equivalente que es igual a la carga P_u actuando con una excentricidad e como se presenta en la Figura 4.

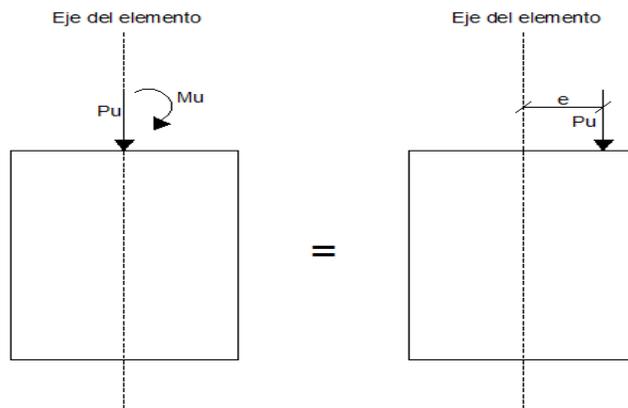


Figura 4. Condición de fuerza equivalente al momento.

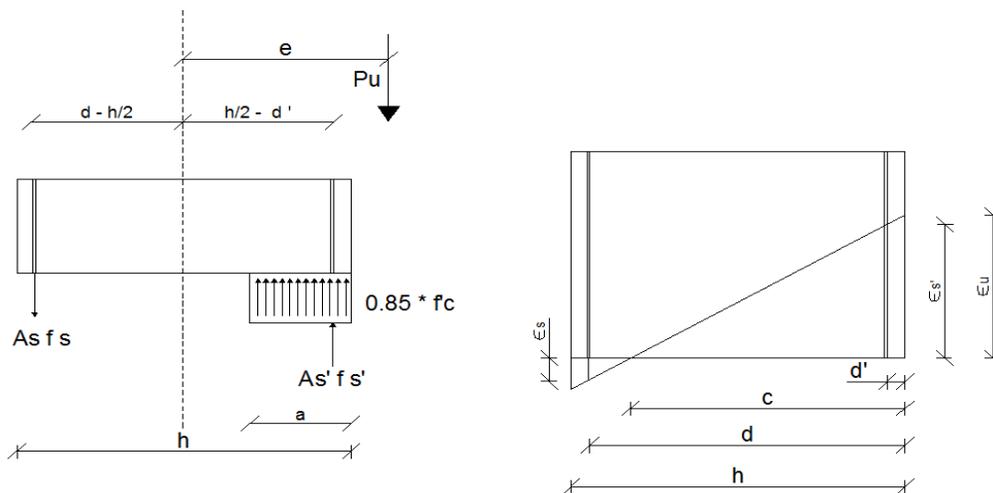


Figura 5. Distribución de esfuerzos y deformaciones en el momento de la falla de la columna.

La Figura 5, muestra la distribución de esfuerzos y deformaciones en una sección sometida a flexo-compresión con igual cantidad de refuerzo en ambas caras. Por sumatoria de fuerzas verticales se obtiene que la carga axial P_n en el momento de la falla es igual a:

$$P_n = 0.85 * f'_c * a * b + A_s' * f_s' - A_s * f_s \quad (9)$$

Donde f_s' y f_s son los esfuerzos en los aceros del lado a compresión en tensión respectivamente. Los valores de estos esfuerzos se calculan con base en la compatibilidad de deformaciones (Figura 5). En función de la ubicación del eje neutro (c) el esfuerzo en el acero de compresión pueden estar en fluencia o por debajo de este (Segura, 2011).

Igualmente, haciendo sumatoria de momentos con respecto al eje del elemento se obtiene que el momento resistente en la falla es:

$$M_n = P_n * e = 0.85 * f'_c * a * b * \left(\frac{h}{2} - \frac{a}{2}\right) + A_s' * f_s' * \left(\frac{h}{2} - d'\right) + A_s * f_s * \left(d - \frac{h}{2}\right) \quad (10)$$

Estas ecuaciones aplican para el caso en que se tienen solo dos filas de barras de refuerzo en las fibras extremas de la sección como se muestra en la Figura 5. En este trabajo se consideró que el armado de las columnas es simétrico y en dos capas por lo tanto se siguieron las ecuaciones (9) y (10).

3. Metodología de optimización

3.1 Planteamiento de la Metodología de Optimización

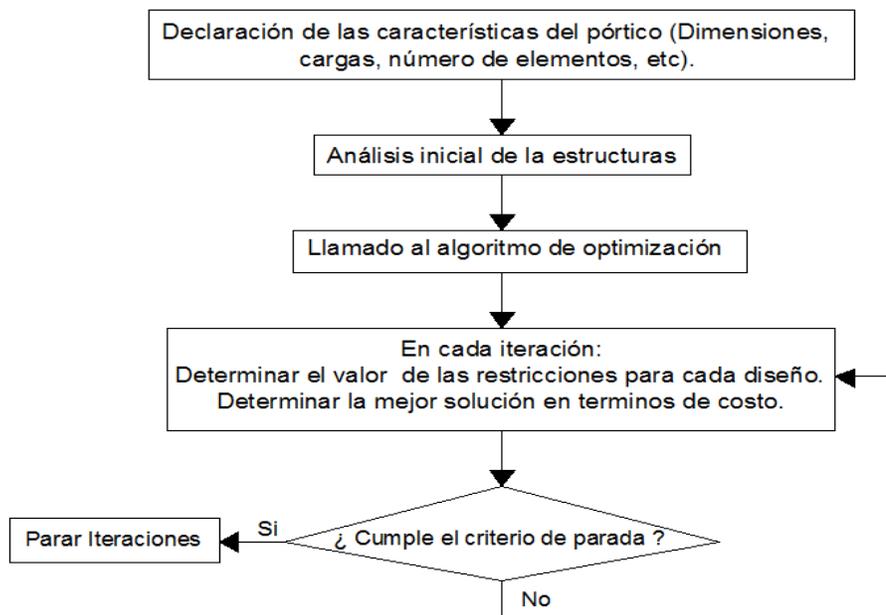


Figura 6. Esquema general de la metodología de optimización.

La Figura 6 muestra un esquema general de la metodología de optimización utilizada. Primero el usuario debe introducir las propiedades del pórtico como lo son las dimensiones, el número de elementos, tipos de apoyos, cargas actuantes, etc. Teniendo la geometría del pórtico, se generan unas dimensiones iniciales de los elementos de forma aleatoria con las cuales se lleva a cabo el análisis de la estructura. Luego, se utiliza el algoritmo (*PSO* ó *SQP*) para iniciar el proceso de optimización. En cada iteración el algoritmo determina el valor tanto de las restricciones como de la función objetivo para cada diseño, y escoge cual es la mejor solución (menor costo con cero violaciones de restricciones). Cuando se alcanza un criterio de parada, el algoritmo detiene la búsqueda. A continuación se describen las partes de la metodología de optimización utilizada en este trabajo.

3.1.1 Variables del problema

Como variables de optimización se utilizaron las dimensiones de concreto de los elementos del pórtico así como las áreas de acero de refuerzo. Para un elemento columna, el área de acero longitudinal se consideró como constante en toda su longitud y las secciones transversales podían ser rectangulares o cuadradas. El acero transversal se dividió en 3 partes como se muestra en la Figura 7 siendo la separación entre estribos la variable de diseño. S_s es la separación superior, S_i la inferior y S_c

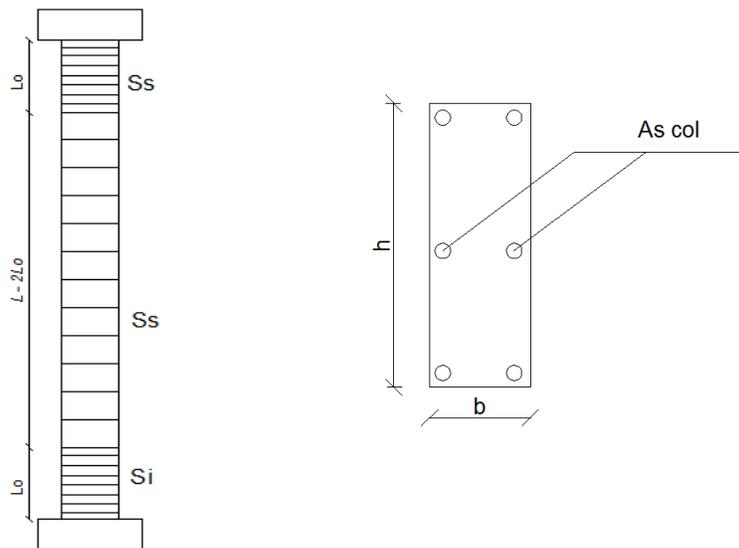


Figura 7. Variables de optimización en las columnas.

Por otro lado, en las vigas no es común tener un área de acero constante en toda la longitud del elemento. Es por ello que se dividió el refuerzo longitudinal en las vigas en seis partes. El refuerzo superior se dividió en las variables A_{is} (Acero izquierdo superior), A_{cs} (Acero central superior) y A_{ds} (Acero derecho superior), con desarrollos iguales a un tercio de la longitud de la viga. Por otro lado, en el refuerzo inferior se consideró un área A_{ci} (Acero central inferior) que se desarrolla un 80 % de la longitud, y en los nodos, las áreas A_{ii} (Acero izquierdo inferior) y A_{di} (Acero derecho inferior) que se desarrollan solo un 10 % de la longitud. La Figura 8 muestra la distribución de acero longitudinal usada en las vigas, que si bien es una simplificación (ya que

las vigas de un proyecto no suelen tener todas la misma distribución de aceros) permite considerar diferentes áreas de acero en las secciones de una viga.

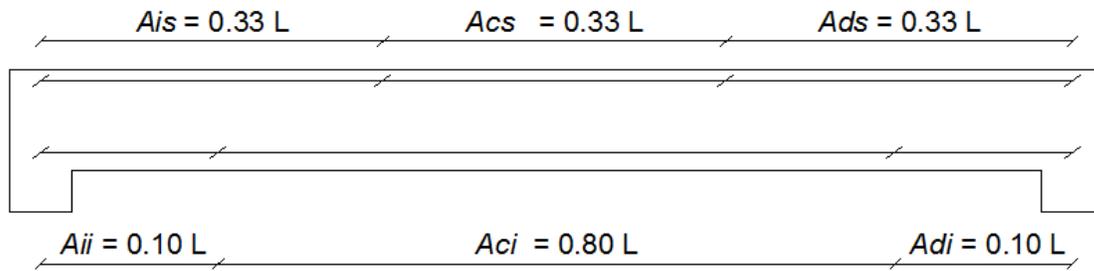


Figura 8. Distribución de acero longitudinal usado en las vigas.

También se consideró el refuerzo de cortante en las vigas. Se usaron tres separaciones: dos en los nodos (distribuidas en una distancia $2h$ desde el apoyo) y una para el tramo central en una longitud igual a $(l - 4 * h)$. De esta forma, al incluir las dimensiones de concreto b y h se obtuvieron 11 variables por cada viga. Con este número de variables se trató de obtener un diseño lo más real posible en comparación con otras metodologías de optimización donde solo se utilizan 3 variables por viga (Kaveh & Sabzi, 2011), (Gharehbaghi & Fadaee, 2012). También, el tener esta cantidad de variables permite aumentar la dificultad del problema, poniéndose a prueba las capacidades de los algoritmos de optimización.

En la optimización también se usaron grupos de elementos. Así, por ejemplo, las vigas de un mismo piso fueron asignadas al mismo grupo, con el fin de que quedaran con el mismo diseño. Lo anterior es también una simplificación que se ha aplicado en las metodologías de optimización antes mencionadas.

Fue necesario para el proceso de optimización, definir rangos para los valores que las variables podían tomar. Estos se muestran en la Tabla 1.

Tabla 1. Rangos de valores para las variables.

Var	min	Max	unidad
b viga	200	600	mm
h viga	500	1200	mm
As viga	254	15000	mm ²
b col	250	700	mm
h col	250	700	mm
As col	3600	19000	mm ²
s est.	75	300	mm

Las variables en la optimización se usaron con valores continuos, a diferencia de Payá et al., (2006), Kaveh & Sabzi, (2011) y Gharehbaghi & Fadaee, (2012) donde se usaron valores discretos. Lo anterior se debe a que para usar el *SQP* se requiere que las variables sean continuas.

3.1.2 Función Objetivo

Como función a optimizar se tomó el costo total del pórtico. A la vez, este se divide en el costo de las vigas y de las columnas que lo conforman. Las ecuaciones de la (11) a la (13), muestran el planteamiento matemático de la función costo usada en la optimización:

$$Costo = Costo_{vigas} + Costo_{col} \quad (11)$$

$$Costo_{vigas} = \sum_{i=1}^{Nv} Cc * b_i * h_i * L_i + Cs * \gamma_s * As_i * L_i + Cf * (b_i + 2 * h_i) * L_i + Cs * \gamma_s * l_{est} * N_{est} * A_{est} \quad (12)$$

$$Costo_{col} = \sum_{i=1}^{Nc} Cc * b_i * h_i * L_i + Cs * \gamma_s * As_i * L_i + Cf * (b_i + 2 * h_i) * L_i + Cs * \gamma_s * l_{est} * N_{est} * A_{est} \quad (13)$$

Donde:

Cc = Costo del concreto por volumen.

Cs = Costo del acero por peso.

Cf = Costo de la formaleta por área.

γ_s = Peso específico del acero de refuerzo.

L = Longitud del elemento, sea viga o columna.

Nc y Nv = Número de columnas y de vigas presentes en un pórtico.

b y h = Para vigas, representan la base y la altura, mientras que para columnas representan el ancho y la profundidad. (h es la dimensión de la columna paralela al plano del pórtico).

As = Área de acero longitudinal en el elemento.

l_{est} = longitud de un estribo cuadrado.

N_{est} = Número de estribos en el elemento.

A_{est} = área transversal de la barra usada para estribos.

3.1.2 Restricciones del problema

Las restricciones se tomaron de la norma NSR-10, para pórticos de concreto del tipo DMO y se clasificaron en dos grupos: las que aplican a vigas y las de columnas. A la vez, dentro de estos grupos se separaron en restricciones de forma y de resistencia. A continuación se plantean las restricciones tomadas y sus expresiones matemáticas.

Restricciones de forma para vigas:

El ancho mínimo de las vigas debe ser igual o mayor a 200 mm:

$$g_1(x) = 200 - b_{viga} \leq 0 \quad (14)$$

Para limitar la deflexión en las vigas, la altura h debe ser igual o mayor a la longitud de la viga entre 19:

$$g_2(x) = \frac{l}{19} - h_{viga} \leq 0 \quad (15)$$

La resistencia a momento positivo en la cara del nudo no debe ser menor a un tercio de la resistencia a momento negativo en ese mismo nudo. Esta restricción se puede asociar a las áreas de refuerzos colocadas. De esta forma se puede escribir:

$$g_3(x) = \frac{1}{3}(As_{nudo}^-) - As_{nudo}^+ \leq 0 \quad (16)$$

La resistencia a flexión sea a momento positivo o negativo en cualquier sección de la viga no debe ser menor a un quinto de la máxima resistencia a momento en los nudos:

$$g_4(x) = \frac{1}{5}(\max(As_{nudo}^\pm)) - As_{tramo}^\pm \leq 0 \quad (17)$$

La cuantía de refuerzo en cualquier sección debe ser al menos igual o mayor al mayor valor obtenido de las expresiones $\frac{1.4}{fy}$ y $\frac{0.25*\sqrt{f'c}}{fy}$. De forma matemática se puede plantear como:

$$g_5(x) = \max \left\{ \begin{array}{l} \frac{1.4}{fy} * b_{viga} * d_{viga} \\ \frac{0.25 * \sqrt{f'c}}{fy} * b_{viga} * d_{viga} \end{array} \right\} - As_{tramo}^\pm \leq 0 \quad (18)$$

La cuantía de acero de refuerzo en cualquier sección debe ser menor al 0.75 de la cuantía balanceada que se calcula como:

$$\rho_{bal} = 0.85^2 * \frac{f'_c}{f_y} * \frac{0.003}{0.003 + \frac{f_y}{E_s}} \quad (19)$$

La restricción matemática queda como:

$$g_6(x) = A_{s_{tramo}}^+ - (0.75 * \rho_{bal} * b_{viga} * d_{viga}) \leq 0 \quad (20)$$

La separación de estribos en los nodos de las vigas debe ser menor a un cuarto de la altura efectiva de la viga, 24 veces el diámetro del estribo, 8 veces el diámetro de la barra longitudinal o 300 mm. En este caso se consideró que el diámetro de la barra longitudinal es de 5/8". Así, la expresión matemática queda:

$$g_7(x) = s_{nodo} - \min \left\{ \frac{d_{viga}}{4} ; 24 * d_e ; 8 * d_l ; 300 \text{ mm} \right\} \leq 0 \quad (21)$$

El centro de la viga la separación de estribos debe ser menor a la mitad de la altura efectiva de la viga. Entonces:

$$g_8(x) = s_{central} - d/2 \leq 0 \quad (22)$$

Restricción de resistencia para vigas:

$$g_9(x) = Mu - \phi Mn \leq 0 \quad (23)$$

Donde Mn se calcula como se mostró en la ecuación (8). El factor de reducción de resistencia ϕ se toma igual a 0.9.

