

*Universidad de Matanzas
Facultad de Ciencias Técnicas*



**OPTIMIZACIÓN ESTRUCTURAL DE ARMADURAS PLANAS
EMPLEANDO TÉCNICAS HEURÍSTICAS (ALGORITMOS GENÉTICOS)**

Trabajo de Diploma en Ingeniería Civil

Autor(a): Anisley Chávez Hernández.

Tutor: Alejandro Hernández Hernández.

Matanzas, 2018

PENSAMIENTO

“Así como la selección natural trabaja exclusivamente para y por el bien de cada ser viviente, todas las dotes mentales y corporales tienden a progresar en dirección a la perfección.”

Charles Darwin, El Origen de las Especies (1859)

DECLARACIÓN DE AUTORIDAD

Por medio de la presente declaro que soy la única autora de este trabajo de diploma y, en calidad de tal, autorizo a la Universidad de Matanzas a darle el uso que estime más conveniente.

NOTA DE ACEPTACIÓN

Miembros del Tribunal:

_____	_____	_____
Presidente	Secretario	Vocal

DEDICATORIA:

“En especial a mi abuelita, que es la mejor super abuela del mundo. A mi mamá, mi papá y mi hermano, las tres personas más influyentes en mi vida. A mis tíos que sin ellos no lo hubiese logrado. A mi familia, a mi novio y a mis amigos que me dieron su apoyo cuando más lo he necesitado.”

AGRADECIMIENTOS:

- A mi abuelita María por ser Genial y estar pendiente a mi todo el tiempo.
- A mis padres por hacer hasta lo imposible por apoyarme en este largo camino y complacerme en TODO.
- A mi hermano por ser incondicional, por recordarme cada segundo que sí puedo, por sus celos también y en especial por ser el mejor hermano-padre que existe.
- A mis tíos Ofelia y Jaime, porque son los principales responsables de que esté hoy aquí.
- A mis cuatro preciosos sobrinos que son los únicos que tienen el poder de hacerme olvidar todos los problemas que pueda tener con un simple “tía”.
- A mi novio Junior por aguantarme todo este tiempo, pero en especial por su apoyo, comprensión y Amor.
- A todos mis tíos en especial a Norma, Martha, Ramona, Emelina, Carlos, Jesús, Esteban que siempre han estado ahí para mí.
- A mi tía Rosi, que siempre decía que yo iba a ser una gran ingeniera, y por eso trataré de complacerla.
- A mi prima Betsy en especial, y a mis amigas Lorena, Gabriela, Graciela porque cuando todos corren ellas deciden quedarse...
- A mis primos Jimmito, Yailyn, Yudelky, Taimara, Lisi, Arianna, Yeny, Yuniel, Yuny, Yady, a todos por ser tan especiales conmigo.
- A mis amigas de siempre Merlyn, Ana, Yuliet, que a pesar de la distancia nos mantenemos unidas.
- A mi cuñada Yaima, y a Jorge Inda porque su ayuda fue fundamental para la realización de esta tesis.

- A mi tutor Alejandro por la confianza y paciencia y porque siempre encuentra un espacio para atenderme.
- A Danae por su super apoyo y amistad en estos cinco años y en los que vendrán.....
- A mis compañeros de la universidad por todos los momentos (por los de fiesta y los de estudio) que hemos vivido juntos, en especial aLusda, Cheyla, Luis I, Wilma, Cesar, Richard.
- A los profesores que me impartieron clases durante estos cinco años y a los de enseñanzas pasadas que aportaron a mi formación profesional.
- A todas las personas que de una forma u otra hicieron posible este sueño...

A todos, Muchas Gracias.

RESUMEN

En el pasado, muchos de los procesos de diseño fueron realizados por la experiencia e intuición del diseñador en vez de una aplicación intensiva de la teoría de optimización. Recientemente esta forma de pensar ha cambiado debido a la importancia que ha tomado el campo de la optimización estructural en el diseño, ya que mediante su aplicación se logra reducir costos, materiales y tiempo. El propósito de aplicar el concepto de diseño óptimo a la ingeniería estructural es el de obtener una solución a un problema de ingeniería que cumpla con todas las limitaciones y restricciones impuestas, y que a la vez resulte ser la mejor en cuanto a uno o varios criterios de diseño previamente establecidos. En esta investigación se pretende lograr una configuración más racional de las armaduras planas mediante la optimización del diseño de estructuras utilizando técnicas heurísticas (Algoritmos Genéticos), para ello se realiza el análisis de las tres tipologías de armaduras planas que aparecen en el *software SAP2000* a través de una programación elaborada en el *software MATLAB* empleando dichas técnicas de optimización, posteriormente se compara con el diseño de éstas mismas tipologías pero realizando un predimensionamiento basado en la experiencia del diseñador con la finalidad de comprobar cuan eficiente pueden ser éstas técnicas al ser aplicadas en el diseño de armaduras planas.

Palabras claves: optimización, armaduras planas, configuración racional, técnicas heurísticas.

ABSTRACT

In the past, many of the design processes were made by the experience and intuition of the designer instead of an intensive application of optimization theory. Recently this way of thinking has changed due to the importance that the field of structural optimization has taken in the design, since through its application it is possible to reduce costs, materials and time. The purpose of applying the concept of optimal design to structural engineering is to obtain a solution to an engineering problem that complies with all the limitations and restrictions imposed, and which at the same time is the best in terms of one or more criteria. Of previously established design. This research aims to achieve a more rational configuration of the flat reinforcements by optimizing the design of structures using heuristic techniques (Genetic Algorithms), for which the design of the three types of flat reinforcements that appear in the *SAP2000 software* is carried out. Through a programming developed in the *MATLAB software* using these optimization techniques, it is later compared with the design of these same typologies but making a pre-dimensioning based on the experience of the designer in order to check how efficient these techniques can be when applied in the design of flat armour.

Keywords: optimization, flat armours, rational configuration, heuristic techniques.

TABLA DE CONTENIDO

Introducción	1
Capítulo 1 Marco Teórico Referencial	8
1.1 Optimización.....	8
1.1.1 Función objetivo	10
1.1.2 Restricción	12
1.1.3 Espacio de búsqueda.....	12
1.1.4 Variables	13
1.1.5 Óptimo local, Óptimo global y de Pareto	13
1.2 Antecedentes de los algoritmos evolutivos genéticos.....	14
1.3 Algoritmos genéticos	17
1.3.1 Población.....	22
1.3.2 Función de evaluación (<i>Fitness</i>)	22
1.3.3 Operadores Genéticos	23
1.3.4 Selección	23
1.3.5 Cruzamiento o Reproducción	23
1.3.6 Mutación	23
1.3.7 Elitismo	24
1.3.8 Ventajas, desventajas y limitaciones de los algoritmos genéticos.....	25
1.4 Técnicas heurísticas de optimización.....	26
1.4.1 Ventajas sobre las técnicas tradicionales.....	28
1.5 Armaduras planas. Conceptualización, utilización, principales tipologías.....	30
1.5.1 Armaduras planas.....	30
1.5.2 Tipologías de armaduras planas.....	31
Capítulo 2 Análisis de las diferentes tipologías de armaduras y funcionamiento de los Algoritmos Genéticos	38
2.1 Métodos para la determinación de las funciones.....	38
2.2 Breve reseña de los métodos de los nudos y de las secciones	40
2.2.1 Método de los nudos.....	40
2.2.2 Método de las secciones.....	41
2.3 Descripción de las diferentes tipologías.....	42
2.4 Definición de las funciones a optimizar y las variables que influyen en ella.....	44
2.5 Análisis de los parámetros de AG.....	45
2.5.1 PopulationType	45
2.5.2 PopInitRange.....	46
2.5.3 PopulationSize	46
2.5.4 EliteCount	46
2.5.5 CrossoverFraction	46
2.5.6 ParetoFraction	46
2.5.7 MigrationDirection	47
2.5.8 MigrationInterval	47
2.5.9 MigrationFraction	47
2.5.10 Generations	47

2.5.11 TimeLimit	47
2.5.12 FitnessLimit	48
2.5.13 StallGenLimit.....	48
2.5.14 TolFun.....	48
2.5.15 TolCon	48
2.5.16 InitialPopulation.....	48
2.5.17 InitialScores	48
2.5.18 Initial Penalty.....	49
2.5.19 PenaltyFactor	49
2.5.20 CreationFcn.....	49
2.5.21 FitnessScalingFcn	49
2.5.22 SelectionFcn.....	49
2.5.23 CrossoverFcn	50
2.5.24 MutationFcn.....	50
2.5.25 DistanceMeasureFcn.....	51
2.5.26 HybridFcn	Error! Bookmark not defined.
2.5.27 Display	51
2.5.28 OutputFcns.....	51
2.5.29 PlotFcns.....	51
2.5.30 PlotInterval.....	51
2.5.31 Vectorized.....	51
2.5.32 UseParallel	52
Capítulo 3 Caso de estudio	53
3.1 Análisis de los resultados obtenidos en cada caso de estudio con las mejores heurísticas.	53
3.2 Posibles soluciones.	53
3.2.1 Solución propuesta por el diseñador (Sloped).	54
3.2.2 Solución Optimizada (Sloped).	55
3.2.3 Solución propuesta por el diseñador (Pratt).....	57
3.2.4 Solución Optimizada (Pratt).	58
.....	59
3.2.5 Solución propuesta por el diseñador (Vertical).	60
3.2.6 Solución Optimizada (Vertical).	61
Conclusiones.....	64
Recomendaciones	65
Referencias Bibliográficas	66
Anexos	73
Anexo 1 Configuración de la Armadura Sloped.....	73
Anexo 2 Configuración de la Armadura Pratt	77
Anexo 3 Configuración de la Armadura Vertical	85
Anexo 4 Programación (Sloped).....	111
Anexo 5 Programación (Pratt)	112
Anexo 6 Programación (Vertical).....	113

INTRODUCCIÓN

En la inteligencia artificial, la programación genética es una metodología basada en los algoritmos evolutivos, e inspirada en la evolución biológica para desarrollar automáticamente programas de computadoras que realicen una tarea definida por el usuario. Es una técnica de aprendizaje automático utilizada para optimizar una población evaluando su capacidad de acuerdo a una función de ajuste o aptitud, permitiendo al usuario elegir la solución más idónea. (Sonmez, 2007)

La programación genética es una efectiva técnica de búsqueda que se basa en la selección natural, la idea básica es combinar buenas soluciones para un cierto problema para gradualmente mejorar el resultado. Todas las soluciones son inicialmente creadas al azar, y ellas están individualmente representadas por una cuerda binaria con algunas similitudes. La importancia de minimizar el número de posibilidades está enfatizada en generar problemas razonables de optimización que se refieren generalmente a las tres categorías del diseño; el tamaño, la forma y la topología. (Berzal)

Como metodología científica, la heurística es aplicable a cualquier ciencia e incluye la elaboración de medios auxiliares, principios, reglas, estrategias y programas que faciliten la búsqueda de vías de solución a problemas; o sea, para resolver tareas de cualquier tipo para las que no se cuente con un procedimiento algorítmico de solución. Según Horst Müller: *Los procedimientos heurísticos son formas de trabajo y de pensamiento que apoyan la realización consciente de actividades mentales exigentes.*

Galileo Galilei fue el primer científico en estudiar la optimización estructural (Galilei, 1638), esta disciplina ha ido evolucionado con el paso del tiempo hasta convertirse en un área de ingeniería llamada optimización estructural. El interés creciente en esta área en las últimas décadas es debido a la disponibilidad de computadoras baratas y poderosas, junto con los desarrollos rápidos en los métodos de análisis estructural y de optimización que juega un papel principal en muchos campos de ingeniería.