Restricciones de forma para columnas:

La dimensión mínima de la columna debe ser mayor o igual a 250 mm:

$$g_{10}(x) = 250 - \begin{Bmatrix} b_{col} \\ h_{col} \end{Bmatrix} \leq 0 \quad (24)$$

El área de refuerzo no debe ser menor del 1% ni mayor de 4% del área gruesa:

$$g_{11}(x) = (0.01 * b_{col} * h_{col}) - A_{s_{col}} \leq 0 \quad (25)$$

$$g_{12}(x) = A_{s_{col}} - (0.04 * b_{col} * h_{col}) \leq 0 \quad (26)$$

Restricción de resistencia para columnas:

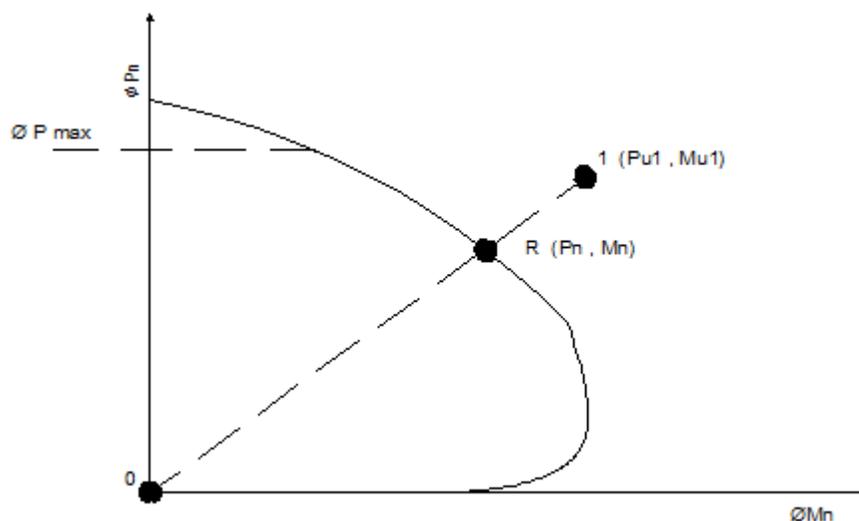


Figura 9. Diagrama de Interacción Axial-Momento (P-M) para columnas.

Para evaluar la resistencia de las columnas, el algoritmo genera un diagrama de interacción axial-momento como el de la Figura 9. Por cada columna, se determina cual es la combinación de carga que tiene mayor efecto sobre el elemento y se ubica dentro del diagrama de interacción representado en la Figura 9 por el punto 1. Hecho esto, se calcula la distancia entre el punto 0 y 1:

$$\overline{01} = \max(\sqrt{Pu_i^2 + Mu_i^2}) \quad (27)$$

Donde Pu_i y Mu_i , son la carga axial y el momento producido por la combinación más desfavorable. Una vez determinado el máximo efecto en la columna, se debe establecer la resistencia para compararlo con este valor. Para ello, el algoritmo traza una recta desde el origen (punto 0) hasta el diagrama P-M, con la misma pendiente de la recta $\overline{01}$ para obtener el punto R en la Figura 9. Se determina la distancia $\overline{0R}$ y se compara con la $\overline{01}$. Si la primera es mayor que la segunda la columna cumple por resistencia en caso contrario no. Por lo tanto la restricción se puede escribir como:

$$g_{12}(x) = \overline{01} - \overline{0R} \leq 0 \quad (28)$$

También, las derivas de pisos se restringieron al 1 % de la altura del piso.

Las combinaciones de carga utilizadas fueron:

$$1.4 D \quad (29)$$

$$1.2 D + 1.6 L \quad (30)$$

$$1.2 D + L \pm E \quad (31)$$

$$0.9 D \pm E \quad (32)$$

Donde D es la carga muerta, L la viva y E la sísmica.

3.2 Algoritmo Particle Swarm Optimization (PSO)

En la literatura se encuentran muchas variaciones del algoritmo original de *PSO* (Ratnaweera & Halgamuge, 2004), por lo tanto se realizó una revisión bibliográfica con el fin de escoger un algoritmo de *PSO* auto-adaptativo. Se compararon 4 versiones de *PSO* encontradas, (Zhan et al., 2009), (Ratnaweera & Halgamuge, 2004), (Guo & Chen, 2009) y (Parsopoulos & Vrahatis, 2007). De estas, se escogió la versión presentada por Zhan et al., en 2009. En este algoritmo todos los parámetros del *PSO* ($c1$, $c2$ y w) se adaptan durante el proceso de optimización, es decir, ninguno es escogido de forma subjetiva por el usuario. Lo anterior, evita tener hacer corridas del algoritmo para calibrar parámetros. Además, el algoritmo presenta buenos resultados al ser utilizado en problemas de prueba.

El algoritmo escogido, se basa en la determinación de un factor f en cada iteración, para estimar su estado evolutivo y calcular los parámetros $c1$, $c2$ y w . El factor f depende de las distancias medias entre las partículas y se calcula como:

$$f = \frac{dg - dmin}{dmax - dmin} \quad (33)$$

Siendo, dg la distancia media de la partícula con la mejor posición en el enjambre, $dmax$ y $dmin$ la distancia máxima y mínima respectivamente.

La distancia media de cada partícula se calcula como:

$$di = \frac{1}{N-1} \sum_{j=1, j \neq i}^N \sqrt{\sum_{k=1}^D (x_i^k - x_j^k)^2} \quad (34)$$

Dónde:

N = Número de partículas.

D = Número de variables en el problema.

x_i^k = Valor de la variable k en la partícula i .

x_j^k = Valor de la variable k en la partícula j .

La evolución del algoritmo pasa por 4 estados, estos son, exploración, explotación, convergencia y “salto” o (*jumping out*).

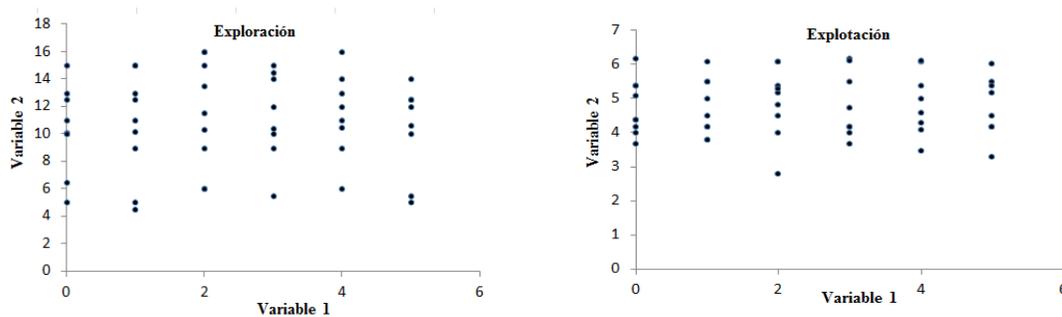


Figura 10. Estados de explotación y exploración del PSO. (Adaptado de (Zhan et al., 2009))

El significado del estado evolutivo se representa en las Figuras 10 y 11 para un espacio de búsqueda de 2 variables. En el estado de exploración las partículas se encuentran dispersas en el espacio de búsqueda, lo que hace que los valores ($dg - dmin$) y ($dmax - dmin$) en la ecuación (33) sean parecidos y el factor evolutivo sea cercano a 1. En la fase de explotación las partículas del enjambre se van acercando entre ellas, haciendo que el factor evolutivo disminuya a valores entre 0.3 y 0.4.

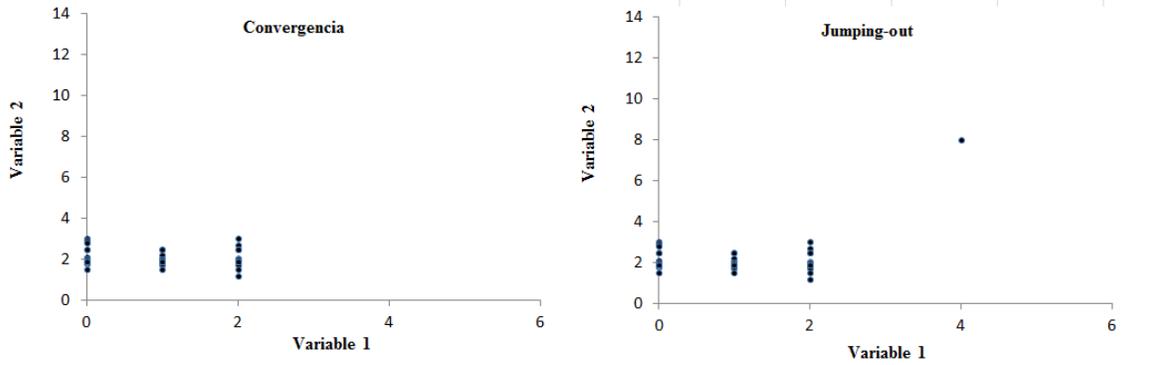


Figura 11. Estados de convergencia y jumping-out del PSO. (Adaptado de (Zhan et al., 2009))

En el estado de convergencia, las partículas se van agrupando alrededor de la mejor solución, que para las primeras iteraciones suele ser un mínimo local. Así, el factor evolutivo pasa a tener su valor más bajo de 0.1. Cuando alguna partícula cambia su posición y consigue un valor de la función objetivo mejor al que ha encontrado el enjambre, esta pasa a ser la mejor partícula, y se dice que el algoritmo entra en un estado *jumping-out*, obteniéndose nuevamente un incremento en el valor del factor evolutivo. El factor evolutivo se comporta como se muestra en la Figura 12. Cada vez que el valor del factor evolutivo incrementa, indica que el algoritmo entró en el estado de *jumping-out*.

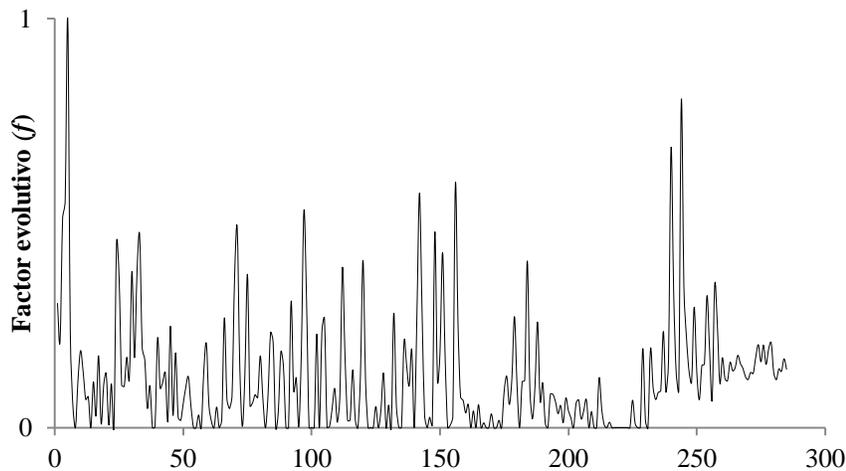


Figura 12. Comportamiento del factor evolutivo en una corrida de PSO.

Así, una vez calculado el factor evolutivo, el algoritmo determina el valor del peso inercial w , el cual se determina a partir de la ecuación (35):

$$w(f) = \frac{1}{1 + 1.5 e^{-2.6f}} \quad (35)$$

Para determinar los valores de $c1$ y $c2$ el algoritmo usa las siguientes estrategias en función del estado evolutivo:

Tabla 2. Cambio en los valores de $c1$ y $c2$ según el estado evolutivo. (+ : Aumentar ; - : Disminuir ; /+: Aumentar levemente ; /- : Disminuir levemente).

Estado	$c1$	$c2$
Exploración (S1)	+	-
Explotación (S2)	/+	/-
Convergencia (S3)	/+	/+
Jumping-out (S4)	-	+

En cada iteración, se determinan cuatro valores S en función de f , asociados a un estado evolutivo. Así, $S1$ es exploración, $S2$ es explotación, $S3$ es convergencia y $S4$ es *jumping out*. El estado en que se encuentra el algoritmo será aquel con mayor valor S . Para determinar $S1$, $S2$, $S3$ y $S4$, se utilizan las siguientes ecuaciones:

$$S1 = \begin{cases} 0, & 0 \leq f \leq 0.4 \\ 5 * f - 2, & 0.4 < f \leq 0.6 \\ 1, & 0.6 < f \leq 0.7 \\ -10 * f + 8, & 0.7 < f \leq 0.8 \\ 0, & 0.8 < f \leq 1 \end{cases} \quad (36)$$

$$S2 = \begin{cases} 0, & 0 \leq f \leq 0.2 \\ 10 * f - 2, & 0.2 < f \leq 0.3 \\ 1, & 0.3 < f \leq 0.4 \\ -5 * f + 3, & 0.4 < f \leq 0.6 \\ 0, & 0.6 < f \leq 1 \end{cases} \quad (37)$$

$$S3 = \begin{cases} 1, & 0 \leq f \leq 0.2 \\ -5 * f + 1.5, & 0.1 < f \leq 0.3 \\ 0, & 0.3 < f \leq 1 \end{cases} \quad (38)$$

$$S4 = \begin{cases} 0, & 0 \leq f \leq 0.7 \\ -5 * f + 1.5, & 0.7 < f \leq 0.9 \\ 1, & 0.9 < f \leq 1 \end{cases} \quad (39)$$

Además de determinar el estado evolutivo mediante las ecuaciones anteriores, el algoritmo respeta la secuencia de estados, la cual va en el siguiente orden:

$$S1 \rightarrow S2 \rightarrow S3 \rightarrow S4$$

Así, por ejemplo, si en la iteración i el estado calculado con las ecuaciones es $S2$ pero en la iteración $i-1$ el estado fue $S3$, entonces se establece que el estado evolutivo queda en $S3$ para no perder la secuencia del algoritmo. Ahora bien, si en la iteración i el estado evolutivo es $S1$ y en la iteración $i+1$ se determinó a través de las ecuaciones que el estado es $S2$, entonces sí se pasa a este nuevo estado porque se está respetando la secuencia. De igual manera, pasar de estado $S4$ a $S1$ se considera también que respeta la secuencia del algoritmo, ya que es equivalente al reinicio del ciclo. A continuación se presenta un esquema de los pasos realizados por el algoritmo *PSO* en cada iteración:

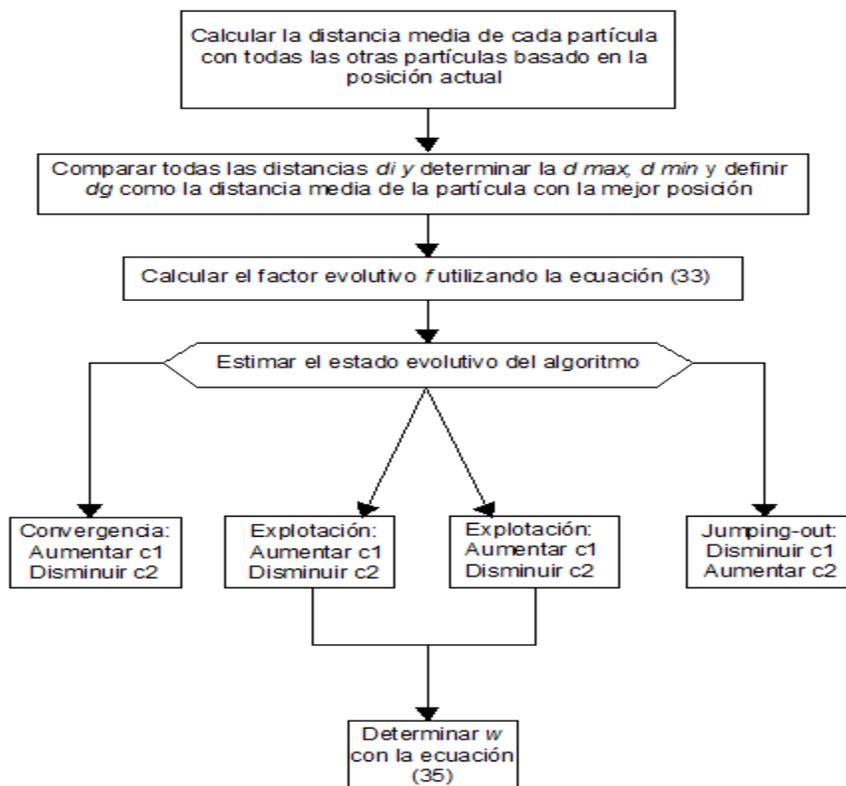


Figura 13. Esquema para determinar los parámetros del algoritmo PSO en cada iteración. (Adaptado de (Zhan et al., 2009))

Para que el algoritmo *PSO* funcionara en la optimización restringida, fue necesario plantear una función de penalización, la cual está constituida por la función objetivo multiplicada por un factor penalizador. La función de penalización usada fue:

$$Fp = (1 + G)^2 * F \quad (40)$$

Donde $(1 + G)^2$ es el factor penalizador y F la función objetivo de la ecuación (11). El factor penalizador G , es la suma de las restricciones violadas por un diseño, y por lo tanto siempre tendrá un valor positivo (recordar que una restricción es violada cuando su valor es mayor que cero). De esta forma una solución puede ser muy económica, pero si viola restricciones, la función Fp aumentará, haciéndola poco atractiva para el algoritmo.

A pesar de que el algoritmo *PSO* es auto-adaptivo siempre el usuario tendrá que definir tanto el número de partículas como el criterio de parada. El número de partículas depende de la dificultad del problema, lo cual es proporcional al número de variables tomadas en cuenta. La definición del criterio de parada usado en este trabajo se muestra en la siguiente sección.

Otro aspecto que hubo que definir fue si en la primera iteración, las partículas del enjambre debían estar dentro del espacio factible como en Kaveh & Sabzi, (2011) y Gharehbaghi & Fadaee, (2012) o podían iniciar en cualquier punto del espacio de búsqueda sin importar si violaban o no las restricciones del problema.

3.3 Algoritmo Sequential Quadratic Programming (SQP)

El algoritmo de programación cuadrática secuencial utilizado es el planteado por Arora, en 2012. Para ejecutar este algoritmo, se utilizó la herramienta de optimización del software Matlab™. Los valores de las variables en la primera iteración se generaron aleatoriamente dentro de los rangos indicados en la Tabla 1. En la definición del problema de programación cuadrática se utilizó la estrategia de la Restricción Potencial (*Potential Constraint Strategy*), la cual consiste en tomar solo las restricciones activas para definir el sub-problema cuadrático planteado en la ecuación (3), disminuyendo el tiempo consumido por el algoritmo para definir el problema cuadrático.

Como función descendente se utilizó la siguiente función (Flores & Rodríguez, 2010):

$$|\varphi(x) = f(x) + \sum_{i=1}^m r_i * g_i(x) \quad (41)$$

Donde $\varphi(x)$ representa el valor de la función descendente, $f(x)$ la función objetivo utilizada, m el número de restricciones, $g_i(x)$ el valor de la restricción i y r_i es un parámetro que se calcula como:

$$r_{i_{k+1}} = \max \left\{ \lambda_i, \frac{r_{i_k} + \lambda_i}{2} \right\} \quad (42)$$

λ_i Representa el multiplicador de LaGrange asociado a la restricción i .

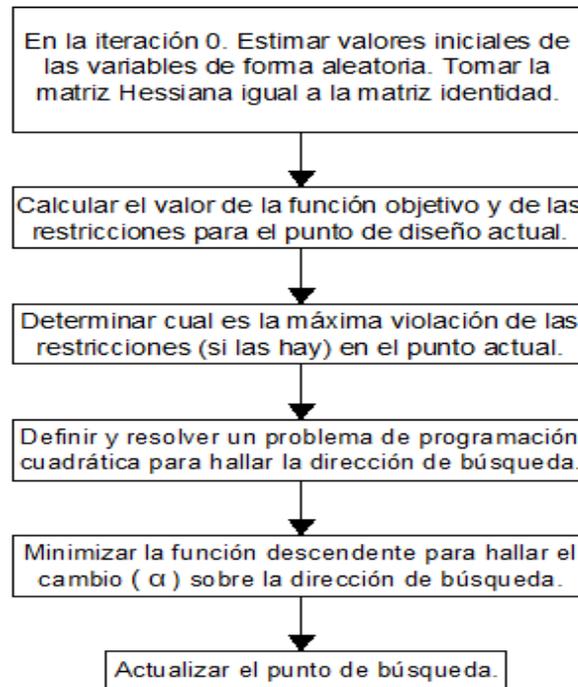


Figura 14. Esquema de pasos ejecutados por el algoritmo SQP.
(Arora, 2012)

La Figura 14 muestra un esquema de los pasos llevados a cabo por el algoritmo *SQP* en cada iteración.

3.4 Análisis de las estructuras

Las fuerzas actuantes en cada elemento se determinaron mediante un análisis lineal-elástico. Para ello, se elaboró un programa de análisis matricial

utilizando Matlab™. Dependiendo de la distribución de rigideces en el pórtico, las fuerzas actuantes en cada elemento cambian. En las metodologías de optimización encontradas en la literatura (Payá et al., (2006), Kaveh & Sabzi, (2011) Balling & Yao, (1997), entre otros) en cada iteración del algoritmo de optimización se realiza un análisis del diseño actual, lo cual genera una alta demanda computacional, sobre todo para el *PSO*, donde en cada iteración sería necesario analizar tantas estructuras como partículas en el enjambre hayan. Por ello, en la metodología utilizada en este trabajo, se realiza un análisis del pórtico con unas secciones iniciales en la primera iteración. Las fuerzas máximas actuantes determinadas se mantienen hasta la última iteración donde se hace un re-análisis de la estructura con el fin de verificar que el diseño final cumpla con las fuerzas actuantes asociadas a la distribución de rigideces obtenida. Hay que aclarar que hacer un solo análisis inicial es una simplificación ya que se está despreciando el cambio de rigideces en el pórtico que ocurre en cada iteración con el cambio de las secciones transversales. Sin embargo, su aplicación fue necesaria para la programación del algoritmo *SQP*.

Todas las corridas de los programas se realizaron usando un computador con un procesador i7 de 2.2 GHz y 8 GB de memoria RAM. Los algoritmos se programaron usando el Software Matlab™.

4. Ejemplos Numéricos

4.1 Ejemplos Numéricos utilizados

A continuación se presentan los modelos de pórticos utilizados en este trabajo, los cuales fueron tomados de Kaveh & Sabzi, (2011). En los ejemplos, los costos unitarios para los materiales en dólares fueron del concreto 308 \$/m³, del acero 2.30 \$/kgf y del encofrado 264 \$/m². Estos se determinaron a partir de la base de datos de la revista Construdata, (2014) e incluyen los costos por mano de obra, equipos y la materia prima.

Además de los costos, las propiedades de los materiales también se dejaron como constantes. La calidad de acero usado fue $f_y = 420$ MPa tanto para acero longitudinal como para transversal, y del concreto fue $f'_c = 25$ MPa. El diámetro de los estribos fue de 3/8". Los valores de módulos de elasticidad fueron, para el acero 200000 MPa y para el concreto 23500 MPa.

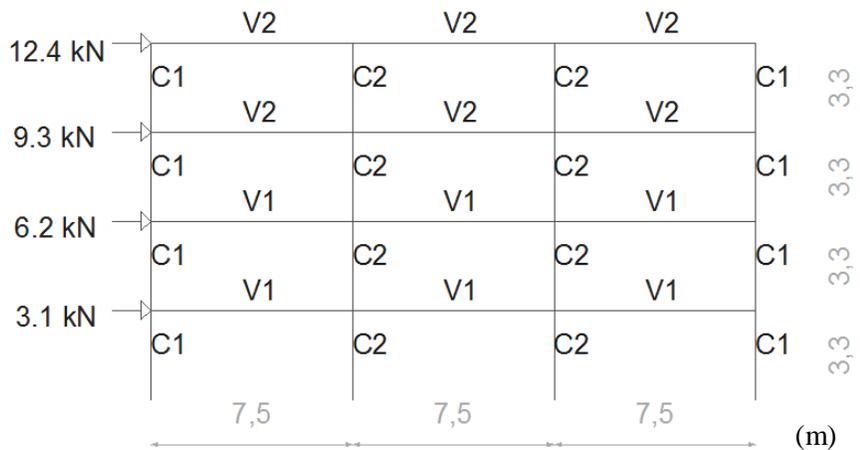
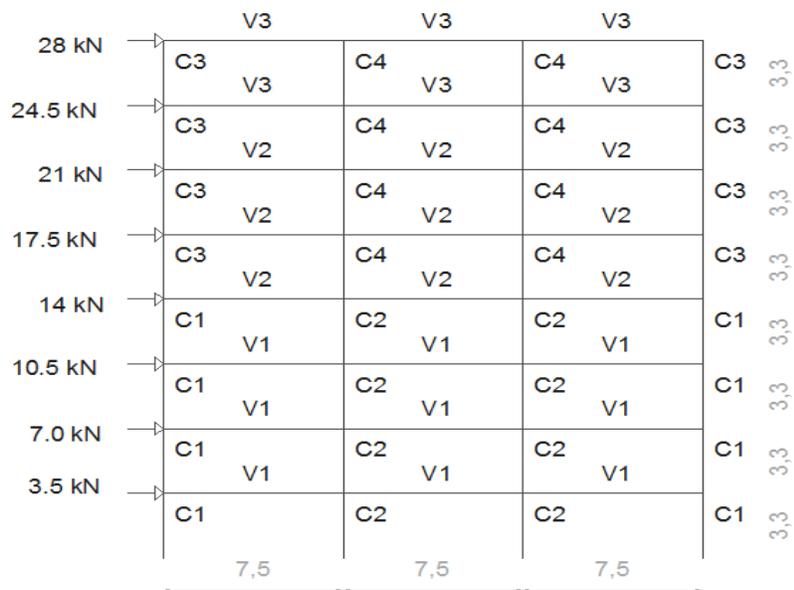
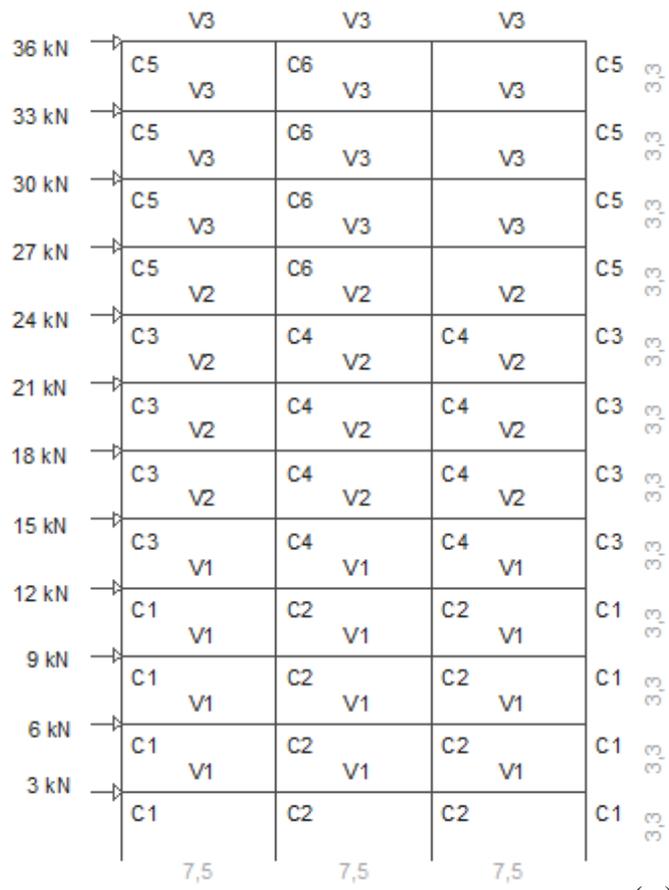


Figura 15. Pórtico 1: 4 pisos y 3 luces.



(m)

Figura 16. Pórtico 2: 8 pisos y 3 luces.



(m)

Figura 17. Pórtico 3: 12 pisos y 3 luces.

Las cargas laterales mostradas en las Figuras 15, 16 y 17, son parte de los ejemplos presentados por Kaveh & Sabzi, (2011) donde además se aplicó en todos los tramos de las vigas una carga uniformemente distribuida de 22.3 kN/m como muerta y de 10.7 kN/m como viva. Como se observa también en las Figuras 15, 16 y 17 los elementos están agrupados por tipos. Así por ejemplo, todas las columnas pertenecientes al grupo *CI* en el pórtico 3 tendrán el mismo diseño.

4.1 Efecto de la función objetivo penalizada en la búsqueda del *PSO*

Como se mencionó en la metodología, para utilizar el *PSO* en problemas de optimización restringida, es necesario introducir un factor de penalización dentro de la función objetivo. La ecuación (40) se re-escribe aquí por conveniencia:

$$Fp = (1 + G)^2 * F \quad (43)$$

Donde G es el factor de penalización, F es el costo del pórtico y Fp es el costo penalizado.

Las Figuras 18, 19 y 20, muestran la convergencia del algoritmo *PSO* para los tres pórticos utilizados en este trabajo. En las tres Figuras se observa que en las primeras iteraciones, el valor de la función costo disminuye rápidamente. Luego, el descenso se hace más lento. El rápido descenso inicial, se debe a que en las iteraciones iniciales las soluciones encontradas por el algoritmo violan las restricciones del problema, por lo tanto, el factor G de la ecuación (43) es alto. A medida que avanzan las iteraciones, el algoritmo ajusta las soluciones del problema, de forma tal que ya no haya violaciones de restricciones, haciendo que factor G disminuya. Hasta este punto la búsqueda se guía por el costo con penalización. Una vez que el algoritmo consigue soluciones en el espacio factible ($G = 0$), la búsqueda empieza a guiarse con la función costo sin penalización. En esta segunda fase, el descenso es menos marcado que en la primera, pero aun así, se mantiene la disminución, como se observa en las gráficas aumentadas.