Por otra parte numerosos problemas a los que se enfrentan los ingenieros civiles están descritos por la optimización de estructuras. En este trabajo se realiza un exhaustivo estudio y análisis de una de las técnicas heurísticas (Algoritmos Genéticos) para optimizar diferentes tipologías estructurales de armaduras.

Las armaduras derivan su utilidad y apariencia distintiva de su construcción sencilla, pues los elementos de las barras ejercen sólo fuerzas concéntricamente armadas axiales y se conectan con soldaduras o las juntas fijadas con pernos. Estas estructuras son ubicadas constantemente en el mundo industrializado pero a la hora de modelarlas es sumamente complicado, ya sea en obras de mucha o poca envergadura.(Rivera, 2015)

Por ejemplo consecuentemente el modelado a mano, de la Torre Eiffel que es una estructura de hierropudelado diseñada por Maurice Koechlin y Émile Nouguier, construida por el ingenierofrancésGustave Eiffel y una de las estructuras más famosas del mundo, que contiene más de 15 000 vigas conectadas; es sumamente difícil y tedioso por lo que se hace necesario un método automatizado de generarlo, al igual que el modelado de estructuras más simples; como ferrocarriles, que contienen centenares de extremidades de largos diversos.

La optimización de armaduras se ocupa de la selección de nodos y su conectividad, en otro nivel, crea búsquedas de optimización para encontrar las coordenadas óptimas de nodos existentes. Finalmente, las secciones transversales de miembros de la armadura pueden ser optimizadas, lo cual es conocido como optimización de tamaño. En cada caso, el problema de optimización es subordinado a algunas restricciones en los desplazamientos del nodo, las cargas críticas de alabeo, etc. La función objetivo es usualmente minimizar el peso de estructura que se correlaciona con el costo global.(Rivera, 2015)

Desde hace varias décadas, se han realizado estudios en el campo de optimización estructural. El desarrollo de metodologías eficientes para encontrar estructuras óptimas es muy importante, debido a que facilitaría resolver problemas complejos en áreas como la arquitectura, la ingeniería civil y mecánica.

Elegir la mejor solución y mejorarla, depende fuertemente de la experiencia del diseñador, por lo que no es simple y no se dispone de los parámetros o la capacidad suficientes para asegurar que la última propuesta de solución que se obtenga sea la óptima, ya que es físicamente casi imposible evaluar el espectro completo de posibles soluciones. Entonces, resulta atractivo tener una herramienta que permita obtener soluciones de un problema de diseño lo más cercanas posibles a la óptima, cuyo desempeño no dependa de la experiencia del diseñador.

La optimización hace posible reducir la cantidad de recursos que son destinados a la solución de un problema de ingeniería, facilitando el desarrollo de los procesos y orientándolos en la dirección apropiada, disminuyendo el tiempo y por ende los costos económicos (Cheng, 2010). A lo largo de esta investigación se profundiza en el origen y funcionamiento de estas técnicas, pues existe la **necesidad** de lograr la optimización del diseño de estructuras de forma racional.

Commented [N1]: Situación problemática

La ciencia y la tecnología seguirán evolucionando a medida que pase el tiempo, por lo que los problemas que antes se resolvían gracias a la intuición y experiencia de los investigadores serán resueltos con precisión, exactitud y rapidez a través de las nuevas técnicas que surjan, como es el caso de esta investigación que pretende **lograr** una configuración más racional de las armaduras planas mediante la optimización del diseño de estructuras utilizando técnicas heurísticas (Bellido et al.).

Commented [N2]: Problema científico

Objeto de investigación: Optimización estructural de armaduras planas.

Campo de acción: Diseño de armaduras planas.

A través de la elaboración de una programación, usando técnicas heurísticas de optimización se **pudiera** obtener la configuración más óptima de diferentes tipologías de armaduras planas, resultando estas más ligeras y económicas y a la vez resistentes, por lo que se traza como objetivo general: Elaborar una programación para optimizar el diseño de diferentes tipologías de armaduras planas utilizando técnicas heurísticas y como objetivos específicos:

Commented [N3]: Hipótesis

- Caracterizar las armaduras planas.

- Definir el funcionamiento de técnicas heurísticas de optimización (Algoritmos Genéticos).
- Analizar los diferentes tipos de armaduras para obtener matemáticamente su descripción y la función objetivo.
- Aplicar la técnica heurística seleccionada (Algoritmos Genéticos) en la optimización de la configuración de una armadura plana.
- Analizar las configuraciones obtenidas por cada tipología usando la técnica seleccionada.

Variables independientes: Configuración geométrica de la armadura plana, condiciones de apoyo, técnica de búsqueda.

Variable dependiente: Diseño de una programación de armadura plana.

Como resultado se espera en la investigación, obtener un documento guía para la posterior profundización de esta temática, así como un elemento que sirva a los diseñadores y proyectista para obtener diseños más óptimos.

De los objetivos específicos, se derivan las siguientes tareas principales de la investigación:

- Establecimiento de las bases teóricas y conceptuales:

Caracterización y descripción de los diferentes tipos de armaduras y del funcionamiento de técnicas heurísticas de optimización (Algoritmos Genéticos).

- Análisis y propuestas:

Elaboración de una programación para optimizar el diseño de diferentes tipologías de armaduras planas utilizando técnicas heurísticas.

- Validación de las soluciones dadas al problema.

Aplicación de las técnicas heurísticas en la optimización de la configuración de diferentes tipologías de armaduras planas, resultando estas más ligeras y económicas.

Los valores que destacan de la investigación son:

- **Valor social:**

Adquirir una herramienta que permita obtener soluciones de un problema de diseño lo más cercanas posibles a la óptima, cuya elección, que no es simple, no dependa de la experiencia del diseñador.

- **Valor económico:**

Se pretende lograr una configuración más racional de las armaduras planas mediante la optimización del diseño de estructuras utilizando técnicas heurísticas. El valor económico está representado en la guía de trabajo que permite el diseño de una programación que logre optimizar el diseño de armaduras planas, lo cual disminuirá el costo total del objeto y brindará facilidades de trabajo.

- **Valor metodológico:**

Se realiza un exhaustivo estudio y análisis de técnicas heurísticas para la optimización de armaduras que puede ser utilizado como una base para el estudio y aplicación de los algoritmos genéticos en múltiples campos de la Ingeniería Civil.

- **Valor práctico:**

Con este trabajo de diploma se va profundizando en el origen y funcionamiento de estas técnicas, pues existe la necesidad de lograr la optimización del diseño de estructuras de forma racional. Además de proporcionar una herramienta que facilita y orienta la comprensión del funcionamiento de los algoritmos genéticos en la optimización de estructuras.

Métodos Científicos:

Para desarrollar la presente investigación se emplearán diferentes métodos teóricos, entre los que figuran:

- **Histórico – Lógico:**

Permitirá recorrer los antecedentes hasta llegar a la situación actual de las técnicas heurísticas en la resolución de problemas de optimización.

- **Analítico – Sintético:**

Para el procesamiento de las fuentes de información a fin de determinar los diferentes enfoques y criterios relacionados con la resolución de problemas de optimización, estableciendo puntos de concatenación entre la presente indagación y materiales anteriores en cuanto a enfoque, visión y perspectiva.

- **Inductivo – Deductivo:**

Aplicado por la autora para generalizar los rasgos más importantes obtenidos a partir de la documentación para definir los modelos y analizar las investigaciones ingenieras.

Métodos Empíricos:

- **Revisión de documentos.**

Para el desarrollo de la investigación se consultaron diferentes fuentes tanto en formato digital como impreso en diferentes idiomas.

El trabajo de diploma queda estructurado de la manera siguiente: Resumen/ *Abstract*, Introducción, Capítulo1, Capítulo2, Capítulo3, Conclusiones, Recomendaciones, Bibliografía y Anexos.

Para una visión global del campo de optimización, esta tesis comienza cubriendo estos contornos antes mencionados en los subsiguientes capítulos:

Capítulo1: Marco Teórico Referencial

Descripción, conceptualización, utilización y principales tipologías de las armaduras planas y principales ventajas de las técnicas heurísticas de optimización sobre las tradicionales.

Capítulo 2: Análisis de las diferentes tipologías de armaduras y funcionamiento de los Algoritmos Genéticos.

Descripción de las diferentes tipologías a utilizar que serán las que están en el *SAP 2000* en las plantillas de armaduras 2D (*Pratt, English, Vertical*), usando el método de los nudos y las secciones. Análisis de una de las técnicas heurísticas (Algoritmos Genéticos) para optimizar diferentes tipologías estructurales de armaduras.

Capítulo 3: Caso de estudio

Finalmente se analizarán los resultados obtenidos en cada caso de estudio con la aplicación de los Algoritmos Genéticos con las programaciones establecidas.

Resultados Esperados: Al resolver o diseñar una armadura con la programación elaborada empleando Algoritmos Genéticos se obtenga una solución óptima en un plazo de tiempo razonable y que satisfaga las condiciones establecidas sin depender de la experiencia e intuición del diseñador.

CAPÍTULO 1 MARCO TEÓRICO REFERENCIAL.

En este capítulo se realiza una descripción, conceptualización y utilización de las principales tipologías de las armaduras planas, además de una breve reseña de los avances que ha tenido la optimización estructural (Algoritmos Genéticos) en los últimos tiempos y las principales ventajas de las técnicas heurísticas de optimización sobre las tradicionales, así como la definición de los conceptos fundamentales que se deben conocer del amplio campo referente a la optimización.

1.1 Optimización

El hombre desde sus inicios se ha enfrentado constantemente a situaciones que requieren de la selección de una de varias alternativas, con el fin de dar solución a un problema o suplir alguna necesidad. Hasta hace poco tiempo, bastaba con que la alternativa escogida fuera útil para lo que se requería, sin embargo los recursos disponibles son limitados, entonces no basta con elegir una alternativa solo porque esta sirve, sino que se hace necesario elegir la “mejor” de todas las posibilidades, es decir, la requiere de menos recursos para obtener el mismo resultado.(Chen, 2013)

De este modo, surge el concepto de optimización, que se refiere a la selección del mejor elemento de un conjunto de elementos posibles, de acuerdo a un criterio de selección dado. Para esto, se deben emplear métodos de optimización que permitan conocer los mejores valores que puede asumir dicho conjunto de variables.(Ramos et al., 2010)

De forma genérica, puede definirse la optimización como aquella ciencia encargada de determinar las mejores soluciones a problemas matemáticos que a menudo modelan una realidad física.(Coello et al., 2000, LÓPEZ, 2010)

Una solución eficiente y precisa a este problema depende no sólo del tamaño del problema en términos del número de restricciones y del número de parámetros de diseño sino también de las características de la función objetivo y de las restricciones.

El problema de la optimización en estructuras se puede abordar desde diferentes enfoques, que se dividen fundamentalmente los siguientes:(Ramírez, 2014)

- **Optimización del tamaño:**Trata de determinar las medidas óptimas de los elementos que constituyen una estructura de forma predefinida. Es el enfoque más sencillo ya que el número de variables del problema es bastante limitado.