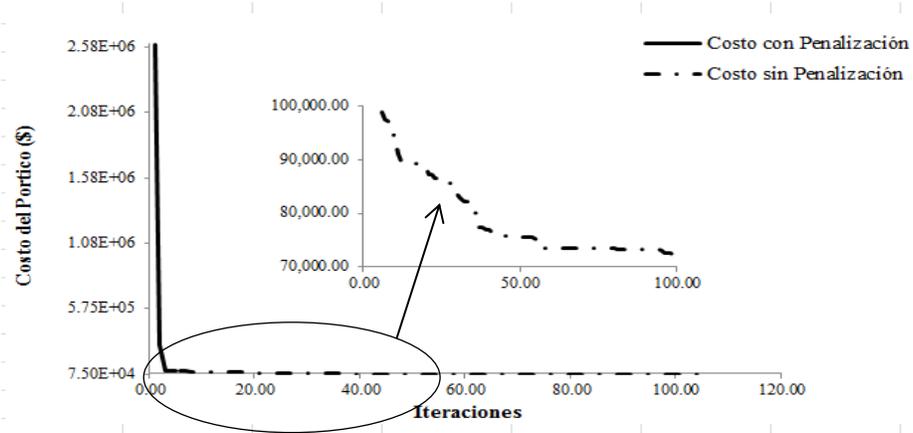


Figura 18. Convergencia del algoritmo de PSO para el pórtico 1.

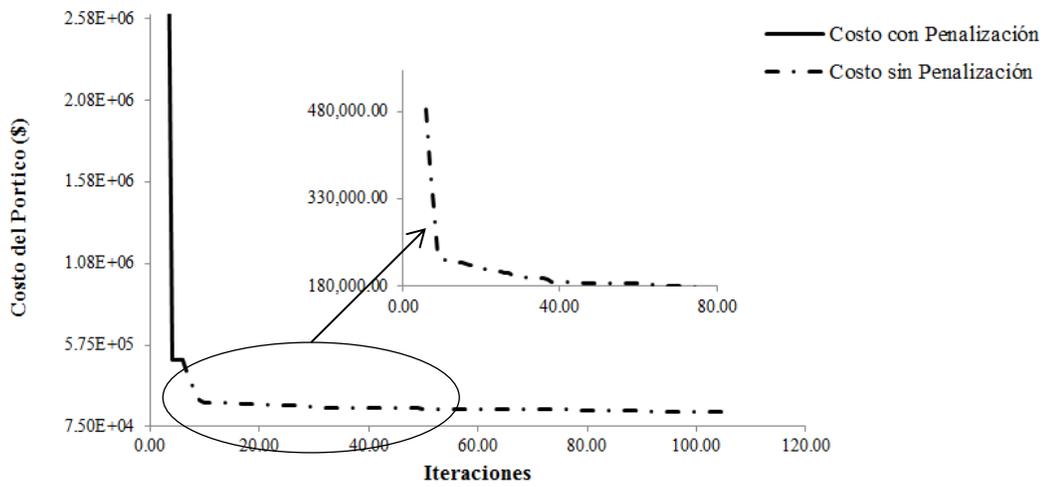


Figura 19. Convergencia del algoritmo de PSO para el pórtico 2.

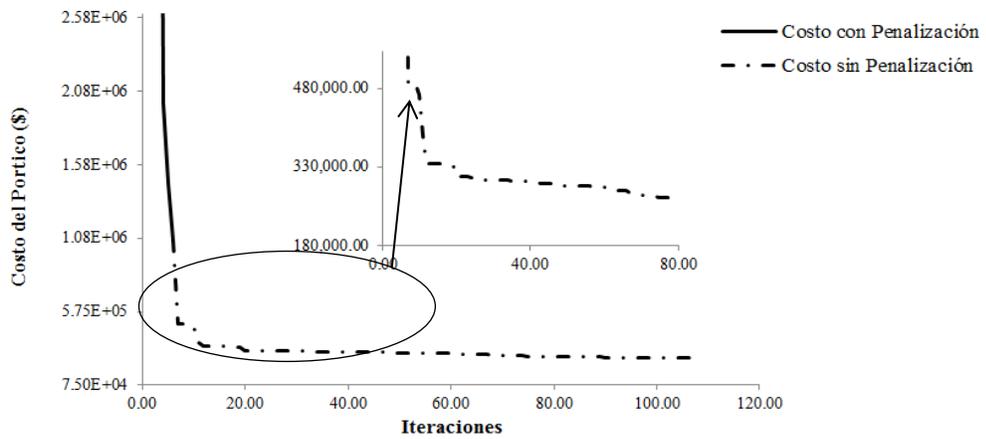


Figura 20. Convergencia del algoritmo de PSO para el pórtico 3.

Las Figuras 18, 19 y 20 demuestran que el factor de penalización incluido en la función de desempeño, cumple su objetivo de guiar al algoritmo *PSO* hacia el espacio de soluciones factibles para los problemas planteados.

4.2 Criterio de parada del *PSO*

En esta sección se muestra el efecto de variar el número de iteraciones usadas como criterio de parada para el algoritmo *PSO*. La Figura 21, 22 y 23 muestran los resultados para los 3 pórticos usados. Se usaron tres criterios de parada 20, 50 y 80 iteraciones.

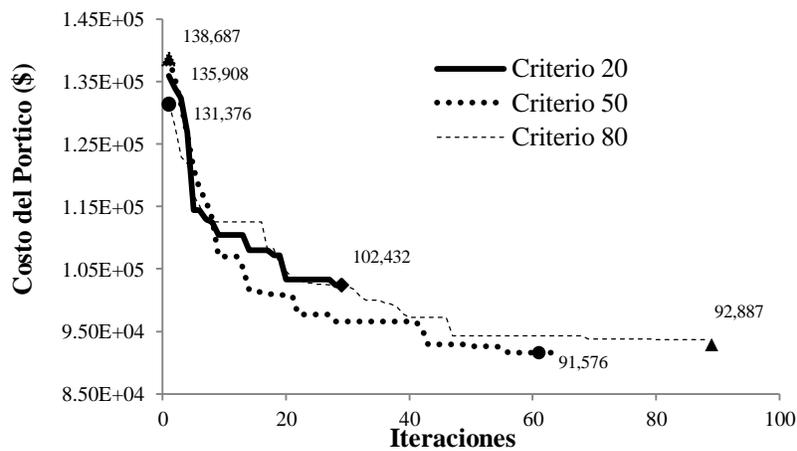


Figura 21. Convergencia del algoritmo de PSO para el pórtico 1 variando el criterio de parada

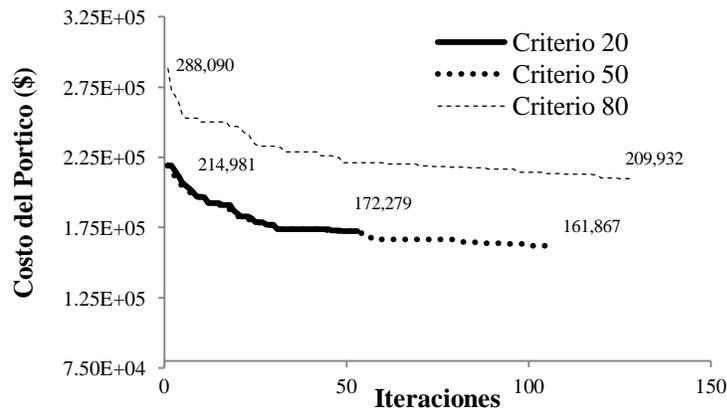


Figura 22. Convergencia del algoritmo de PSO para el pórtico 2 variando el criterio de parada.

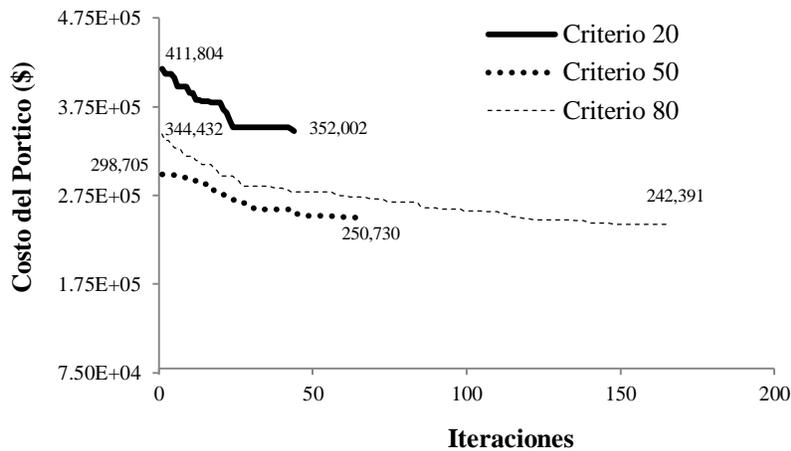


Figura 23. Convergencia del algoritmo de PSO para el p3rtico 3 variando el criterio de parada.

Dado que el criterio de parada est1 relacionado con que tanto puede reducir el algoritmo la funci3n objetivo, se consider3 la reducci3n entre el punto inicial y final como punto de comparaci3n. Usando 20 iteraciones se obtuvo una reducci3n de costo de 24% en el p3rtico 1, 19.8% en el 2 y 14.5% en el 3. Con 50 iteraciones, la reducci3n fue del 31% para el p3rtico 1, 24.7% para el 2 y 16% para el 3. Con 80 iteraciones, la reducci3n de costo fue del 28% para el p3rtico 1, 22.4 para el 2 y de 27% para el 3. Usando 50 iteraciones, se logr3 mejorar los resultados que se obtuvieron usando solo 20. Sin embargo, pasar de 50 a 80 iteraciones solo mejor3 el resultado en el tercer p3rtico. Por lo tanto, se escogi3 50 como el n3mero de iteraciones que ofrece mejor resultado.

En el trabajo de Kaveh & Sabzi, 2011, el n3mero de iteraciones usadas como criterio de parada fue de 20, para los mismos ejemplos que se utilizaron en esta investigaci3n, pero no se reporta como se lleg3 a este valor. Si bien no existe una forma directa para escoger un criterio de parada, en los resultados obtenidos se observa, que utilizar solo 20 iteraciones como criterio de parada hace que el algoritmo detenga la b3squeda prematuramente, haciendo que la exploraci3n del espacio de soluciones quede limitada. Es probable que de haberle dado m1s iteraciones a los algoritmos de Kaveh & Sabzi, (2011), hubiesen obtenido mejores resultados.

De las Figuras 21, 22 y 23, tambi3n se puede decir que el resultado final arrojado por el algoritmo no solo depende del criterio de parada, sino tambi3n de las posiciones iniciales de las part3culas. Es por ello que para un mismo problema, se recomienda realizar varias corridas del algoritmo, partiendo en cada una desde posiciones iniciales diferentes.

4.2 Comparación del desempeño del PSO al generar la primera población dentro y fuera del espacio factible

Se utilizó el pórtico de 4 pisos y 3 luces (Pórtico 1) que se muestra en la Figura 15 para probar las dos formas de crear la población inicial del algoritmo *PSO*. Se utilizaron 100 partículas en el enjambre para este problema de 28 variables.

Tabla 3. Resultados del PSO para el Pórtico 1 generando la población inicial desde puntos factibles

Costo Inicial (\$)	Costo Final (\$)	Reducción del costo (%)	Tiempo (s)	Iteraciones totales	1era Generación (s)	(%)
118,300.00	90,940.00	23.13	2894	50	996.5	34.43
121,900.00	78,650.00	35.48	3356	96	1007	30.01
124,100.00	83,420.00	32.78	3169	173	634	20.01

Tabla 4. Resultados del PSO para el Pórtico 1 generando la población inicial desde puntos no factibles

Costo Inicial (\$)	Costo Final (\$)	Reducción del costo (%)	Tiempo (s)	Iteraciones totales
135,900.00	102,400.00	24.65	786	30
138,700.00	91,580.00	33.97	1181	64
131,400.00	90,920.00	30.81	1778	91

Las Tablas 3 y 4, muestran los resultados del *PSO* para el pórtico 1, generando la población inicial en puntos factibles y no factibles respectivamente. Cada fila en las Tablas representa una corrida del algoritmo partiendo desde puntos iniciales diferentes. En la Tabla 3, se tiene el registro del tiempo requerido para generar la primera población aleatoriamente, y cuanto representa del tiempo total de corrida (columnas 6 y 7). Se observa que en las tres corridas realizadas, el tiempo promedio requerido para generar la primera población es superior al 25 % del tiempo total. Es decir, más de un cuarto del tiempo de trabajo del algoritmo es consumido en generar la primera población.

Comparando la Tabla 3 con la 4, se observa que el costo final obtenido por el algoritmo iniciando desde puntos factibles es menor al obtenido en la Tabla 4. Sin embargo, la reducción del costo es casi la misma. En cuanto a los tiempos de corrida el algoritmo que inicia desde poblaciones no factibles se tarda menos de la mitad del tiempo que necesita el otro. Esto indica que si bien generar la población inicial dentro del espacio factible da mejores soluciones, el tiempo requerido para lograrlo es una desventaja.

Esta misma prueba se realizó utilizando el pórtico de 8 pisos y 3 luces (pórtico 2), el cual se muestra en la Figura 15. Para este pórtico el número de partículas se fijó en 200.

Dado el número de variables involucradas en este ejemplo, el algoritmo no alcanzó a generar aleatoriamente una partícula en el rango factible después de 10,800 s de trabajo. Esto indica que generar una población dentro del espacio factible, puede dejar de ser práctico para problemas con muchas variables, debido al tiempo requerido para lograr colocar todas las partículas del enjambre en posiciones que cumplan con todas las restricciones.

A partir de estos resultados se recomienda generar la población inicial en puntos aleatorios, sin importar si cumplen o no con las restricciones. Además, como ya se vio en la sección 4.2, el factor de penalización en la función objetivo, cumple con su función de guiar la búsqueda del algoritmo, hacia el espacio factible, por lo que no es necesario ubicar las partículas en posiciones sin violaciones de restricciones desde el inicio.

4.3 Comparación de los resultados encontrados por el PSO y el SQP

En esta sección se comparan los resultados obtenidos por ambos métodos al ser aplicados a los tres ejemplos presentados. Si bien en las secciones anteriores se consideró la reducción de costo como criterio de comparación para calibrar los parámetros del *PSO*, en esta parte la comparación entre ambos algoritmo se hace en función del menor costo final encontrado. Para cada ejemplo y con cada algoritmo se realizaron 30 corridas de las cuales se presentan los resultados a continuación:

4.3.1 Pórtico 1:

Tabla 5. Resultados del SQP para el Pórtico 1 en 30 corridas.

Corrida	Costo Inicial	Costo Final	% Reducción	Iteraciones
1	100,700.00	63,560.00	36.88	162
2	98,580.00	63,560.00	35.52	159
3	100,700.00	63,560.00	36.88	154
4	99,550.00	63,560.00	36.15	145
5	93,180.00	63,560.00	31.79	154
6	105,900.00	63,560.00	39.98	140
7	99,980.00	63,560.00	36.43	163
8	99,510.00	63,560.00	36.13	156
9	94,050.00	63,560.00	32.42	137
10	101,000.00	63,560.00	37.07	157
11	108,600.00	63,560.00	41.47	148
12	107,200.00	63,560.00	40.71	147
13	86,010.00	63,560.00	26.10	62
14	240,700.00	63,560.00	73.59	189

15	330,700.00	63,560.00	80.78	158
16	608,000.00	63,560.00	89.55	129
17	257,100.00	63,560.00	75.28	132
18	241,900.00	63,560.00	73.72	139
19	332,200.00	63,560.00	80.87	129
20	556,900.00	63,560.00	88.59	140
21	527,100.00	63,560.00	87.94	146
22	266,100.00	63,560.00	76.11	113
23	423,800.00	66,670.00	84.27	108
24	417,900.00	63,560.00	84.79	144
25	195,000.00	63,560.00	67.41	111
26	535,500.00	63,560.00	88.13	133
27	185,800.00	63,560.00	65.79	114
28	289,200.00	63,560.00	78.02	133
29	219,200.00	63,560.00	71.00	129
30	170,500.00	63,560.00	62.72	135
promedio	236,418.67	63,663.67	59.87	138

En la tabla 5, se tienen los resultados arrojados por el *SQP*, al ser aplicado en la optimización del pórtico 1. Se observa que en las 30 corridas realizadas, el algoritmo logró encontrar un diseño que cumple con las restricciones, pero sin embargo convergió siempre al mismo valor de la función objetivo (63,560), a pesar de iniciar cada corrida desde puntos diferentes. Lo anterior indica que el algoritmo está quedando atrapado en un mínimo local por lo que no logra cambiar a una mejor solución al llegar a este punto.

Tabla 6. Resultados del PSO para el Pórtico 1 en 30 corridas.

Corrida	Costo Inicial	Costo Final	% Reducción	Iteraciones
1	109,670.00	64,900.00	40.82	164.00
2	95,580.00	62,690.00	34.41	1,001.00
3	100,700.00	73,790.00	26.72	150.00
4	93,900.00	66,780.00	28.88	491.00
5	81,070.00	68,440.00	15.58	326.00
6	105,900.00	64,740.00	38.87	143.00
7	101,200.00	63,000.00	37.75	145.00
8	102,100.00	66,540.00	34.83	327.00
9	96,600.00	67,060.00	30.58	495.00
10	101,000.00	67,000.00	33.66	144.00
11	112,200.00	68,120.00	39.29	1,001.00

12	110,900.00	69,860.00	37.01	146.00
13	104,400.00	68,750.00	34.15	271.00
14	95,640.00	64,400.00	32.66	617.00
15	95,710.00	65,200.00	31.88	215.00
16	102,400.00	62,140.00	39.32	676.00
17	98,840.00	68,260.00	30.94	137.00
18	97,350.00	64,090.00	34.17	135.00
19	102,600.00	65,570.00	36.09	267.00
20	102,700.00	67,360.00	34.41	122.00
21	98,150.00	73,630.00	24.98	169.00
22	92,530.00	67,100.00	27.48	134.00
23	84,470.00	66,830.00	20.88	125.00
24	93,310.00	72,700.00	22.09	114.00
25	99,840.00	71,210.00	28.68	199.00
26	108,100.00	70,520.00	34.76	149.00
27	97,180.00	69,770.00	28.21	210.00
28	102,500.00	63,860.00	37.70	290.00
29	98,840.00	69,470.00	29.71	213.00
30	105,800.00	68,180.00	35.56	152.00
promedio	100,438.00	66,751.33	33.14	375.73

La Tabla 6, muestra los resultados del *PSO* para el p3rtico 1. Se puede observar que en cada corrida, el *PSO* alcanz3 un valor de la funci3n objetivo diferente. Tambi3n, en las corridas 2, 7 y 16, el costo final del p3rtico arrojado por el *PSO* fue menor al encontrado por el *SQP*, mejor3ndolo hasta en un 2.2%.

Comparando los resultados de los algoritmos se observa que el *PSO* logr3 alcanzar un dise1o m3s econ3mico en comparaci3n al *SQP*, sin embargo la diferencia es de solo 2.2%, por lo tanto se puede decir que para este ejemplo el desempe1o de ambos algoritmos es similar. Tambi3n el hecho de que el *PSO* haya arrojado resultados tan diversos en las 30 corridas, muestra que el espacio de b3squeda es complejo, con muchos 3ptimos locales, y que el resultado dado por el *SQP* no es m3s que uno de estos 3ptimos y este algoritmo est3 quedando atrapado en un m3nimo local.

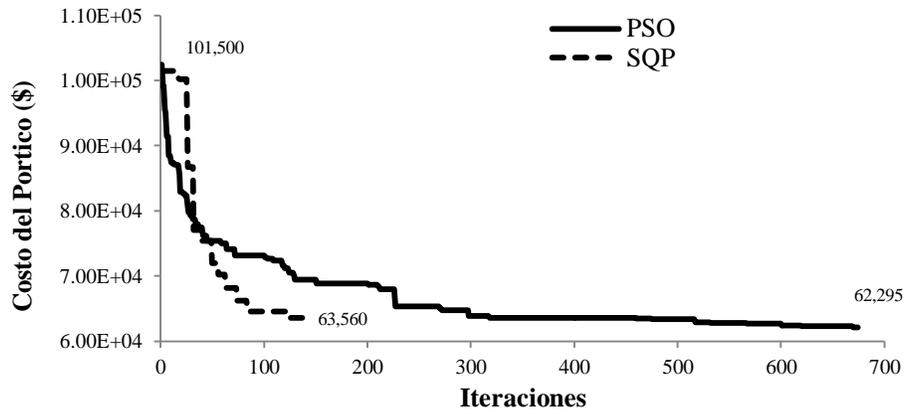


Figura 24. Comparación de convergencia de los algoritmos para el pórtico 1

La Figura 24 muestra la convergencia de una de las corridas del *PSO* y del *SQP* al ser aplicados al pórtico 1 con el fin de observar el comportamiento de los dos algoritmos con el paso de las iteraciones. Se puede observar la diferencia en el número de iteraciones realizadas por cada algoritmo, más de 650 por el *PSO* y solo 140 por el *SQP*. El hecho de que el *SQP* se detenga tan rápido en comparación con el *PSO* probablemente se deba a que el algoritmo queda atrapado en un mínimo local. Lo anterior quiere decir que al llegar a esta solución el algoritmo no consigue moverse a otra donde se disminuya la función objetivo y por lo tanto se detiene.

A continuación se comparan las relaciones de capacidad/demanda en los elementos, entre los diseños óptimos encontrados por cada algoritmo.

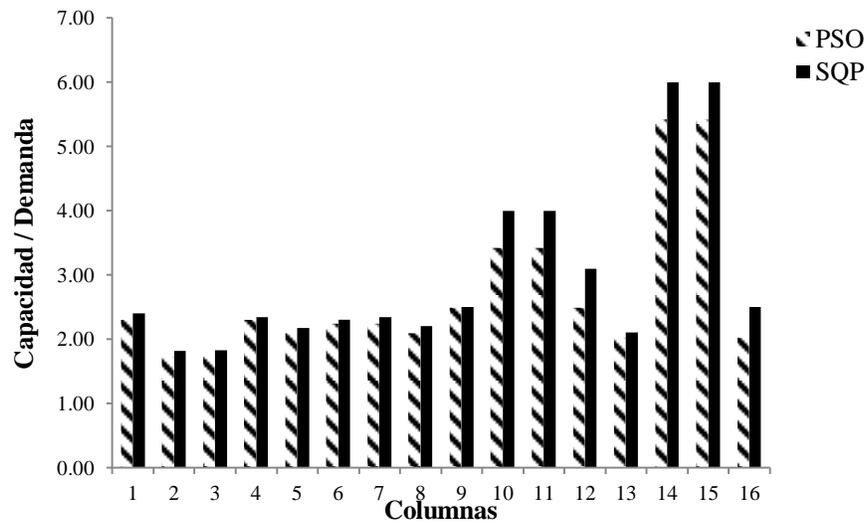


Figura 25. Relación capacidad/demanda en las columnas pórtico 1.

La Figura 25 muestra la relación capacidad/demanda en las 16 columnas del pórtico 1. Se observa que para todas las columnas la relación capacidad/demanda en el diseño del *SQP* es mayor que el obtenido en el diseño del *PSO*. En ambos diseños, todas las columnas quedaron con relaciones capacidad/demanda iguales o superiores a 2, menos en las columnas 2 y 3, que por ser las del primer piso y centrales están más cargadas.

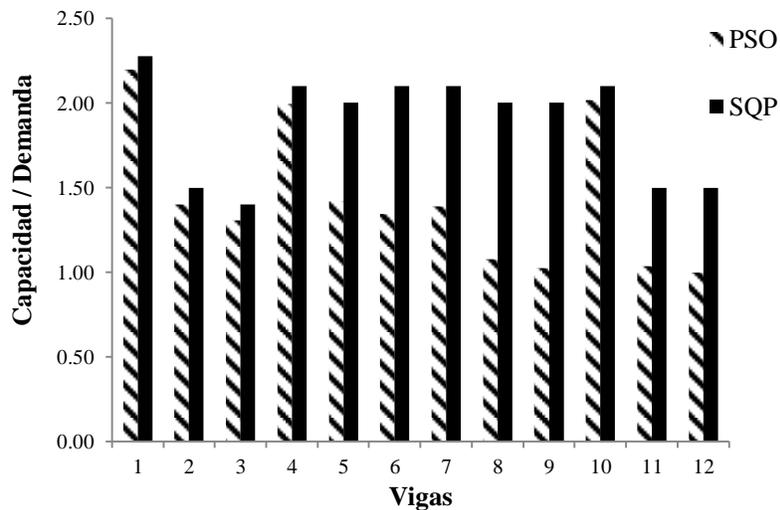


Figura 26. Relación capacidad/demanda para momento superior izquierdo en las vigas pórtico 1.

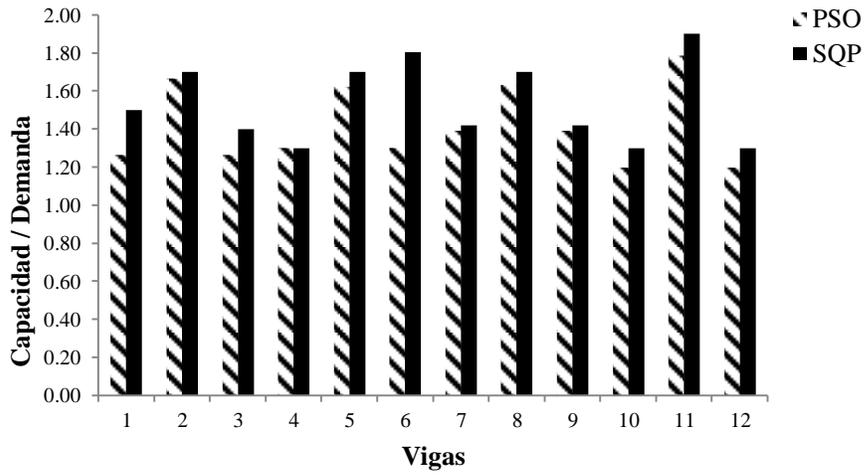


Figura 27. Relación capacidad/demanda para momento inferior central en las vigas pórtico 1.

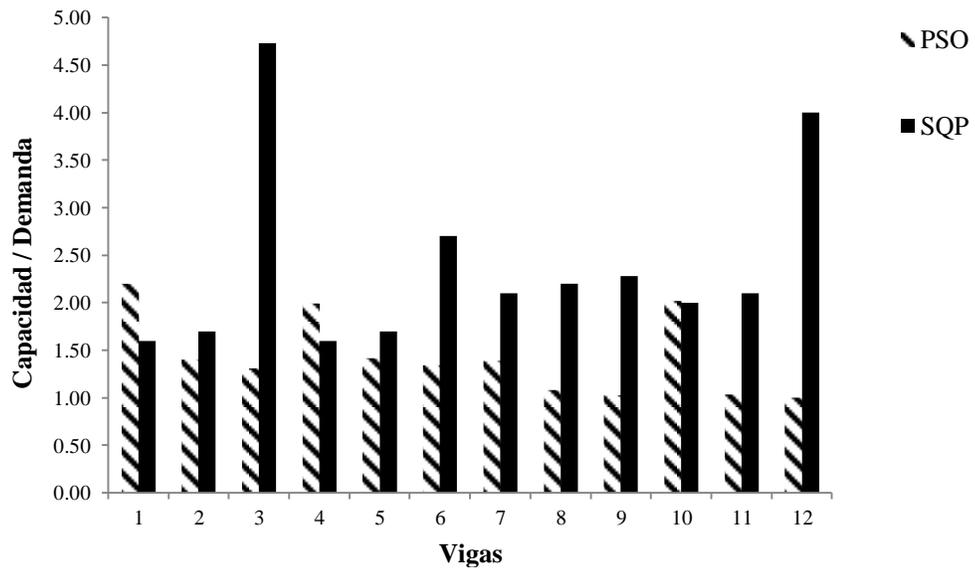


Figura 28. Relación capacidad/demanda para momento superior derecho en las vigas pórtico 1.

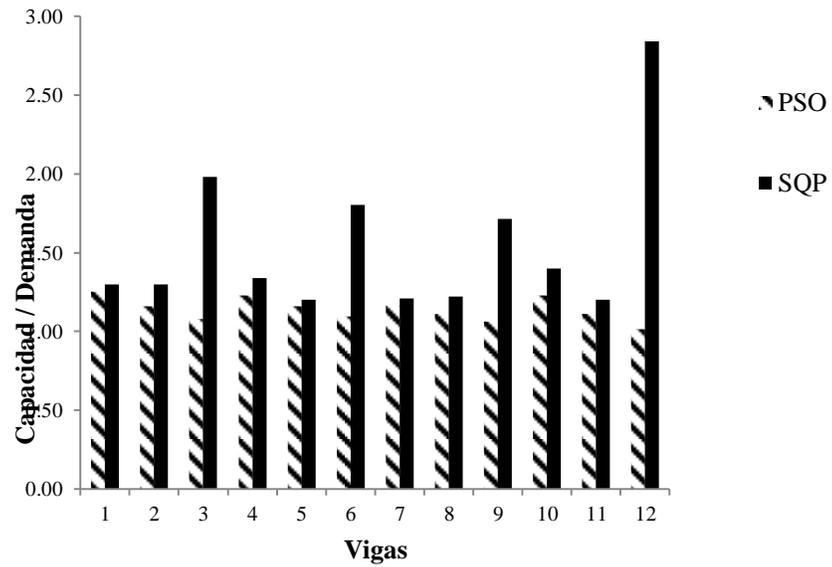


Figura 29. Relación capacidad/demanda para cortante nodo izquierdo en las vigas pórtico 1.

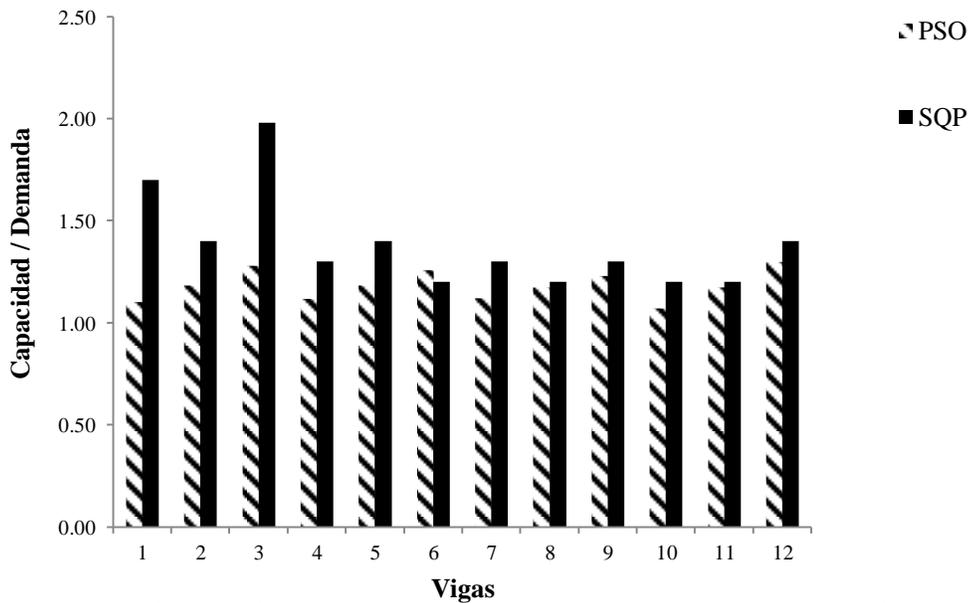


Figura 30. Relación capacidad/demanda para cortante nodo derecho en las vigas pórtico 1.