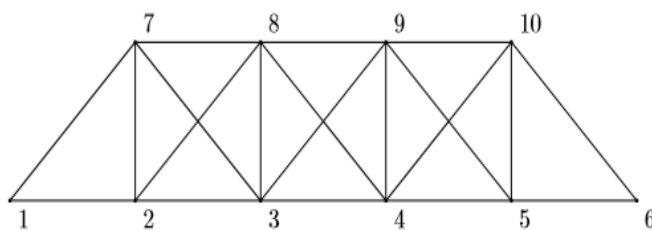


Figura 1.1 Optimización de tamaño(Ramírez, 2014)

- **Optimización de la forma:**Trata de buscar la forma óptima de una estructura de topología fija. En este enfoque, partiendo de una forma predefinida se busca optimizar esta mediante la modificación de la conectividad entre elementos o mediante la eliminación de los elementos menos esforzados.

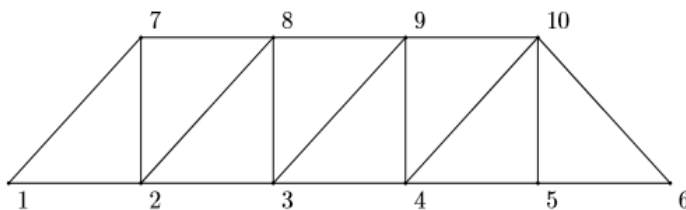


Figura 1.2 Optimización de forma (Ramírez, 2014)

- **Optimización de la topología:**Trata de buscar la distribución óptima del material en una estructura, mediante la supresión de elementos y/o la modificación de la conectividad y/o de las coordenadas nodales. Este enfoque es el más complicado

porque la cantidad de variables que aborda es elevada. En todos los trabajos publicados hasta la fecha se parte siempre de una morfología previa.

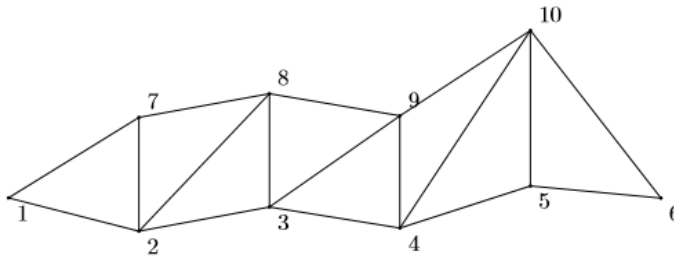


Figura 1.3 Optimización de topología (Ramírez, 2014)

1.1.1 Función objetivo

La función objetivo, es la función que mide cualitativamente el funcionamiento del sistema en un proceso de optimización, de modo que lo que se busca es la maximización o minimización de esta, enmarcado en una gama de posibles soluciones.

Es la medida cuantitativa del rendimiento del sistema a optimizar (maximizar o minimizar). Dos aspectos que resultan cruciales en el comportamiento de los Algoritmos Genéticos son la determinación de una adecuada función de adaptación o función objetivo, así como la codificación utilizada.(Alcazar, 2016)

Idealmente es de gran interés construir funciones objetivo con "ciertas regularidades", es decir funciones objetivo que verifiquen que para dos individuos que se encuentren cercanos en el espacio de búsqueda, sus respectivos valores en las funciones objetivo sean similares. Por otra parte una dificultad en el comportamiento del Algoritmo Genético puede ser la existencia de gran cantidad de óptimos locales, así como el hecho de que el óptimo global se encuentre muy aislado.(Bonelli, 2016)

La regla general para construir una buena función objetivo es que ésta debe reflejar el valor del individuo de una manera "real", pero en muchos problemas de optimización

combinatoria, donde existe gran cantidad de restricciones, buena parte de los puntos del espacio de búsqueda representan individuos no válidos.

Otra técnica que se ha utilizado en el caso en que la computación de la función objetivo sea muy compleja es la denominada evaluación aproximada de la función objetivo. En algunos casos la obtención de "n" funciones objetivo aproximadas puede resultar mejor que la evaluación exacta de una única función objetivo.

Un problema habitual en las ejecuciones de los Algoritmos Genéticos surge debido a la velocidad con la que el algoritmo converge. En algunos casos la convergencia es muy rápida, lo que suele denominarse convergencia prematura, en la cual el algoritmo converge hacia óptimos locales, mientras que en otros casos el problema es justo el contrario, es decir se produce una convergencia lenta del algoritmo. Una posible solución a estos problemas pasa por efectuar transformaciones en la función objetivo. El problema de la convergencia prematura, surge a menudo cuando la selección de individuos se realiza de manera proporcional a su función objetivo. En tal caso, pueden existir individuos con una adaptación al problema muy superior al resto, que a medida que avanza el algoritmo "dominan" a la población. Por medio de una transformación de la función objetivo, en este caso una comprensión del rango de variación de la función objetivo, se pretende que dichos "superindividuos" no lleguen a dominar a la población.

El problema de la lenta convergencia del algoritmo, se resolvería de manera análoga, pero en este caso efectuando una expansión del rango de la función objetivo.(Bonelli, 2016)

Asimismo, existen diferentes tipos de funciones objetivo, de acuerdo al tipo de variable, formulación matemática y operatividad. Entre las más importantes:

- Lineal y No Lineal.
- Unimodal y Multimodal.
- Estática y Dinámica.

- Sin Ruido y Con Ruido.
- Un objetivo y Multiobjetivo.

Sea cual sea el método y función de optimización escogido, este tiene que tener la capacidad de representar la totalidad de las soluciones, debe existir una ruta entre dos conexiones en el espacio y además la representación debe ser fácilmente operada, es decir, el método debe ser eficiente.(Ceferino, 2008)

1.1.2 Restricción

Una restricción es un parámetro que permite ceñir, circunscribir o reducir a un límite un elemento determinado, de modo que haya validez en el diseño. (Cerroloza y Annicchiarico, 1996).

Dado que estas restricciones son aplicadas en problemas de tipo físico y matemático, se pueden clasificar en dos grandes grupos:

- Funcionales: Se refieren al comportamiento físico-mecánico del sistema.
- Geométricas: Se refieren a las dimensiones y tipología de los elementos.

En el ámbito estructural, las restricciones se refieren al comportamiento de la estructura y su estabilidad, y se imponen generalmente a los esfuerzos y deformaciones en los elementos y por ende a sus dimensiones mínimas y máximas. Además, deben imponerse restricciones constructivas para que los elementos obtenidos tengan una secuencia y configuración lógica y factible de ser llevada a cabo en la realidad.

1.1.3 Espacio de búsqueda

La representación de una solución potencial y su correspondiente interpretación brinda el espacio de búsqueda y su tamaño. Este es un punto clave para el problema: el tamaño de dicho espacio no viene determinado por el problema sino por la representación e interpretación. Este paso es de vital importancia para dar una correcta solución al problema, además de evitar contratiempos de duplicaciones de resultados, etc.

Se define como espacio de búsqueda al conjunto de todos los elementos que pueden ser operados por los métodos de optimización. La búsqueda generalmente se realiza con los mejores individuos de distintas áreas del espacio del problema (LÓPEZ, 2010)

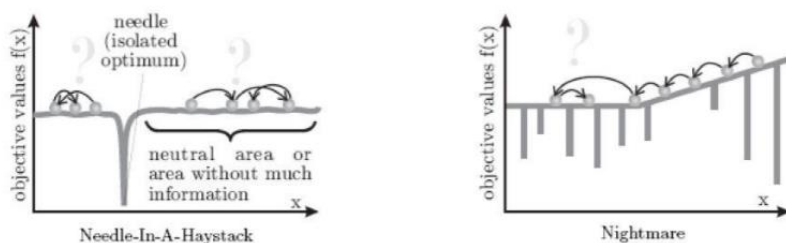


Figura 1.4 Ejemplo de espacios de búsqueda particulares donde los algoritmos genéticos no funcionan correctamente (Thomas, 2009).

1.1.4 Variables

Representan las decisiones que se pueden tomar para variar el valor de la función objetivo. Se pueden clasificar como dependientes o independientes.

1.1.5 Óptimo local, Óptimo global y de Pareto

Como definición básica, en matemáticas aplicadas y ciencia computacional, un óptimo local de un problema de optimización es una solución óptima (un máximo o un mínimo) de un conjunto vecino de soluciones candidatas. En contraposición, un óptimo global es la solución óptima de entre todas las soluciones posibles. En relación a estos dos tipos de óptimos, se pueden distinguir entre algoritmos locales y globales.

Los algoritmos de optimización local, en un caso general, sólo son capaces de detectar óptimos locales de las funciones objetivo. Se trata de algoritmos que dirigen su búsqueda evaluando la derivada de la función objetivo o alguna aproximación de ésta.(Alcazar, 2016)

Por su parte los algoritmos de optimización global permitirían detectar óptimos globales, o al menos una buena aproximación de los mismos. La mayoría de los métodos metaheurísticos realizan una búsqueda estocástica por todo el espacio solución, de modo que estarían en condiciones de detectar óptimos globales de las funciones objetivo.

Según (LÓPEZ, 2010) las partículas que se mueven por el espacio de búsqueda atraídas por la posición de mayor aptitud lograda son denominadas óptimo local, así como la de mejor aptitud lograda en todo el enjambre se conoce como óptimo global, cada una de estas puede representar una posible solución del problema.

Óptimo de Pareto: Este término aparece cuando se afronta problemas de optimización de más de una variable y con restricciones. En estos casos no todas las soluciones son viables. Y más aún, los óptimos de Pareto hacen referencia a soluciones en las que no se puede mejorar más los valores de una variable, sin empeorar alguna de las otras condiciones. Es decir que más que alcanzar una solución única, lo que se consigue es definir una frontera de soluciones equilibradas (frente de Pareto) donde no es posible encontrar una solución que mejore en ningún sentido sin empeorar en otro (Coello et al., 2007).

1.2 Antecedentes de los algoritmos evolutivos genéticos.

Los Algoritmos Evolutivos incluyen a todas las técnicas relacionadas con la optimización metaheurística, estas surgieron por el año 1960. Con el auge que tomaron estas teorías, y por medio de los descubrimientos de nuevas tecnologías que permitían un desarrollo computacional y matemático mucho más elevado, empezaron a investigarse teorías y algoritmos artificiales sentando sus bases en los mecanismos naturales con los trabajos de Holland (FRASER, 1958 1957, Holland, 1975).

A partir de esta fecha, los estudios sobre algoritmos evolutivos, en cualquiera de sus facetas, se han incrementado de tal forma que ya es difícil encontrar un área de investigación en la que no hayan irrumpido con gran fuerza. Se han encontrado implementados en ámbitos tan diversos como finanzas, inversión, educación, transporte, redes de comunicación, planificación, logística, producción, optimización, etc.(Biethan

and Nissen, 1995). Las primeras ideas en las que se basan los algoritmos genéticos se pueden encontrar en los artículos de Holland de principios de los años 1960. En ellos se establecen los aspectos básicos para la comprensión de los principios de los sistemas adaptativos. Estos sistemas son capaces de automodificarse en respuesta a su interacción con el medio en el que están funcionando. Las teorías sobre los sistemas adaptativos están orientadas a facilitar, por un lado, la comprensión de las formas complejas de adaptación que aparecen en los sistemas naturales y, por otro, la habilidad para diseñar sistemas adaptativos robustos. A mediados de los años 1960, las ideas de Holland empezaron a plasmarse en modelos implementados en ordenador. En cada uno de estos sistemas se representaban elementos por medio del uso de genomas en los que los mecanismos de evolución y herencia eran abstracciones de operadores genéticos como la mutación, el cruce o la inversión.