Las Figuras de la 26 a la 30 presentan la relación de capacidad/demanda para las vigas tanto por flexión como por cortante. En general, el diseño del *SQP*

posee relaciones de capacidad mayores al del *PSO*, lo cual es acorde con el mayor costo que tiene.

En las Figuras 29 y 30 se observa que a pesar de que dentro de las restricciones no se tomó en cuenta comparar la resistencia contra el cortante actuante, sino solo cumplir con las separaciones máximas, se obtuvieron relaciones capacidad/demanda siempre mayores a 1 por ambos algoritmos. Ahora, si bien para los ejemplos utilizados no fue necesario incluir el chequeo de la capacidad por cortante en las restricciones, de incrementarse la carga actuante probablemente sea necesario hacerlo.

A continuación se presentan los valores de las variables en los puntos óptimos encontrados por los algoritmos:

Tabla 7. Dimensiones de columna y área de acero a colocar para el diseño óptimo del *PSO*.

Grupo	b (mm)	h (mm)	As (mm ²)	ρ (Adim.)
1	300.00	350.00	3,837.00	0.037
2	350.00	350.00	3,939.00	0.032

Tabla 8. Dimensiones de columna y área de acero a colocar para el diseño óptimo del *SQP*.

Grupo	b (mm)	h (mm)	As (mm ²)	ρ (Adim.)
1	300.00	400.00	3,970.00	0.033
2	350.00	350.00	4,100.00	0.033

Tabla 9. Dimensiones de vigas y áreas de acero a colocar para el diseño óptimo del *PSO*.

Grupo	b (mm)	h (mm)	ρ (As izq sup)	ρ (As izq inf)	ρ (As cen. inf)	ρ (As cen. sup)	ρ (As der sup)	ρ (As der inf)	Sep. Izq. (mm)	Sep.Cen. (mm)	Sep. Der. (mm)
1	350	500	0.0112	0.0148	0.0057	0.0094	0.0134	0.0123	115	200	115
2	300	550	0.0084	0.0171	0.0061	0.0076	0.0079	0.0157	130	200	130

Tabla 10. Dimensiones de vigas y áreas de acero a colocar para el diseño óptimo del *SQP*.

Grupo	b (mm)	h (mm)	ρ (As izq sup)	ρ (As izq inf)	ρ (As cen. inf)	ρ (As cen. sup)	ρ (As der sup)	ρ (As der inf)	Sep. Izq. (mm)	Sep.Cen. (mm)	Sep. Der. (mm)
1	300	600	0.0102	0.0133	0.0036	0.0067	0.0117	0.0111	130	250	130
2	300	600	0.0102	0.0133	0.0036	0.0067	0.0117	0.0111	130	250	130

En cuanto a los diseños arrojados por los algoritmos, las cuantías en columnas se mantuvieron dentro del rango permitido (entre 0.02 y 0.04) siendo menores para el diseño del *SPQ*, debido a que las dimensiones de columna son mayores.

También en las vigas, ambos diseños cumplen con los requisitos de cuantía mínima y máxima, así como con las dimensiones mínimas de concreto.

4.3.2 Pórtico 2:

Tabla 11. Resultados del SQP para el pórtico 2.

Corrida	Costo Inicial	Costo Final	% Reducción	Iteraciones
1	215,200.00	136,000.00	36.80	156.00
2	210,000.00	136,000.00	35.24	184.00
3	197,700.00	136,000.00	31.21	176.00
4	197,400.00	136,000.00	31.10	171.00
5	212,500.00	136,000.00	36.00	175.00
6	204,400.00	136,000.00	33.46	149.00
7	214,200.00	136,000.00	36.51	140.00
8	204,500.00	136,000.00	33.50	163.00
9	217,800.00	136,000.00	37.56	149.00
10	213,300.00	136,000.00	36.24	158.00
11	199,000.00	136,000.00	31.66	168.00
12	211,100.00	136,000.00	35.58	177.00
13	208,600.00	136,000.00	34.80	183.00
14	209,690.00	136,000.00	35.14	166.00
15	209,400.00	136,000.00	35.05	143.00
16	212,500.00	136,000.00	36.00	165.00
17	198,300.00	136,000.00	31.42	177.00
18	179,200.00	136,000.00	24.11	187.00
19	204,700.00	136,000.00	33.56	183.00
20	212,400.00	136,000.00	35.97	187.00
21	207,400.00	136,000.00	34.43	184.00
22	211,300.00	136,000.00	35.64	170.00
23	202,300.00	136,000.00	32.77	182.00
24	201,300.00	136,000.00	32.44	154.00
25	215,800.00	136,000.00	36.98	181.00
26	186,100.00	136,000.00	26.92	178.00
27	193,900.00	136,000.00	29.86	177.00
28	222,220.00	136,000.00	38.80	184.00
29	213,300.00	136,000.00	36.24	164.00
30	207,700.00	136,000.00	34.52	179.00
Promedio	206,440.33	136,000.00	33.98	170.33

La Tabla 11, muestra los resultados del *SQP* para el pórtico 2 donde el número de variables es ahora 57. Se observa de nuevo que a pesar de que el algoritmo

inicia cada corrida desde puntos diferentes, la solución final en todas las corridas termina con el mismo valor, lo que refuerza la idea de que el algoritmo tiende a quedar atrapado en mínimos locales.

Tabla 12. Resultados del PSO para el pórtico 2.

Corrida	Costo Inicial	Costo Final	% Reducción	Iteraciones
1	186,500.00	132,900.00	28.74	543
2	189,500.00	138,000.00	27.18	125
3	182,900.00	123,700.00	32.37	559
4	187,000.00	153,500.00	17.91	179
5	167,700.00	117,900.00	29.70	445
6	180,800.00	126,200.00	30.20	194
7	177,900.00	145,800.00	18.04	153
8	190,100.00	131,200.00	30.98	497
9	170,100.00	127,400.00	25.10	258
10	197,800.00	131,200.00	33.67	200
11	178,300.00	130,300.00	26.92	330
12	173,600.00	124,800.00	28.11	410
13	182,000.00	154,400.00	15.16	186
14	172,000.00	129,800.00	24.53	332
15	187,900.00	126,400.00	32.73	204
16	178,200.00	128,700.00	27.78	209
17	182,800.00	122,800.00	32.82	147
18	178,800.00	143,300.00	19.85	111
19	180,800.00	131,700.00	27.16	330
20	175,800.00	138,600.00	21.16	288
21	170,000.00	142,000.00	16.47	250
22	182,400.00	131,800.00	27.74	117
23	187,400.00	132,800.00	29.14	313
24	190,400.00	137,300.00	27.89	137
25	184,000.00	135,600.00	26.30	171
26	177,400.00	148,800.00	16.12	116
27	175,800.00	131,700.00	25.09	228
-28	185,100.00	132,100.00	28.63	166
29	171,400.00	136,900.00	20.13	291
30	180,200.00	129,300.00	28.25	181
Promedio	180,586.36	133,290.91	26.11	275

La Tabla 12, muestra los valores encontrados por el *PSO*. En 19 de las 30 corridas realizadas, el *PSO* alcanzó mejorar la solución encontrada por el *SQP* y en los resultados de este ejemplo, se observa una mayor diferencia entre la mejor solución encontrada por el *SQP* y la del *PSO*, siendo la del segundo menor a la del primero en un 13.3%.

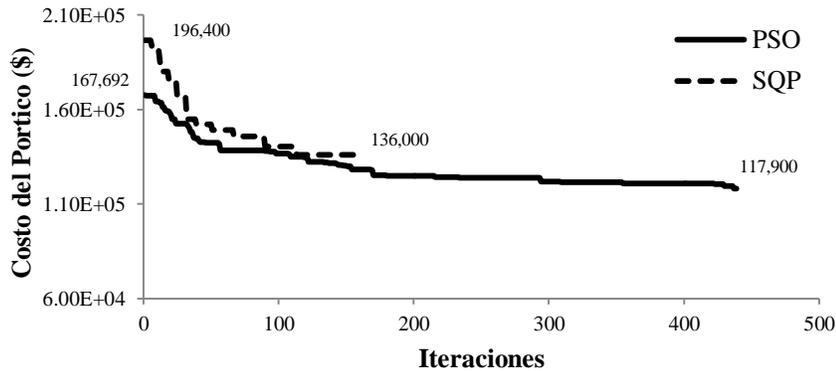


Figura 31. Comparación de convergencia de los algoritmos para el pórtico 2

En la Figura 31 se puede observar como el algoritmo *SQP* detiene la búsqueda mucho antes que el *PSO*, ya que no consigue disminuir la función objetivo, quedando de nuevo atrapado en un mínimo local.

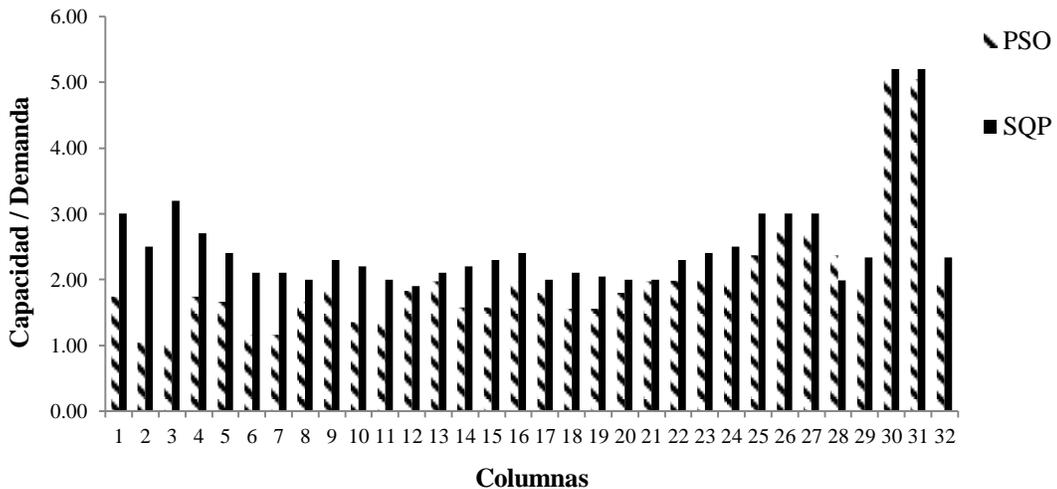


Figura 32. Relación capacidad/demanda para columnas del pórtico 2.

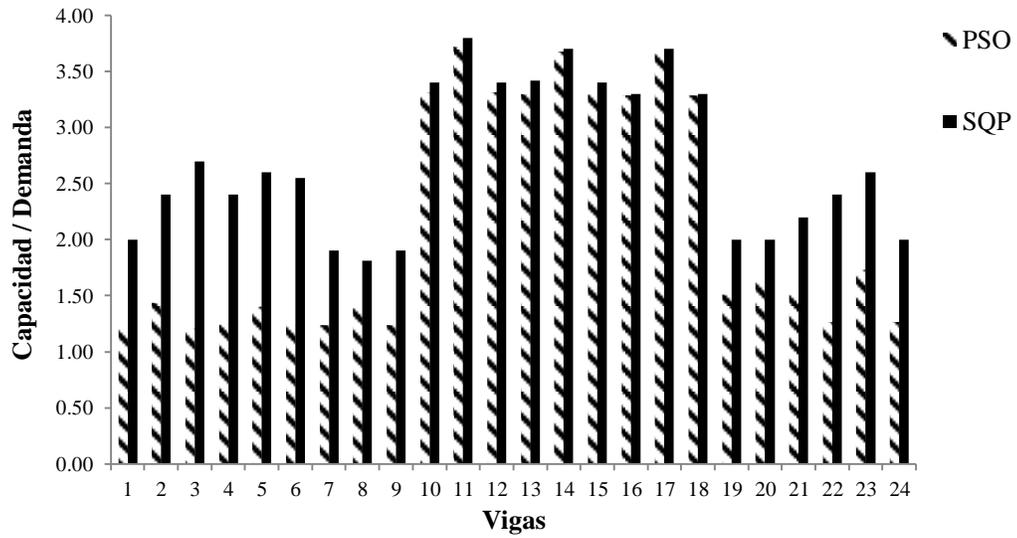


Figura 33. Relación capacidad/demanda para momento inferior central pórtico 2.

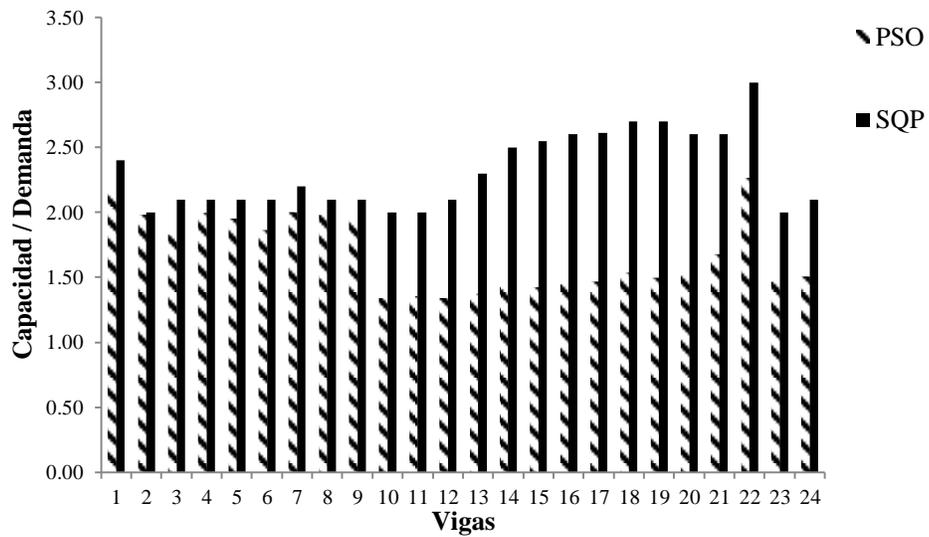


Figura 34. Relación capacidad/demanda para momento superior izquierdo pórtico 2.

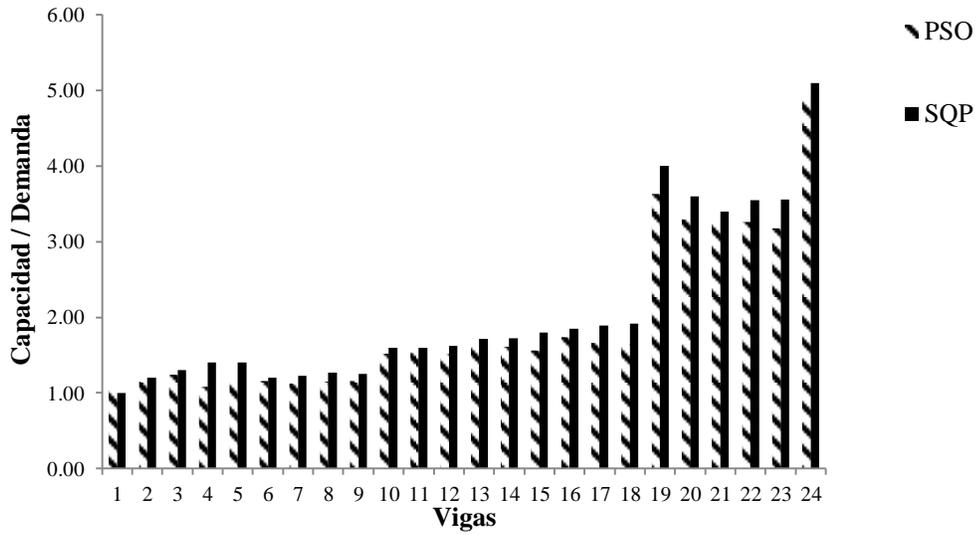


Figura 35. Relación capacidad/demanda para momento superior derecho pórtico 2.

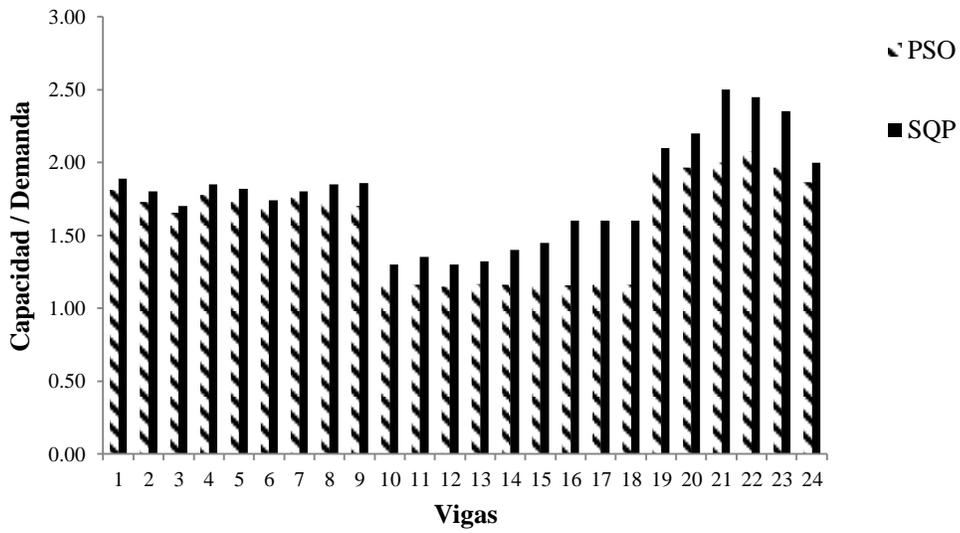


Figura 36. Relación capacidad/demanda para cortante nodo izquierdo pórtico 2.

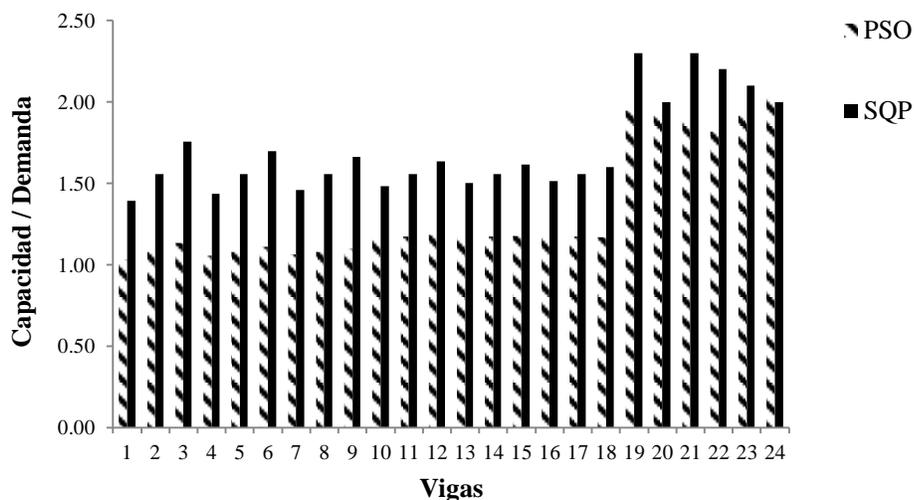


Figura 37. Relación capacidad/demanda para cortante nodo derecho pórtico 2.

Para el pórtico 2, se observa mayor diferencia entre las relaciones de capacidad/demanda de los elementos, en los diseños encontrados por el *SQP* y el *PSO*, quedando las del *SQP* siempre por encima de las de *PSO*, lo que coincide con el mayor costo del diseño. Se observa que en el diseño arrojado por el *PSO* hay muchos elementos que quedan con una relación capacidad/demanda cercana a 1.

Tabla 13. Dimensiones de columna y área de acero a colocar para el diseño óptimo del *PSO*.

Grupo	b (mm)	h (mm)	As (mm ²)	ρ (Adim.)
1	400	350	5480	0.0391
2	400	350	4800	0.0343
3	350	350	4003	0.0327
4	300	350	4000	0.0381

Tabla 14. Dimensiones de columna y área de acero a colocar para el diseño óptimo del *SQP*.

Grupo	b (mm)	h (mm)	As (mm ²)	ρ (Adim.)
1	450	450	7000	0.035
2	500	550	8500	0.031
3	400	450	6500	0.036
4	450	500	7000	0.031

Tabla 15. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del PSO.

Gru po	b (mm)	h (mm)	ρ (As izq sup)	ρ (As izq inf)	ρ (As central inf)	ρ (As central sup)	ρ (As der sup)	ρ (As der inf)	Sep. Izq. (mm)	Sep.Cen. (mm)	Sep. Der. (mm)
1	300	500	0.0184	0.0160	0.0058	0.0175	0.0067	0.0114	115	200	115
2	300	550	0.0118	0.0050	0.0141	0.0106	0.0181	0.0132	130	200	130
3	350	550	0.0090	0.0147	0.0053	0.0187	0.0143	0.0190	130	180	130

Tabla 16. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del SQP.

Gru po	b (mm)	h (mm)	ρ (As izq sup)	ρ (As izq inf)	ρ (As central inf)	ρ (As central sup)	ρ (As der sup)	ρ (As der inf)	Sep. Izq. (mm)	Sep.Cen. (mm)	Sep. Der. (mm)
1	300	700	0.0121	0.0110	0.0081	0.0031	0.0100	0.0043	70	200	70
2	300	700	0.0129	0.0083	0.0043	0.0086	0.0070	0.0069	127	200	127
3	350	700	0.0071	0.0086	0.0124	0.0052	0.0153	0.0117	127	300	127

En los diseños de las columnas de ambos métodos, se cumple con la restricción de que las dimensiones y áreas de acero del grupo 1 deben ser mayores a las del grupo 3, al igual que las del grupo 2 con respecto a las del 4. El diseño de vigas encontrado por el *SQP* tiene 200 mm más de concreto en la profundidad, lo que lo hace más costoso, además que se incrementan las cuantías de acero cumplir con los requisitos mínimos.

4.3.3 Pórtico 3:

Tabla 17. Resultados del SQP para el pórtico 3.

Corrida	Costo Inicial	Costo Final	% Reducción	Iteraciones
1	280,200.00	218,300.00	22.09	159.00
2	282,000.00	218,300.00	22.59	160.00
3	308,500.00	218,300.00	29.24	181.00
4	307,300.00	218,300.00	28.86	183.00
5	305,600.00	218,300.00	28.53	190.00
6	301,000.00	218,300.00	27.44	200.00
7	296,100.00	218,300.00	26.27	173.00
8	300,500.00	218,300.00	27.35	195.00
9	303,200.00	218,300.00	27.95	205.00
10	295,300.00	218,300.00	26.08	209.00
11	300,500.00	218,300.00	27.35	180.00
12	290,300.00	218,300.00	24.77	179.00
13	295,300.00	218,300.00	26.08	201.00
14	317,900.00	218,300.00	31.31	191.00
15	300,900.00	218,300.00	27.22	162.00

16	278,800.00	218,300.00	21.70	206.00
17	307,900.00	218,300.00	29.10	171.00
18	291,500.00	218,300.00	25.11	178.00
19	307,400.00	218,300.00	28.99	176.00
20	289,900.00	218,300.00	24.70	189.00
21	307,000.00	218,300.00	28.89	173.00
22	304,000.00	218,300.00	28.19	193.00
23	301,200.00	218,300.00	27.52	197.00
24	297,400.00	218,300.00	26.60	221.00
25	285,600.00	218,300.00	23.56	178.00
26	291,400.00	218,300.00	25.09	212.00
27	325,100.00	218,300.00	32.85	204.00
28	283,900.00	218,300.00	23.11	220.00
29	312,700.00	218,300.00	30.19	194.00
30	278,000.00	218,300.00	21.47	170.00
Promedio	298,213.33	218,300.00	26.67	188.33

En los resultados del *SQP* para pórtico 3, se observa que el algoritmo nuevamente converge siempre a un mismo valor en el espacio de búsqueda. Este valor óptimo es superado en 27 de las 30 corridas del *PSO*, lo que indica que para este problema de 69 variables el desempeño del *PSO* es mejor al del *SQP*. En la Tabla 18 se resaltan los valores en los cuales el *PSO* mejoró el valor óptimo arrojado por el *SQP*. La máxima disminución fue del 16 %.

Tabla 18. Resultados del PSO para el pórtico 3.

Corrida	Costo Inicial	Costo Final	% Reducción	Iteraciones
1	243,700.00	204,500.00	16.09	253.00
2	258,600.00	209,600.00	18.95	209.00
3	258,300.00	204,900.00	20.67	136.00
4	242,600.00	197,200.00	18.71	309.00
5	265,400.00	193,900.00	26.94	171.00
6	253,700.00	217,500.00	14.27	188.00
7	250,700.00	193,600.00	22.78	173.00
8	267,100.00	205,400.00	23.10	333.00
9	262,300.00	197,700.00	24.63	225.00
10	254,200.00	209,200.00	17.70	361.00
11	273,300.00	227,100.00	16.90	150.00
12	252,300.00	202,200.00	19.86	178.00
13	243,700.00	223,100.00	8.45	128.00
14	270,100.00	199,600.00	26.10	184.00
15	251,900.00	205,100.00	18.58	250.00

16	245,500.00	190,000.00	22.61	217.00
17	250,300.00	213,800.00	14.58	236.00
18	267,500.00	214,800.00	19.70	262.00
19	251,300.00	199,100.00	20.77	455.00
20	249,500.00	198,700.00	20.36	285.00
21	270,300.00	183,300.00	32.19	508.00
22	272,000.00	204,800.00	24.71	410.00
23	249,000.00	197,900.00	20.52	693.00
24	251,900.00	200,900.00	20.25	198.00
25	246,300.00	186,800.00	24.16	179.00
26	242,700.00	196,200.00	19.16	245.00
27	275,300.00	194,300.00	29.42	561.00
28	262,900.00	221,900.00	15.60	601.00
29	279,200.00	191,400.00	31.45	298.00
30	263,400.00	189,900.00	27.90	338.00
Promedio	257,500.00	202,480.00	21.24	291.13

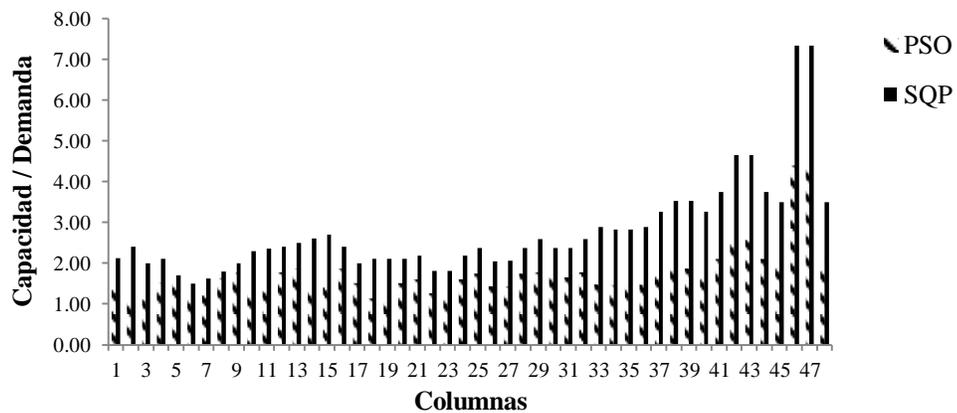


Figura 38. Relación capacidad/demanda para las columnas pórtico 3.

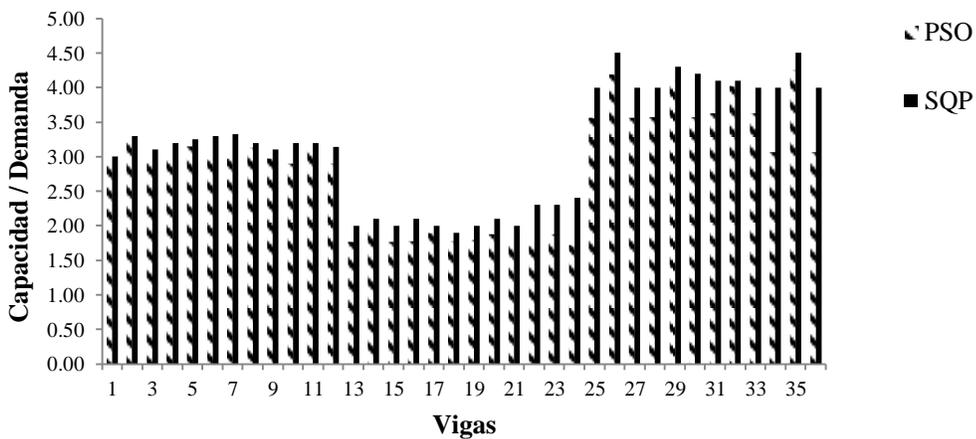


Figura 39. Relación capacidad/demanda para momento inferior central pórtico 3.

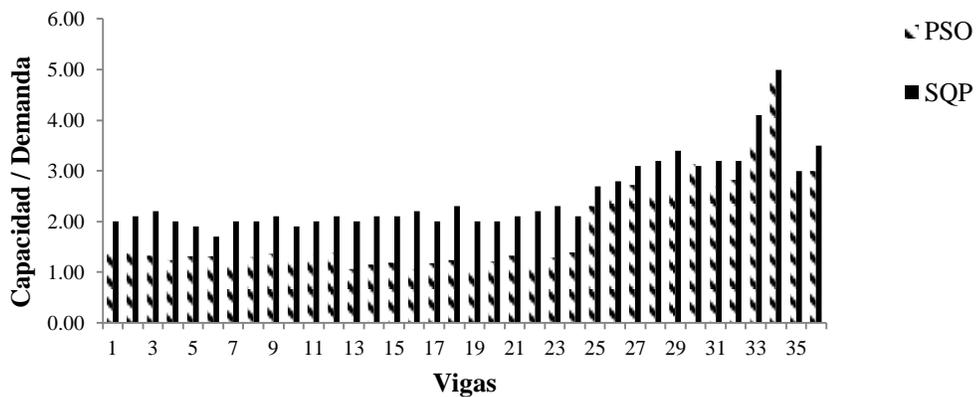


Figura 40. Relación capacidad/demanda para momento superior izquierdo pórtico 3.

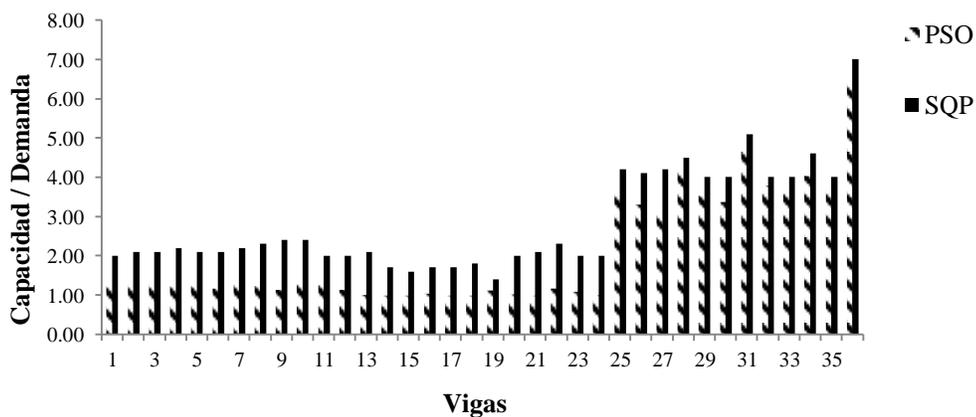


Figura 41. Relación capacidad/demanda para momento superior derecho pórtico 3.