Estos estudios experimentales desembocaron en el tratamiento de problemas de búsqueda más complejos, como el reconocimiento de patrones. Asimismo, se realizaron los primeros estudios con operadores de selección de tipo elitista y las ideas básicas para el empleo de probabilidades adaptativas de mutación y cruce. También en esta época se estudió detalladamente, por primera vez, las diferentes posibilidades de reproducción y cruce. Usando dos espacios de soluciones diferentes, se experimentó con una amplia variedad de operadores de cruce.

En paralelo con estos estudios experimentales, Holland continuó trabajando en una teoría general de los sistemas adaptativos. Posteriormente evidenció todas estas ideas en su libro *“Adaptation in Natural and Artificial Systems”*. Muchas de las propiedades de los algoritmos genéticos identificadas por Holland de forma teórica no pudieron ser observadas experimentalmente debido a la falta de recursos computacionales. Esto obligaba a trabajar con un número limitado de generaciones y con un número muy pequeño de individuos, habitualmente menos de 20 (Holland, 1975).

Ya por estos años había un gran interés en entender mejor el comportamiento de los algoritmos genéticos. Se sabía en concreto que la elección del tamaño de la población, y la elección de los operadores tenían efectos relevantes en el funcionamiento de los

algoritmos genéticos. Así se realizaron estudios en detalle sobre la influencia del cruce y la mutación, y se empezaron a utilizar los operadores de cruce basados en múltiples puntos. Estos estudios fueron ampliados posteriormente de forma teórico-práctica con el análisis de los efectos cruzados de la modificación del tamaño de la población, cruce y mutación en el comportamiento de una familia de algoritmos genéticos utilizados para optimizar un conjunto fijo de funciones de prueba. A partir de estos estudios se empezó a manifestar el enorme potencial de los algoritmos genéticos para resolver problemas de optimización (Jong, 1975, Coello, 1999).

En esta misma época, se empieza a generalizar la investigación sobre los algoritmos genéticos al iniciarse trabajos en diversas universidades. Esta generalización fue muy lenta en un principio debido a cierto escepticismo creado por los malos resultados que habían obtenido las técnicas basadas en sistemas auto-organizativos y las redes de neuronas artificiales con los perceptrones que no cubrieron las expectativas creadas. Entorno a estas técnicas se creó un ambiente de pesimismo pero a pesar de esto se celebró en la Universidad de Michigan el congreso "*An interdisciplinary Workshop in Adaptive Systems*", lo cual impulsó su estudio, consiguiendo que se establecieran diversos grupos de trabajo. Así, se empezaron a estudiar los sistemas clasificadores propuestos por Holland, nuevas mejoras de los algoritmos genéticos como la regla de ponderación para aprendizaje, y también, el estudio de nuevas aplicaciones de optimización.

Dado el creciente interés en los algoritmos genéticos, se organizó en 1985 el primer "*International Congress on Genetic Algorithms*" donde se presentaron los últimos avances tanto teóricos como prácticos en el empleo de estas técnicas. El éxito de esta edición del congreso consiguió que se repitiera su celebración cada dos años. A partir de 1989, las actividades en el campo de los algoritmos genéticos habían crecido mucho, por lo que se fundó la "*International Society for Genetic Algorithms (ISGA)*" (Grefenstette, 1986, Grefenstette and Baker, 1989).

Fue precisamente David Goldberg, un discípulo de Holland quien alcanzó mayor éxito en la temática de optimización. Según parece tras asistir a uno de los seminarios de Holland, Goldberg que estaba interesado en encontrar el diseño óptimo de líneas para el transporte

de gas, le planteo la posibilidad de emplear los Algoritmos Genéticos con este fin. Poco tiempo después logro llevar a cabo dicha tarea en su tesis doctoral. Posteriormente publicaría un libro que se convertiría un hito en el campo de la optimización y que recogía numerosas aplicaciones, muchas de ellas relacionadas con la optimización de estructuras, como el clásico problema de optimización de diez barras y seis nudos, aunque sin la restricción de desplazamientos del problema original (Goldberg, 1989)

En 1982, se intentó optimizar el diseño de una estructura sometida a cargas dinámicas mediante una búsqueda aleatoria. Aunque el algoritmo empleado no fue un Algoritmo Genético, este trabajo representa el primer intento por resolver un problema de optimización de estructuras mediante una técnica heurística. En 1990 se estudió la aplicabilidad de los Algoritmos Genéticos como método de optimización de estructuras.

Posteriormente en 1992 se empleó un Algoritmo Genético binario para optimizar una estructura de tres barras y la clásica estructura de diez barras pero añadiendo las restricciones de desplazamiento del problema original. También aplicaron el algoritmo para optimizar una torre de transmisión de 160 barras.

Ese mismo año se utilizó un Algoritmo Genético binario que empleaba variables discretas para minimizar una estructura de diez barras sujeta a restricciones de desplazamiento. Al año siguiente utilizaron un Algoritmo Genético con dos estrategias de cruce diferentes para optimizar estructuras a gran escala, aplicando el algoritmo a estructuras de 25 y 72 barras. En este mismo año se implementó un Algoritmo Genético binario para optimizar el tamaño y la topología de diversos pórticos. Este es uno de los primeros trabajos en analizar estructuras con elementos de viga. En 1994 el aporte se manifestó en la aplicación de un algoritmo a estructuras tridimensionales (Alcazar, 2016)

1.3 Algoritmos genéticos

Los algoritmos genéticos son métodos metaheurísticos de optimización basados en un planteamiento probabilístico que además realizan la búsqueda por poblaciones. Su importancia radica en que es un método capaz de converger con muy alta probabilidad en un óptimo global, a diferencia de los métodos basados en gradientes que eran los que más

se utilizaban hasta la aparición de estos nuevos algoritmos. Su patrón de búsqueda trata de reproducir los principios de la evolución biológica, es decir, intenta imitar el mecanismo de la teoría de la evolución de Darwin en la que los miembros de una población compiten para sobrevivir y se reproducen, mientras los más débiles mueren. De esta forma, en los algoritmos genéticos se producen nuevas soluciones más evolucionadas (que optimizan la función objetivo del problema) en cada iteración gracias a las operaciones de selección y combinación, de forma análoga a como los genes se transfieren a los nuevos individuos.

Un algoritmo es un conjunto ordenado y finito de operaciones que permite ejecutar un procedimiento o resolver un problema, un algoritmo es genético cuando utiliza principios de la genética y la selección natural.(Castillo)

Los algoritmos genéticos trabajan sobre estructuras de cadena que son una representación de una posible solución al problema denominado cromosomas para comenzar, se inicializa la población completamente al azar. En la inicialización hay que tener en cuenta que la distribución de valores debe ser uniforme para cada rango representado por los cromosomas, estos evolucionan a través del tiempo de acuerdo a reglas de supervivencia que privilegian a los que presenten mejor desempeño dándoles una mayor probabilidad de que a partir de ellos se cree la siguiente generación de acuerdo con la función de adaptación; del mismo modo que en la naturaleza los individuos mejor dotados (es decir, los que ofrezcan las mejores soluciones) serán los que más probabilidades tendrán de ser seleccionados para reproducirse. Mientras que los individuos peor adaptados (los que ofrezcan peores soluciones) tendrán más difícil el propagar su material genético a las nuevas generaciones. Sucesivamente cada nueva generación contará con una mayor proporción de buenas características de forma que, si el algoritmo genético ha sido diseñado correctamente, la población convergerá hacia una solución óptima del problema. Mediante un proceso fuertemente estocástico se genera una nueva población de individuos tomando en cuenta su *fitness*. Básicamente durante la selección se decide cuales individuos serán padres de la nueva generación. La nueva población puede reemplazar completamente a la población anterior o solamente a las peores soluciones.(FRASER, 1958)

En algunos casos para que un elemento de la población pueda pasar a través de la función de aptitud debe ser codificado, este proceso de codificación es clave para permitir que el algoritmo converja de manera más eficiente hacia la solución del problema planteado, en principio la codificación puede realizarse con cualquier alfabeto finito, sin embargo, no todas las codificaciones resultan ser computacionalmente eficientes, es deber del programador dependiendo de la plataforma de *software* o *hardware* que elija escoger que codificación resulte más conveniente.

Luego de haber decidido que codificación usar una serie inicial de cadenas (población) es creada aleatoriamente, entonces se aplican operadores a la población para crear nuevas generaciones que se espera tengan mejor desempeño. Estas candidatas prometedoras se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, pero las copias no son perfectas; se introducen cambios aleatorios durante el proceso de copia.

Al volver al principio del ciclo evolutivo se verifica nuevamente si el mejor individuo supera los requisitos de la búsqueda, en caso contrario se repite todo el proceso para obtener una nueva generación. Este proceso se repetirá en tanto que no se cumpla la condición de parada impuesta previamente. (Coello et al., 2007, Goldberg, 1989)

Por lo tanto, un algoritmo genético consiste en hallar de qué parámetros depende el problema, codificarlos en un cromosoma, se aplican los métodos de la evolución: selección y reproducción sexual con intercambio de información y alteraciones que generan diversidad.

Según (Ison et al., 2005, Vidal, 2013), los Algoritmos genéticos tienen capacidades importantes en la rama de la optimización, donde se puede destacar las siguientes:

- Son métodos iterativos que convergen a partir de un conjunto inicial y arbitrario de puntos del espacio de búsqueda.
- La convergencia no está condicionada por requisitos de continuidad o diferenciabilidad del funcional a maximizar o minimizar.

- En su convergencia al óptimo global escapan de la atracción de múltiples óptimos locales.
- Son aplicables y eficientes en la optimización multicriterio.

Estas capacidades en la última década han obtenido nuevas soluciones en nuevos problemas complejos de optimización, caso de problemas de diseño multidisciplinarios de ingeniería entre otros, reduciendo con estos métodos ciclos de diseño y costes de nuevos productos, y permitiendo resolver eficientemente diversos problemas multicriterio, donde objetivos simultáneos tanto económicos como técnicos están en conflicto.

Por lo que se puede decir que el algoritmo genético comienza con una población inicial de datos, estos datos representan posibles resultados, que serían los cromosomas, los datos contienen *bits*, llamados genes. Los datos obtenidos, pasan a la fase de evaluación, donde se les aplica una función de aptitud, para saber qué tan eficaz es la solución que esta codificada. Al final de la fase cada cromosoma va a recibir una calificación de aptitud. Esta calificación, es importante, porque en base a esto se procede a elegir los cromosomas que serán cruzados en la siguiente generación, ya que una calificación alta, significa que es una solución eficaz para el problema que se quiere resolver y aquellos que reprueban son eliminados. Una vez obtenidos los cromosomas ganadores, comienza el cruce o recombinación de datos y la mutación de datos. El cruce de datos está dictado por diferentes técnicas, que dependen del programador, pero todas se basan en el corte y la combinación de datos en una secuencia de *bits*, mientras que la mutación se basa en solo cambiar un *bit* en la cadena de datos, esta mutación se puede dar al final del cruce de datos, o antes del cruce de datos, o sea, después de la evolución de aptitud. (Ramírez, 2014 .32)

La mutación es un mecanismo necesario para evitar que el algoritmo se quede atorado en alguna solución ineficaz o sin sentido. Terminada la fase de mutación y recombinación de datos, estos son evaluados de nuevo con la función de aptitud para saber qué tan eficaz es la nueva serie de datos que tenemos. Este proceso se repite "n" veces y termina dependiendo de:

- Si encuentra una solución satisfactoria.
- Si alcanza el número fijado de iteraciones.
- Si se rompe el algoritmo.
- Si es detenido manualmente.

Este método, tiene sus ventajas, desventajas y limitaciones. Posee la habilidad de manipular muchos parámetros simultáneamente, lo que permite explorar muchas soluciones, así que es posible evaluar implícitamente muchos esquemas a la vez y funciona particularmente bien resolviendo problemas donde existen muchas soluciones. Por otra parte, se tiene que tener bien definido el problema, se debe utilizar un lenguaje de programación que sea capaz de tolerar cambios aleatorios, que no produzca constantemente errores fatales o resultados sin sentido. La función de aptitud debe considerarse cuidadosamente para que se pueda alcanzar una evaluación real que verdaderamente proporcione una solución para mejorar el problema dado, de lo contrario, cada iteración arrojará a una solución que no significa nada o una solución falsa.

Las principales ventajas de los algoritmos genéticos respecto a los algoritmos clásicos y los demás métodos heurísticos son las siguientes (Goldberg, 1989):

- Trabajan con una codificación de un conjunto de parámetros, no con los parámetros mismos.
- Trabajan con un conjunto de puntos, no con un único punto y su entorno (su técnica de búsqueda es global.) Utilizan un subconjunto del espacio total, para obtener información sobre el universo de búsqueda, a través de las evaluaciones de la función a optimizar. Esas evaluaciones se emplean de forma eficiente para clasificar los subconjuntos de acuerdo con su idoneidad.
- No necesitan conocimientos específicos sobre el problema a resolver; es decir, no necesariamente están sujetos a restricciones. Por ejemplo, se pueden aplicar a

funciones no continuas, lo cual les abre un amplio campo de aplicaciones que no podrían ser tratadas por los métodos tradicionales.

- Utilizan operadores probabilísticos, en vez de los típicos operadores determinísticos de las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo.
- Cuando se usan para problemas de optimización, resultan menos afectados por los máximos locales que las técnicas tradicionales (son métodos robustos).

Para resolver un problema de optimización empleando algoritmos genéticos, deben definirse los siguientes parámetros:

1.3.1 Población

Este parámetro indica el número de cromosomas que se tiene para una generación determinada. Es importante que exista una población lo suficientemente amplia, para que haya un adecuado número de reproducciones y se llegue a una solución óptima. No obstante, el tamaño de la población debe definirse de modo que no sea demasiado amplio y por lo tanto la convergencia tome mucho tiempo. Por lo general, la población inicial se genera como una combinación aleatoria de individuos a partir de un rango de variables de diseño.

1.3.2 Función de evaluación (*Fitness*)

Esta función es la que vincula al AG con el problema de optimización y es la que evalúa la calidad de la solución. Comienza tomando un cromosoma de entrada y retorna un número o conjunto de números que dan una medida de la calidad del cromosoma en relación a la condiciones del problema.

Según Jong (1975), cuyo trabajo se centra en la escogencia de funciones de optimización, se pueden distinguir los siguientes tipos de funciones para el caso de los algoritmos genéticos:

- Continuas / Discontinuas.
- Convexas / No Convexas.
- Unimodales / Multimodales.
- Cuadráticas / No Cuadráticas.
- Dimensionalidad Alta / Dimensionalidad Baja.
- Determinísticas / Estocásticas.

1.3.3 Operadores Genéticos

Se aplican de modo que se altere la composición de los descendientes (mutación o cruce). Análogamente a la teoría de evolución de las especies de Darwin, se distinguen tres operaciones genéticas que se deben aplicar a los individuos para evaluar una población.

1.3.4 Selección

Es el operador que selecciona un grupo de individuos de la población actual para el apareamiento, dependiendo del valor de su *fitness*. (Saka and Geem, 2012)

1.3.5 Cruzamiento o Reproducción

Esta es la segunda operación genética, en la cual los *strings* de los padres seleccionados son fraccionados y algunos genes son intercambiados para generar nuevos individuos.

1.3.6 Mutación

En la evolución, una mutación es un suceso bastante poco común. Estas contribuyen a la diversidad genética de la especie. En un algoritmo genético tendrán el mismo papel, y la misma frecuencia es decir, muy baja.

Una vez aplicado el cruzamiento para obtener una nueva población se aplica este tercer operador genético, la mutación, que consiste en alterar aleatoriamente los genes de algunos de los individuos de la población. Si no hay mutación, los descendientes son los

mismos que había cuando se dio la reproducción. En caso de que haya mutaciones, parte del cromosoma descendiente es modificado y si la probabilidad de mutación es del 100%, significa que la totalidad del cromosoma es alterada.

1.3.7 Elitismo

Este operador se emplea para que las operaciones de cruce y mutación no afecten un buen cromosoma indefinidamente. Lo que hace es copiar el mejor individuo de la población anterior a la siguiente, lo que permite una mejor convergencia del algoritmo. Sin embargo, deben imponerse las restricciones necesarias para que no se generen “superindividuos”.

A continuación se muestra el diagrama de flujo de un algoritmo genético simple, con la secuencia cronológica de las operaciones genéticas descritas anteriormente.

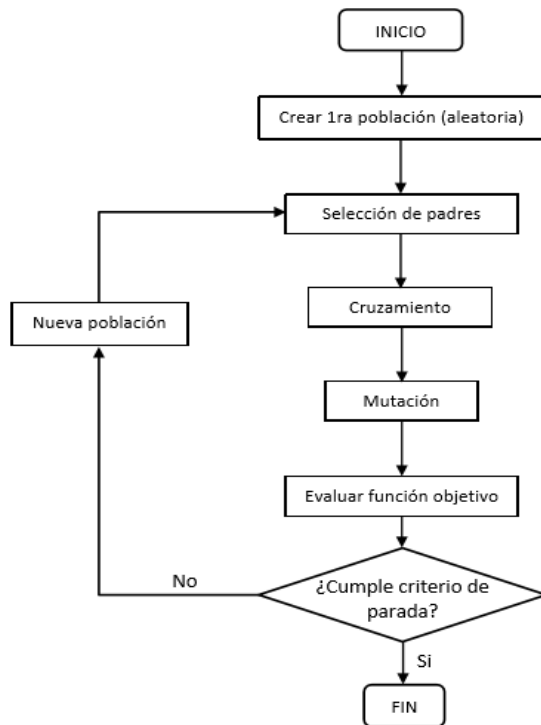


Figura 1.5 Diagrama de Flujo de Algoritmo Genético Simple.(Ramírez, 2014)

1.3.8 Ventajas, desventajas y limitaciones de los algoritmos genéticos.

Ventajas:

- No necesitan conocimientos específicos del problema a resolver.
- Operan de forma simultánea con varias soluciones y no de forma secuencial como las técnicas tradicionales.
- Son fáciles de ejecutar en modernas arquitecturas masivamente paralelas.
- Usan operadores probabilísticos, en vez de los típicos operadores determinístico de las otras operaciones.
- Trabajan con un código del conjunto de parámetros, no con el conjunto mismo. Por trabajar a nivel de código, y no con las funciones y sus variables de control, como los otros métodos, es más difícil que la solución converja a un mínimo o máximo local.
- Buscan una población de puntos, no un único punto. Manteniendo una población de puntos muestrales bien adaptados, se reduce la probabilidad de caer en una solución falsa.
- Emplean la función objetivo, no necesitan derivadas ni otra información complementaria, tan difícil a veces de conseguir. De este modo ganan en eficiencia y en generalidad.
- Cuando se usan para problemas de optimización resultan menos afectados por los máximos o mínimos locales que las técnicas tradicionales.
- Se valen de reglas de transición estocástica, no deterministas. Los Algoritmos Genéticos se valen de operadores aleatorios para guiar la búsqueda de los mejores puntos.

- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en gran medida de los parámetros que se utilicen: tamaño de la población, número de generaciones y porcentajes de los operadores genéticos (Coello et al., 2007, Goldberg, 1989, Holland, 1975, Jong, 1975, Jenkins, 2002, Thierens and Goldberg, 1994, Alander, 1992, Wang and Arora, 2004, Fraser, 1958, Thomas, 2009, Fogel, 1967).

Desventajas:

- Pueden tardar mucho en converger o no converger en absoluto, dependiendo en cierta medida de los parámetros que utilicen tamaño de la población, número de generaciones, etc.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole. (Coello et al., 2007, FRASER, 1958, Goldberg, 1989, Holland, 1975, Marler and Arora, 2004)

Limitaciones:

- El uso de los Algoritmos Genéticos no garantiza que encuentre una solución óptima del problema, existe evidencia de que encuentra soluciones de un nivel aceptable, en un tiempo competitivo con el resto de las demás técnicas de optimización, incluso pueden existir técnicas que sean más rápidas y eficaces que el mismo Algoritmo Genético, pese a todas esas limitaciones; este tipo de algoritmos es una gran herramienta que siguen siendo la opción más aceptable al resolver problemas (Coello et al., 2007, FRASER, 1958, Goldberg, 1989, Holland, 1975, Marler and Arora, 2004).

1.4 Técnicas heurísticas de optimización.

Al resolver una armadura con las técnicas actuales, se diseña y se halla una solución que satisfaga las condiciones establecidas por el método de prueba y error, dependiendo en

gran medida de la experiencia de quien lo calcule. El resultado es una estructura factible desde el punto de vista estructural y de las normas vigentes, pero no necesariamente optimizada.

Los algoritmos heurísticos por lo general tienen un tiempo de ejecución mucho menor y evidencian la posibilidad de mejorar significativamente el diseño de las estructuras, a partir de una línea de desarrollo que se impone en el mundo, pero que en Cuba aún se encuentra poco difundida y estudiada.(Vidal, 2013)

Uno de los inconvenientes de algoritmos exactos es que pueden ser muy lentos. En este caso se pueden utilizar algoritmos aproximados o, si son todavía demasiado lentos, los algoritmos heurísticos. La metodología es muy diferente entre los algoritmos heurísticos y algoritmos exactos y aproximados.

Dentro del grupo de los algoritmos heurísticos se pueden distinguir los denominados métodos metaheurísticos, que imitan fenómenos simples observados en la naturaleza y que parecen estar asociados con la inteligencia artificial. Estos algoritmos tratan de adaptar el comportamiento de diferentes especies a soluciones de problemas altamente complejos mediante optimización. Entre otros, se destacan los siguientes:

- Algoritmos evolutivos (genéticos): basado en modelos biológicos que emulan el proceso natural de evolución.
- Algoritmos basados en el comportamiento de las comunidades.
- Búsqueda heurística (tabú, aleatorios...)

En general, los métodos tradicionales se encargan de buscar y garantizar un óptimo local mientras que los métodos metaheurísticos tienen mecanismos específicos para alcanzar un óptimo global, pero no garantizan ese alcance.(Chen, 2013)

Entre las numerosas técnicas evolucionarias existentes, se han seleccionado los Algoritmos Genéticos debido al hecho de que teórica y experimentalmente se ha demostrado que proveen una búsqueda robusta en espacios complejos de diseño. Estos

algoritmos tienen muchas ventajas sobre los métodos tradicionales de búsqueda. Entre otras consideraciones, no necesitan de información especializada de la función a optimizar (continuidad, gradientes) si no los valores de la función objetivo o de mérito. Esta información es usada para determinar el éxito o fracaso de un diseño en un ambiente en particular. Más aún su habilidad para encontrar la solución óptima o casi-óptima le confiere una posición privilegiada como una herramienta poderosa en problemas no convencionales de optimización.