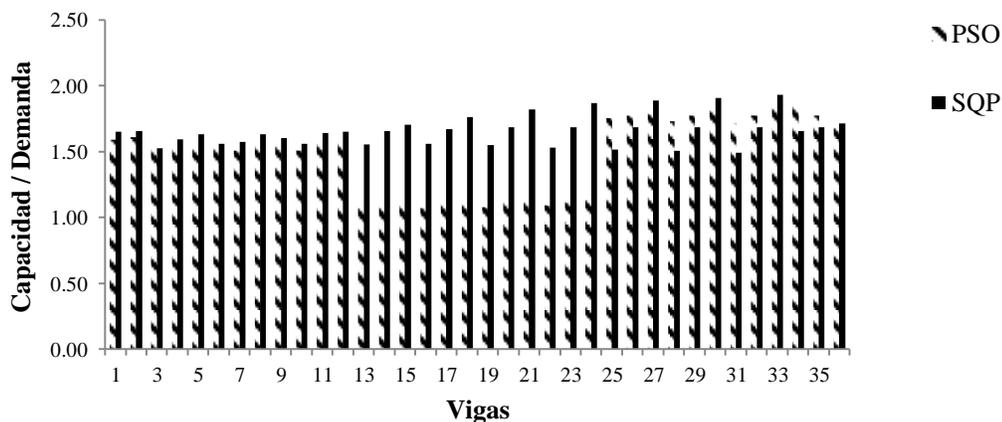


Figura 42. Relación capacidad/demanda para cortante en el nodo izquierdo pórtico 3.

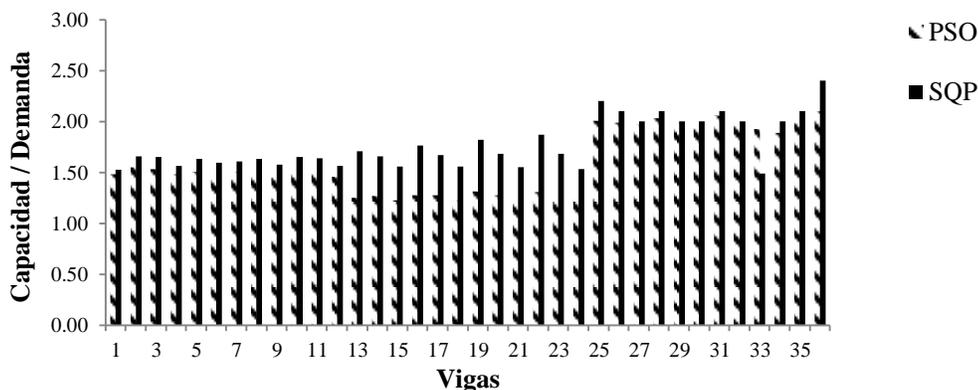


Figura 43. Relación capacidad/demanda para cortante en el nodo derecho pórtico 3.

De las Figuras 38 a la 42 se observa que el diseño encontrado por el SQP tiene para todos los elementos mayor relación de capacidad/demanda en comparación con el del PSO lo cual de nuevo, es consecuente en el mayor costo asociado. Se observa que en el diseño de PSO la mayoría de las columnas quedaron con un relación capacidad/demanda cercana a 1 igual que para la resistencia a flexión de las vigas.

Tabla 19. Dimensiones de columna y área de acero a colocar para el diseño óptimo del PSO.

Grupo	b (mm)	h (mm)	As (mm ²)	ρ (Adim.)
1	500	450	7000	0.0311
2	500	450	7000	0.0311
3	450	450	4600	0.0227
4	400	400	4400	0.0275
5	400	300	4100	0.0342
6	350	350	4200	0.0343

Tabla 20. Dimensiones de columna y área de acero a colocar para el diseño óptimo del SQP.

Grupo	b (mm)	h (mm)	As (mm ²)	ρ (Adim.)
1	600	500	9000	0.0300
2	600	600	8900	0.0247
3	600	500	7100	0.0237
4	550	500	7200	0.0262
5	550	500	5700	0.0207
6	550	500	6000	0.0218

En los diseños de columnas arrojados por ambos algoritmos para el pórtico 3, se observa que en ambos diseños se respeta el criterio de que las columnas de pisos inferiores deben tener mayor dimensión y área de acero que las que superiores. Así, las columnas del grupo 1, son más resistentes que las del grupo 3 y a la vez, estas más que las del 5. Lo mismo para las columnas del grupo 2, 4 y 6. El diseño óptimo del *SQP* incluye mayores cuantías de acero y áreas de concreto que el del *PSO*.

Tabla 21. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del PSO.

	b (mm)	h (mm)	ρ (As izq sup)	ρ (As izq inf)	ρ (As central inf)	ρ (As central sup)	ρ (As der sup)	ρ (As der inf)	Sep. Izq. (mm)	Sep.Cen. (mm)	Sep. Der. (mm)
1	300	500	0.0184	0.0160	0.0058	0.0175	0.0067	0.0114	100	170	100
2	300	550	0.0118	0.0050	0.0141	0.0106	0.0181	0.0132	90	170	90
3	350	550	0.0090	0.0147	0.0053	0.0187	0.0143	0.0190	120	200	120

Tabla 22. Dimensiones de vigas y área de acero a colocar para el diseño óptimo del SQP.

	b (mm)	h (mm)	ρ (As izq sup)	ρ (As izq inf)	ρ (As central inf)	ρ (As central sup)	ρ (As der sup)	ρ (As der inf)	Sep. Izq. (mm)	Sep.Cen. (mm)	Sep. Der. (mm)
1	300	700	0.0121	0.0110	0.0081	0.0031	0.0100	0.0043	70	200	70
2	300	700	0.0129	0.0083	0.0043	0.0086	0.0071	0.0069	110	200	110
3	300	700	0.0071	0.0086	0.0124	0.0052	0.0153	0.0117	90	200	90

4.4 Discusión General

Si bien para los tres pórticos planteados las soluciones arrojadas por el algoritmo *SQP* cumplieron con las restricciones planteadas, en términos de costo, estos diseños no son las mejores soluciones para cada problema. Que el algoritmo no logre mejorar sus soluciones encontradas, es una desventaja, ya

que queda atrapado en mínimos locales. La razón de esto, es que el algoritmo *SQP* en cada iteración genera una sola solución para el problema, y además, se basa en la determinación de una dirección en la que disminuya la función objetivo en cada punto. Si la búsqueda llega a un punto desde el cual no se puede disminuir el valor de la función objetivo, el algoritmo se detendrá. Por otro lado, el *PSO* genera en cada iteración tantas soluciones como partículas haya en el enjambre, lo que le da la posibilidad de explorar mucho más el espacio de búsqueda en comparación al *SQP*, lo cual es una ventaja de la filosofía de optimización. También, el *PSO* tiene la posibilidad de salir de estos mínimos locales a través de los cambios aleatorios incluidos en su algoritmo.

A diferencia de otras investigaciones como (Kaveh & Sabzi, 2011), (Gharehbaghi & Fadaee, 2012), (Payá et al., 2006) y (Flores & Rodríguez, 2010) se incluyeron hasta 11 variables por cada grupo de viga en la metodología de optimización, lo cual hizo más complejo el proceso de optimización. Sin embargo, ambos algoritmos lograron encontrar soluciones factibles para el problema así planteado, lo cual indica que ambos algoritmos son capaces de manejar problemas de múltiples variables.

En la metodología de optimización, se encontró que hacer un único análisis inicial de la estructura para hallar las fuerzas internas y re-analizar la estructura solo al final del diseño para chequear resistencia, permite una disminución del tiempo de ejecución del algoritmo (menos de 60 s por corrida). Esto mejora el desempeño de otras metodologías como la de Payá et al., (2006), donde se reporta que tiempo necesario por corrida fue de 21 horas.

En el trabajo de Kaveh & Sabzi, (2011), se utilizaron los mismos ejemplos de prueba que se usaron en esta investigación. La comparación se puede hacer en términos de la reducción de costo obtenida, ya que en Kaveh & Sabzi, (2011), los precios unitarios de los materiales son diferentes a los que se utilizaron aquí, y no se podría comparar directamente el resultado final. El número de partículas usada fue de 100 para el pórtico 1, 200 para el pórtico 2 y 300 para el 3.

Tabla 23. Resultados reportados por Kaveh & Sabzi, (2011) para el pórtico 1.

Costo Inicial	Costo Final	% Reducción
27,500	21,000	23.64

Tabla 24. Resultados reportados por Kaveh & Sabzi, (2011) para el pórtico 2.

Costo Inicial	Costo Final	% Reducción
65500	48000	26.72

Tabla 25. Resultados reportados por Kaveh & Sabzi, (2011) para el pórtico 3.

Costo Inicial	Costo Final	% Reducción
102000	82000	19.61

Hay que señalar que en (Kaveh & Sabzi, 2011), se reporta solo el resultado de una corrida del algoritmo por cada ejemplo. En esta investigación utilizando el *PSO* y con solo 40 partículas en el enjambre se lograron reducciones de hasta un 40 % en el pórtico 1. Usando 60 partículas para el pórtico 2, se lograron reducciones de 37.5 % y para de 32.19 % para el pórtico 3 usando 100 partículas. Esto indica que el *PSO* auto-adaptativo usado en esta metodología tiene un mejor desempeño en comparación al presentado por (Kaveh & Sabzi, 2011), ya que puede reducir más el valor de la función objetivo, con menos partículas en el enjambre.

Finalmente, en la metodología de optimización usada se buscó reducir el costo inicial de las estructuras mostradas. Como se vio en los resultados, esto conduce a diseños con relaciones capacidad/demanda cercanas a 1. Es posible incluir en la metodología el costo a largo plazo asociado a la ocurrencia de un posible sismo, lo cual convertiría el problema de optimización en uno multi-objetivo. Lo anterior se puede hacer utilizando el *PSO* auto-adaptativo utilizado en este trabajo, en vista de los buenos resultados obtenidos.

5. Conclusiones

Se demostró que la metodología de optimización aplicada al diseño de pórticos de concreto es una alternativa frente al diseño tradicional, obteniéndose diseños que cumplen requisitos normativos mediante la herramienta computacional. Si bien en esta investigación se usaron las restricciones DMO de la norma NSR-10, la metodología se puede extender a pórticos DES.

Se encontró que la función objetivo utilizada en la optimización de los pórticos planos de concreto armado, tiene múltiples mínimos locales lo cual complica el proceso de búsqueda de un valor óptimo.

De los algoritmos estudiados si bien ambos lograron diseños que cumplen las restricciones normativas, se encontró que a medida que la complejidad del problema aumenta el *PSO* tiene un mejor desempeño comparado con el *SQP* al arrojar soluciones más económicas para los problemas planteados. Además, se mostró que el *SQP* tiende a quedar atrapado en mínimos locales, lo cual es una desventaja cuando en el problema de optimización aumenta el número de variables.

Si bien los diseños arrojados por el *PSO* son apropiados debido a que cumplen con los requisitos normativos, los elementos en estos diseños quedan con relaciones capacidad/demanda cercanas a 1, lo que indica que la sobre-resistencia en ellos proviene solo de los factores de seguridad normativos. Es aquí donde hay que resaltar que la metodología de optimización es solo una herramienta de diseño, y que el ingeniero estructural debe usarla con buen juicio.

Se recomienda incluir como función objetivo el costo a largo plazo de la estructura, en el que se tome cuenta la inversión requerida para reparar daños en la estructura en caso de la ocurrencia de un sismo. Para ello, es necesario también pasar del análisis lineal usado en esta investigación, algún método no-lineal.

6. Referencias Bibliográficas

- ACI 318-11. (n.d.). Building code requirements for structural Concrete.
- Arora, J. (2012). *Introduction to Optimum Design* (Third Edition.). Elsevier.
- Arumugam, M. S., & Rao, M. V. C. (2008). On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. *Applied Soft Computing*, 8(1), 324–336.
- Ballig, R., & Yao, X. (1997). Optimization of Reinforced Concrete Frames. *Journal of Structural Engineering*, 123(2), 193–202.
- Belegundu, A. D., & Chandrupatla, P. T. R. (2011). *Optimization Concepts and Applications in Engineering* (2nd ed.). Cambridge University Press.
- Carballo, F. P., Delgado, J. C., Canteli, A. F., & Rey, M. J. L. (2006). *Cálculo matricial de estructuras*. Universidad de Oviedo.
- Construdata. (2014). *Informe Especial Mano de Obra*. Legis.
- Flores, J. L., & Rodríguez, G. (2010). Optimización de secciones transversales de pórticos de hormigón armado mediante uso de la técnica de programación cuadrática secuencial SQP. *Mecánica Computacional*, Vol XXIX, 9719–9738.
- Gharehbaghi, S., & Fadaee, M. J. (2012). Design optimization of RC frames under earthquake loads. *INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING*, 2(4), 459–477.
- Guo, L., & Chen, X. (2009). A Novel Particle Swarm Optimization Based on the Self-Adaptation Strategy of Acceleration Coefficients (Vol. 1, pp. 277–281). Presented at the International Conference on Computational Intelligence and Security, 2009. CIS '09.
- Kaveh, A., & Sabzi, O. (2011). A comparative study of two meta-heuristic algorithms for optimum design of reinforced concrete frames. *International Journal of Civil Engineering*, 9(3).
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In , *IEEE International Conference on Neural Networks, 1995. Proceedings* (Vol. 4, pp. 1942–1948 vol.4).
- Nilson, A. H., & Winter, G. (1994). *Diseño de Estructuras de Concreto* (11a Edición.). McGraw-Hill, InC.

- Nocedal, J., & Wright, S. (2006). *Numerical Optimization* (2da. Edición.). Springer.
- NSR-10. (2010). *Reglamento Colombiano de Construcciones Sismo Resistentes*. Bogotá: Asociación Colombiana de Ingeniería Sísmica.
- Parsopoulos, K. E., & Vrahatis, M. N. (2007). Parameter selection and adaptation in Unified Particle Swarm Optimization. *Mathematical and Computer Modelling*, 46(1–2), 198–213.
- Payá, I., Yepes, V., Clemente, J. J., & Gonzalez, F. (2006). Optimización heurística de pórticos de edificación de hormigón armado. *Revista Internacional de Métodos Numéricos Para Cálculo Y Diseño En Ingeniería*.
- Ratnaweera, A., & Halgamuge, S. K. (2004). Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 8, NO. 3.
- Rochel, R. (2012). *Análisis y diseño sísmico de edificios* (2da Edición.). Medellín, Colombia: EAFIT.
- Ruszczynski, A. (2006). *Nonlinear Optimization*. Princeton, N.J: Princeton University Press.
- Segura, F. (2011). *Estructuras de Concreto I* (7ma Edición.). Universidad Nacional de Colombia.
- Shi, Y., & Eberhart, R. (1998). A Modified Particle Swarm Optimizer. *IEEE World Congress of Computational Intelligence*, pp. 69–73.
- Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. John Wiley & Sons.
- Tena, A. (2007). *Análisis de Estructuras con Métodos Matriciales*. Limusa.
- Tsoulos, I. G., & Stavrakoudis, A. (2010). Enhancing PSO methods for global optimization. *Applied Mathematics and Computation*, 216(10), 2988–3001.
- Uribe E., J. (2000). *Análisis de Estructuras* (2da Edición.). Editorial Escuela Colombiana de Ingeniería.
- Wilde, D., & Biegthler, C. (1976). *Teoría de optimización* (1era. Edición, en español.). Urmo, S.A de ediciones.
- Zhan, Z.-H., Zhang, J., Li, Y., & Chung, H. S.-H. (2009). Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6), 1362–1381.