1.4.1 Ventajas sobre las técnicas tradicionales.

Los métodos de optimización (global) se pueden clasificar en cuatro categorías, en función del grado de exactitud del óptimo que logran alcanzar.

1. Incompletos: Algoritmos que usan la heurística en la búsqueda del óptimo, pero que pueden quedarse atascados en un mínimo (o máximo) local.
2. Asintóticamente completos: Métodos que alcanzan un mínimo (o máximo) global con certeza o probabilidad igual a la unidad si se les permite desarrollarse indefinidamente, pero que no disponen de medios para saber cuándo se ha encontrado dicho óptimo.
3. Completos: También alcanzan un óptimo global con certeza si se les permite desarrollarse indefinidamente, y además aseguran dentro de unas tolerancias prescritas un óptimo global aproximado tras un tiempo de ejecución finito.
4. Riguroso: Métodos que alcanzan un mínimo global con certeza dentro de unas tolerancias dadas, incluso en presencia de errores de redondeo, salvo casos degenerados en los que se puedan exceder los límites de la tolerancia.

Los métodos tradicionales o deterministas pertenecen a las dos últimas categorías; es decir, encuentran un óptimo global en un tiempo limitado. Por ello, estos métodos se usan cuando es de absoluta necesidad encontrar el óptimo global de un problema o cuando se pretende obtener la mejor solución posible.

Estos métodos requieren de una definición analítica del problema y es necesario que se definan de forma rigurosa las relaciones de dependencia (condiciones de contorno) de la

función objetivo en las regiones del espacio. De hecho se puede decir que la principal diferencia entre los métodos tradicionales y los no tradicionales es que los primeros realizan cálculos sobre las regiones del espacio de soluciones mientras los segundos realizan cálculos en puntos individuales. Además no contienen instrucciones que utilicen números aleatorios para decidir de qué forma proceder o cómo modificar los datos.(Ceferino, 2008)

Como se ha comentado anteriormente, cuando la relación entre los candidatos a solución y la función objetivo es muy compleja o la dimensión del espacio solución es muy grande, resolver mediante técnicas tradicionales el problema, se vuelve una tarea muy complicada o incluso irresoluble en algunos casos. Es entonces cuando entran en juego las técnicas de optimización metaheurísticas.

Las técnicas metaheurísticas, a diferencia de las técnicas de optimización tradicionales, no siguen unos métodos o reglas preestablecidas de búsqueda. El principal inconveniente de este tipo técnicas es que no garantizan la localización del óptimo absoluto, por lo que para tener la certeza de haberlo obtenido (o al menos un punto muy próximo) debe ejecutarse el algoritmo varias veces.

Los algoritmos metaheurísticos, del griego "meta" ("más allá") y "heurístico" ("encontrar"), son métodos desarrollados para resolver problemas muy generales de forma más rápida que con los métodos tradicionales o para encontrar una solución aproximada o cercana a la óptima cuando los métodos clásicos no consiguen encontrar la solución exacta. En este sentido, en contraposición a lo que ocurre con los algoritmos deterministas, por norma general los metaheurísticos no garantizan que se pueda encontrar una solución óptima global para algunos tipos de problemas. (MacDonald, 2005)

Realmente el objetivo de las técnicas heurísticas es producir una solución que sea suficientemente buena para el problema en cuestión en un plazo de tiempo razonable. Es decir; esta solución puede no ser la mejor y simplemente ser una aproximación, pero su valor radica en que el tiempo de búsqueda no es prohibitivamente largo.

Los algoritmos metaheurísticos se conciben como “algoritmos inteligentes” que son capaces de desarrollar un autoaprendizaje mediante imitación de procesos y comportamientos que surgen en la naturaleza, la sociología y otras disciplinas. En las últimas cuatro décadas se han desarrollado numerosas e innovadoras técnicas metaheurísticas.(Coello et al., 2007)

1.5 Armaduras planas. Conceptualización, utilización, principales tipologías.

Uno de los elementos estructurales más usados en la actualidad son las Cerchas o Armaduras, las cuales soportan cargas elevadas y cubren grandes luces, generalmente se utilizan en cubiertas de techos y puentes.

1.5.1 Armaduras planas

Es una estructura reticulada simple formada por elementos rectos de sección constante, cuya longitud supera varias veces su sección transversal, se conocen como barras y se conectan rígidamente en sus extremos denominados nodos o nudos, los esfuerzos actúan a lo largo de su eje longitudinal. Las Armaduras planas o cerchas se utilizan para soportar cargas elevadas y cubrir grandes luces, pueden construirse en maderas o acero y usadas en cubiertas de techos, puentes, grúas, torres, etc.

Una armadura plana se idealiza como un sistema de miembros en un plano e interconectados en juntas articuladas. Las cargas pueden consistir de fuerzas concentradas en los nudos que dará como resultado fuerzas axiales de tensión y compresión en los elementos.

Construcción de una armadura

Con el fin de obtener la rigidez de la armadura las barras deben tener una disposición triangular, por ser geoméricamente una figura indeformable, unidas de dos en dos en sus extremos mediante pasadores lisos. Las uniones de las barras se llaman nudos, nudos o juntas y se resuelven generalmente con placas metálicas llamadas cartelas. Partiendo del triángulo base, formado por 3 nudos (ABC) y tres barras (AB, AC, BC) por cada nuevo

nudo (D), se necesitan dos barras (BD, CD), no alineadas, para formar un nuevo triángulo, generando estructuras rígidas.(Jiménez. 2004).

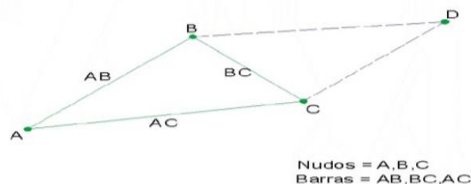


Figura 1.6 Triángulo base (Jiménez. 2004).

Condición de rigidez de las armaduras

La rigidez de una armadura está determinada por su capacidad de mantener la forma original luego de ser aplicadas las cargas de trabajo. La rigidez mide la estabilidad estructural de la armadura.

Equilibrio en las armaduras

Externamente se equilibran mediante apoyos isostáticos. Los extremos de cada barra son articulaciones de pasador permitiendo el giro, alrededor del nudo, el sistema de fuerzas sobre el nudo es concurrente.

Cada barra de la armadura se encuentra sometida a un sistema de dos fuerzas, axiales, iguales, opuestas y colineales, que la mantienen en equilibrio.

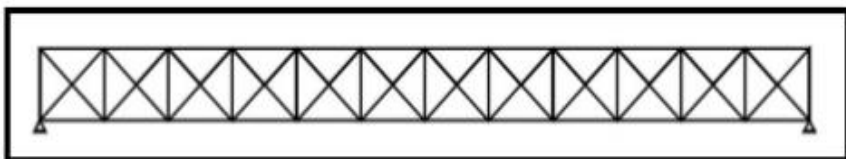
1.5.2 Tipologías de armaduras planas.

Existen diferentes tipologías pero su diferencia radica en la disposición de sus elementos, a continuación se describen algunos de los tipos de armaduras más usadas en la ingeniería.(Bellido et al., 2015)

Armaduras Long

Este tipo de armadura debe su nombre a Stephen H. Long (1784-1864), y tiene su origen hacia 1835. Los cordones superior e inferior horizontales se unen mediante montantes

verticales todos ellos arriostrados por diagonales dobles, usados para aumentar la rigidez de la estructura y su capacidad de resistir cargas laterales, tales como los movimientos sísmicos y la presión de los vientos huracanados.



- Figura 1.7 Armaduras Long(Bellido et al., 2015).

Armaduras Howe

La armadura *Howe* es la inversa a la *Pratt*, fue patentada en 1840 por Wiliam Howe, aunque ya había sido usada con anterioridad. Se usó mucho en el diseño de celosías de madera, está compuesta por montantes verticales entre el cordón superior e inferior. Las diagonales se unen en sus extremos donde coincide un montante con el cordón superior o inferior. Con esa disposición se lograba que los elementos verticales, que eran metálicos y más cortos estuvieran tensionados, mientras que las diagonales más largas estaban comprimidas, lo cual era económico puesto que los elementos metálicos eran más cros y con esta disposición *Howe* se minimizaba su longitud.

Las de dos aguas, son los tipos más comunes de armaduras de peralto medio, y tienen luces máximas de 27 o 30 m.

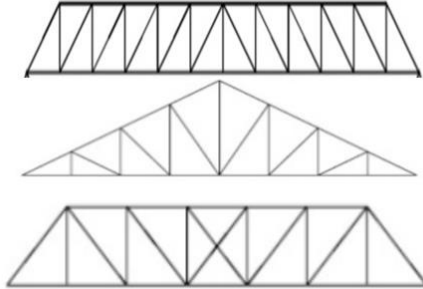


Figura 1.8 Armaduras tipo Howe (Bellido et al., 2015)

Armaduras Warren

La armadura *Warren*, fue patentada por los ingleses James Warren y Willoughby Monzony, en 1840. El rasgo característico de este tipo de armaduras es que forman una serie de triángulos isósceles (o equiláteros), de manera que todas las diagonales tienen la misma longitud. Típicamente en una celosía de este tipo y con cargas aplicadas verticales en sus nudos superiores, las diagonales presentan alternativamente compresión y tracción. Esto, que es desfavorable desde el punto de vista resistente, presenta en cambio una ventaja constructiva. Si las cargas son variables sobre la parte superior de la celosía (como por ejemplo en una pasarela) las armaduras presentan resistencia similar para diversas configuraciones de carga.

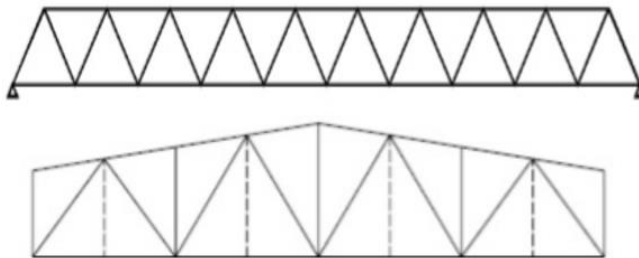


Figura 1.9 Armadura tipo Warren. (Bellido et al., 2015)

Armaduras Vierendeel

La armadura Vierendeel, en honor al ingeniero belga A. Vierendeel, tiene como características principales:

- Las uniones obligatoriamente rígidas.
- Ausencia de diagonales inclinadas.

De esta manera no aparecen formas triangulares como en la mayoría de las armaduras, sino una serie de marcos rectangulares. Se trata por tanto de una armadura empleada en edificación por el aprovechamiento de sus aperturas.

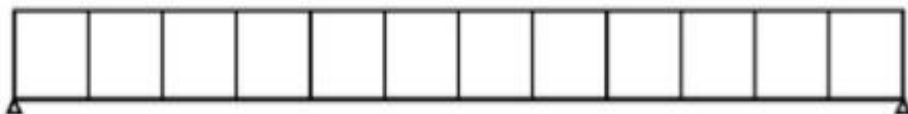


Figura 1.10 Armaduras Vierendeel.(Bellido et al., 2015)

Armaduras Pratt Plana

Representa la adaptación de las armaduras al uso más generalizado de un nuevo material de construcción de la época: el acero. A diferencia de una armadura *Howe*, las barras están inclinadas en sentido contrario, de manera que las diagonales están sometidas a tensión, mientras que las barras verticales están comprimidas.

En esencia tiene una tipología y uso muy parecidos al *Warren*. Para la armadura de cuerdas paralelas, la *Pratt* ofrece las ventajas:

- Menos efecto de pandeo (los miembros más largos a tracción y los cortos a compresión)
- Representa la adaptación de un nuevo material de la época: el acero.
- Se usan en techos de luces moderados entre 18 y 30 metros. Si se requiere de mayor luz serían más recomendable las armaduras de abanico o las armaduras *Fink*.

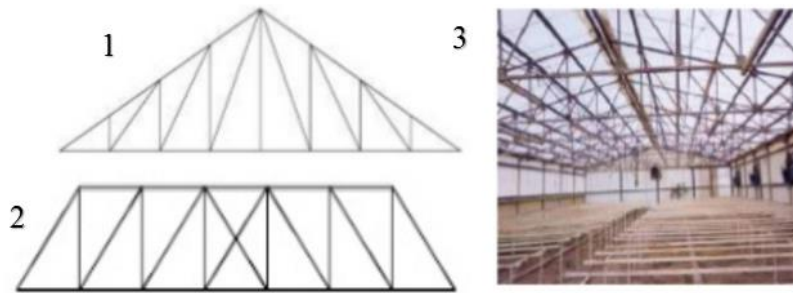


Figura 1.11 Armaduras Pratt 1.) Armadura tipo Pratt 2.) Armadura tipo Pratt (cuerdas paralelas)3.) Armadura tipo Pratt (invernadero).(Bellido et al., 2015)

Armaduras Fink

Para techos mayores (más de 15 grados) las armaduras *Fink* son muy usadas, las *Howe* y *Pratt* también pueden usarse pero no son tan económicas, la armadura *Fink* ha sido utilizada para claros del orden de los 37 metros. Un hecho que la hace más económica es que la mayoría de sus miembros está en tensión, mientras que los sujetos a compresión son bastante cortos, además es importante saber que la triangulación de una armadura se proyecta tomando en cuenta el espaciamiento de los largueros, ya que usualmente es conveniente localizar los largueros sólo en los vértices de los triángulos, la triangulación principal puede subdividirse. La armadura *Fink* puede ser dividida en un gran número de triángulos y coincidir con cualquier espaciamiento de largueros.

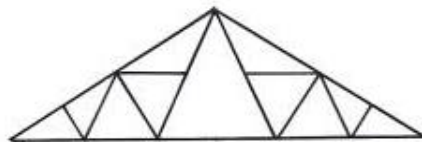


Figura 1.12 Armadura tipo Fink(Bellido et al., 2015)

Armadura tipo tijera

Cercha a dos aguas cuyas barras de tracción van desde la base de cada cordón hasta el punto medio del otro cordón superior. Se puede emplear para distancias pequeñas que requieran altura libre superior.

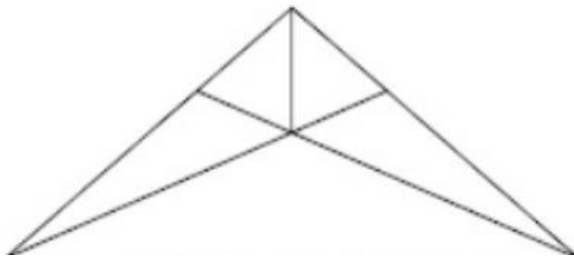


Figura 1.13 Armadura tipo tijera.(Bellido et al., 2015)

Armadura Diente de sierra

Estas armaduras pueden utilizarse cuando la separación entre columnas no es objetable y se desea una iluminación natural adecuada por medio de ventanales en construcciones anchas. Sus caras más inclinadas llevan los ventanales y están generalmente orientadas al norte para una iluminación difusa más pareja. Estructuralmente es una estructura aporticada muy eficiente y se usa mucho en fábricas de textiles.

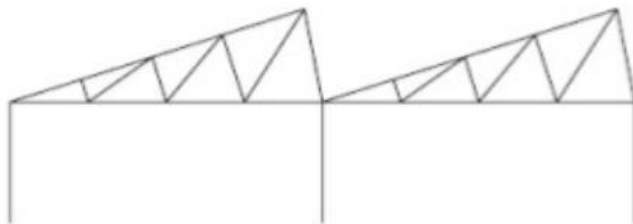


Figura 1.14 Armadura Diente de sierra.(Bellido et al., 2015)

Armadura de cuerda y arco

Se emplea para garajes, casas decampo, gimnasios y hangares pequeños.

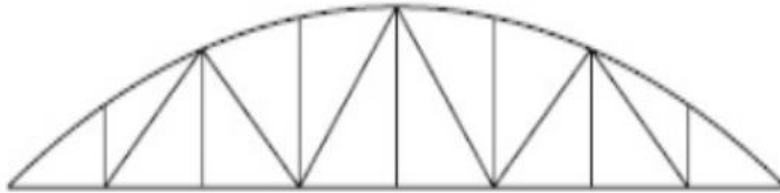


Figura 1.15 Armadura de cuerda y arco.(Bellido et al., 2015)

Conclusiones Parciales

1. Las armaduras planas son elementos estructurales simples que trabajan a flexión, compuestas por elementos dispuestos de forma que sean capaces de resistir las cargas y transmitir las a los apoyos actuando como una unidad, su forma más simple es la triangular, por ser esa figura geométrica que proporciona mayor rigidez y estabilidad, trabajan en conjunto como una viga, soportan cargas de flexión, pero sus miembros individualmente soportan cargas de tracción y compresión, se emplean para cubrir grandes luces, se construyen de madera, acero u hormigón armado.
2. Los algoritmos genéticos tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez, si un camino resulta ser un callejón sin salida, pueden eliminarlo fácilmente y continuar el trabajo en avenidas más prometedoras, dándoles una mayor probabilidad en cada ejecución de encontrar la solución, además constituyen junto con las estrategias y la programación evolutiva un nuevo enfoque a la solución de numerosos problemas de optimización y búsqueda dentro de lo que se puede destacar su aplicación en diversas ramas de la Ingeniería Civil.

CAPÍTULO 2 ANÁLISIS DE LAS DIFERENTES TIPOLOGÍAS DE ARMADURAS Y FUNCIONAMIENTO DE LOS ALGORITMOS GENÉTICOS.

Este capítulo tiene como objetivo fundamental describir las tipologías que serán las que están en el *software SAP 2000* en las plantillas de armaduras 2D (*Pratt, English, Vertical*), y describir matemáticamente usando el método de los nudos y el de las secciones, las funciones de cada barra, suponiendo una cantidad de divisiones entre apoyos de hasta 50 fraccionamientos, para realizar una programación empleando el *software MATLAB* y así optimizar el diseño de éstas tipologías de armaduras planas utilizando técnicas heurísticas, en este caso como se aborda en el capítulo anterior a través de los Algoritmos Genéticos. Posteriormente se definen las funciones a optimizar y las variables que influyen en ella.

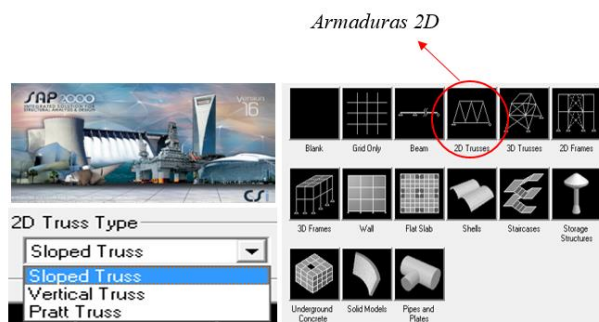


Figura 1.16 Plantillas de armaduras 2D del Software SAP2000.

Fuente: Elaborado por la autora.

2.1 Métodos para la determinación de las funciones.

Las armaduras planas pueden ser calculadas con suficiente aproximación, sin considerar las deformaciones, usando únicamente ecuaciones de estática. Existen diversos métodos basados en aplicar las ecuaciones de la estática de manera eficiente y rápida, para una armadura de n nudos:

- **Método de los nudos:** consiste en estimar que cada uno de los nudos está en equilibrio, lo que implica que la suma vectorial de las fuerzas actuantes sobre cada barra se equilibran. Al existir n nudos es necesario resolver $2n$ ecuaciones lineales. Este método sólo funciona para celosías estáticamente determinadas (internamente isostáticas) con $2n-3$ barras, siendo n el número de nudos. Para armaduras complejas el método de los nudos conduce a un sistema con más incógnitas que ecuaciones y no permite determinar los esfuerzos.
- **Método de Cremona-Maxwell:** es un sencillo método gráfico basado en el método de los nudos, usando una operación de dualidad geométrica, por la cual, a cada estructura reticular se le asigna un diagrama de puntos, en donde cada punto representa una retícula de la estructura, y cada segmento, entre estos puntos dados, representa la magnitud del esfuerzo de la barra situada entre dos retículos. La suma vectorial de las fuerzas actuantes sobre cada barra se equilibra gráficamente.
- **Método de Ritter o de las secciones:** consiste en realizar cortes en una armadura con el fin de encontrar las fuerzas internas en una armadura, tomando en cuenta la sección cortada en equilibrio y utilizando las 3 ecuaciones de equilibrio determinar las fuerzas internas. Este método únicamente permite realizar un corte en el cual se corten 3 barras (al menos una de las cuales no sea paralela a las otras dos).
- **Método matricial** que requiere resolver un sistema de $2n-3$ ecuaciones para los desplazamientos desconocidos, a partir del cual se calculan fácilmente las reacciones y los esfuerzos sobre las barras. En general resulta algorítmicamente más trabajoso que los otros, pero es fácilmente programable y tiene la gran ventaja de ser extensible casi sin modificaciones a celosías externamente hiperestáticas.(Bellido et al., 2015)

2.2 Breve reseña de los métodos de los nudos y de las secciones.

El primer análisis científico de armadura fue realizado en 1847 por Squire Whipple, un constructor de puentes norteamericano de la ciudad de Utica, Nueva York. En 1850 D.J. Jouraawski, un ingeniero ferroviario ruso, creó el método de solución de los nudos, por el cual se obtienen los esfuerzos en los miembros considerando las condiciones de equilibrio de cada nudo a la vez, sin embargo esto no se conoció en Occidente hasta que el ingeniero ferroviario alemán Kart Culmann, profesor del Politécnico de Zurich, lo publicó independientemente unos años después en 1866.

En 1862 el ingeniero alemán A. Ritter, planteó otro método analítico: el método de las secciones. Ritter cortó la armadura a lo largo de una línea imaginaria y sustituyó las fuerzas internas por fuerzas externas equivalentes. Haciendo sumatoria de momento en puntos convenientes (puntos de *Ritter*) pueden obtenerse todas las fuerzas internas.

Estos métodos de cálculo suponen que los miembros de las armaduras se unen por articulaciones y en realidad las primeras armaduras así se unieron. Por ejemplo: la armadura patentada por el inglés James Warren en 1848 eran miembros de hierro colado que trabajaban a compresión o tensión con agujeros para los pasadores incorporados en la fundición: una clásica articulación.

2.2.1 Método de los nudos.

Se denomina estructura a cualquier sistema de cuerpos unidos entre sí que sea capaz de ejercer, soportar o transmitir esfuerzos. Las estructuras están formadas por partes interconectadas entre sí llamadas barras, las cuales se diseñan determinado la fuerza y los pares o momentos que actúan sobre ellas. Las barras están unidas en sus extremos por articulaciones o nudos.

Procedimiento de análisis

1. Trazar el diagrama de cuerpo libre de un nudo que tenga por lo menos una fuerza conocida y como máximo dos fuerzas desconocidas.
2. Usar cualquier método para establecer el sentido de una fuerza desconocida.

3. Orientar los ejes x,y de manera que las fuerzas en el diagrama de cuerpo libre puedan ser resueltas fácilmente en sus componentes x,y. Luego aplicar las dos ecuaciones de equilibrio de fuerzas y obtener las fuerzas desconocidas y verificar el sentido correcto.
4. Continuar con el análisis de cada uno de los demás nudos, que tenga cuando menos dos incógnitas y por lo menos una fuerza conocida.
5. Una vez encontrada la fuerza en un miembro a partir del análisis de un nudo en uno de sus extremos, el resultado se puede utilizar para analizar las fuerzas que actúan en el nudo en su otro extremo.

2.2.2 Método de las secciones.

El método de las secciones se usa para determinar las cargas que actúan dentro de un cuerpo. Este método se basa en el principio de que si un cuerpo está en equilibrio, entonces cualquier parte del cuerpo está también en equilibrio.

Este método puede usarse también para cortar o seccionar los miembros de toda una armadura. Si la sección pasa por la armadura y se traza el diagrama de cuerpo libre de cualquiera de sus dos partes, entonces se pueden aplicar las ecuaciones de equilibrio o esa parte para determinar las fuerzas del miembro en la sección cortada.

Como sólo tres ecuaciones independientes de equilibrio pueden ser aplicadas a la parte aislada de la armadura, se debe seleccionar una sección que, en general, pase por no más de tres miembros en que las fuerzas sean desconcentradas.

$$\sum F_x = 0 \quad (2.1)$$

$$\sum F_y = 0 \quad (2.2)$$

$$\sum M = 0 \quad (2.3)$$

Aplicación del método de las secciones:

Ecuaciones de equilibrio

Los momentos deben sumarse con respecto a un punto que se encuentra en la intersección de las líneas de acción de dos fuerzas desconocidas y las fuerzas internas serán determinadas directamente a partir de la ecuación de momento.

Si dos de las fuerzas desconocidas son paralelas, las otras fuerzas se pueden sumar perpendicularmente a la dirección de esas incógnitas para determinar directamente la tercera fuerza desconocida. (Bellido et al., 2015)

2.3 Descripción de las diferentes tipologías.

Con la aplicación de los métodos explicados en el epígrafe anterior se obtienen las funciones para cada tipología analizada (referenciadas en el Anexo) y de ellas se derivan sus respectivas variables.

Sloped

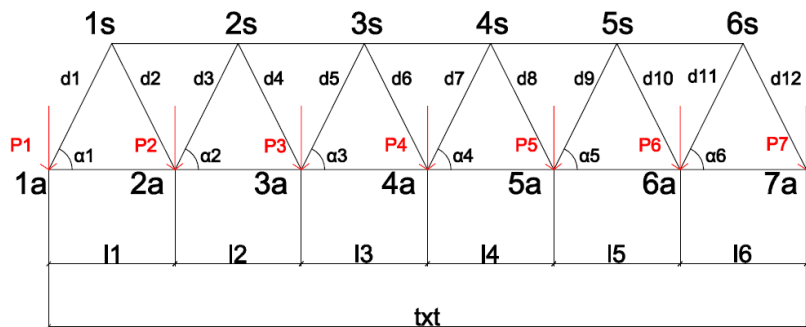


Figura 2.1 Armadura Sloped.

Fuente: Elaborado por la autora en el software AutoCAD.

Vertical

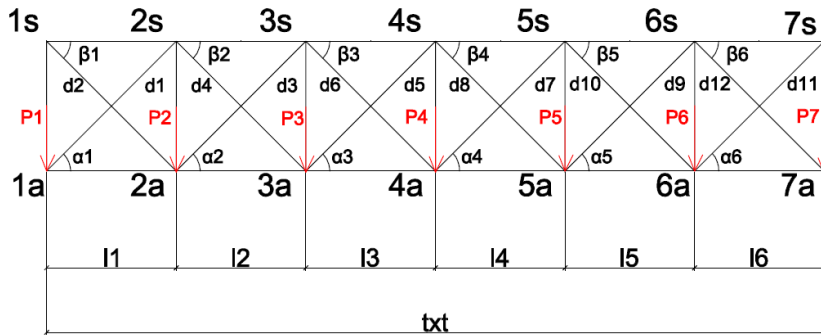


Figura 2.2 Armadura Vertical.

Fuente: Elaborado por la autora en el software AutoCAD.

Pratt

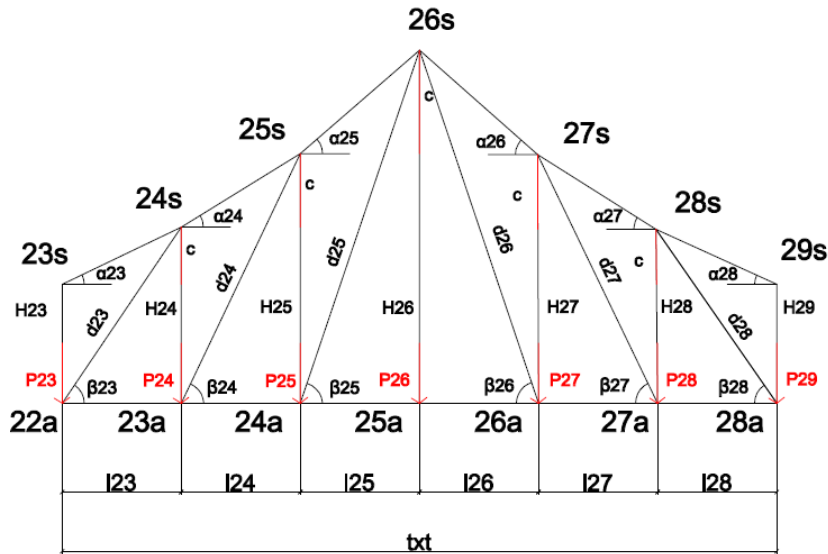


Figura 2.3 Armadura Pratt.

Fuente: Elaborado por la autora en el software AutoCAD.

2.4 Definición de las funciones a optimizar y las variables que influyen en ella.

Como se aprecia en la Figura 2.4, en el caso de la armadura *Sloped* se definen como variables, la altura (h), las longitudes (l_n) y diagonales (d_n) y además la mitad de cada longitud que se define como (x_n), por lo que consta con un total de 126 variables, es válido aclarar que éstas variables se analizan solo 25 fraccionamientos, pues se tiene en cuenta que la armadura es simétrica.

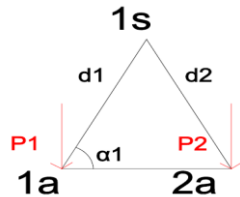


Figura 2.4 Variables de la armadura Sloped.

Fuente: Elaborado por la autora en el software AutoCAD.

En el caso de la *Vertical*, Figura 2.5, se definen como variables, la altura (H), las longitudes (l_n) y diagonales (d_n) y además una incógnita x que necesita el *softwareMATLAB* para prever los resultados esperados, por lo que consta con un total de 77 variables y al igual que en la anterior se tiene en cuenta que la armadura es simétrica por lo que se analiza solo la mitad.

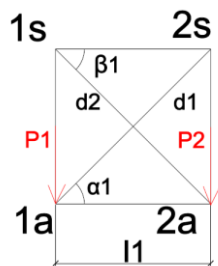


Figura 2.5 Variables de la armadura Vertical.

Fuente: Elaborado por la autora en el software AutoCAD.

Mientras que en la Pratt, Figura 2.6, existe un total de 105 variables, dentro de las que se incluyen las longitudes (l_n), las respectivas alturas (H_n), las diagonales (d_n), los cordones superiores (b_n) y como se observa en la figura siguiente una incógnita denominada c que es la relación entre alturas.

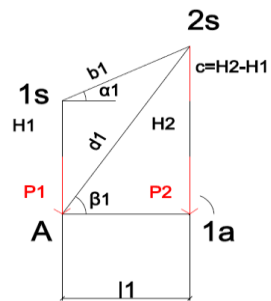


Figura 2.6 Variables de la armadura Pratt.

Fuente: Elaborado por la autora en el software AutoCAD.

2.5 Análisis de los parámetros de AG.

2.5.1 PopulationType

Este parámetro tiene en cuenta el tipo de población ingresada, para esta programación se utiliza *doubleVector* pero el software *MATLAB* brinda otras opciones como:

- *bitstring*
- *custom*
- *doubleVector*

2.5.2 PopInitRange

En este parámetro se concreta el rango inicial de valores que una población puede tener utilizando:

- *Matrix*

2.5.3 PopulationSize

El tamaño de población se define como el escalar positivo que indica el número de individuos.

- *positive scalar*

2.5.4 EliteCount

Este parámetro lo que hace es copiar el mejor individuo de la población anterior a la siguiente, lo que permite una mejor convergencia del algoritmo, es decir es el valor escalar positivo que indica el número de los mejores individuos que sobrevivieron para la segunda generación sin cambios.

- *positive scalar*

2.5.5 CrossoverFraction

Valor escalar positivo de la fracción de genes intercambiados entre individuos.

- *positive scalar*

2.5.6 ParetoFraction

En este caso se hace referencia a soluciones en las que no se puede mejorar más los valores de una variable, sin empeorar alguna de las otras condiciones, es decir es el escalar positivo que indica la fracción de la población en el frente no dominado.

- *positive scalar*

2.5.7 MigrationDirection

Dirección que se adapta a los individuos de las diferentes subpoblaciones que pueden migrar a otras subpoblaciones, en este caso se trabaja con *forward*.

- *Both*
- *forward*

2.5.8 MigrationInterval

Escalar positivo que define la cantidad de generaciones entre la migración de los individuos más aptos para otras subpoblaciones.

- *positive scalar*

2.5.9 MigrationFraction

Valor escalar positivo que indica la fracción de aquellos individuos que obtienen el mejor puntaje que migrará.

- *positive scalar*

2.5.10 Generations

En este parámetro se define el valor escalar positivo que muestra el número máximo de generaciones permitidas.

- *positive scalar*

2.5.11 TimeLimit

Parámetro que define el valor escalar positivo del tiempo máximo (en segundos) permitido.

- *positive scalar*

2.5.12 FitnessLimit

Límite de condición física: valor escalar mínimo de la función de aptitud deseada.

- *scalar*

2.5.13 StallGenLimit

Valor positivo escalar que define el tiempo máximo durante el cual el cambio en la función de aptitud física es menor que cero.

- *positive scalar*

2.5.14 TolFun

Tolerancia de terminación en el valor escalar positivo de la función de aptitud.

- *positive scalar*

2.5.15 TolCon

Parámetro que define el valor escalar positivo de la tolerancia de terminación en restricciones.

- *positive scalar*

2.5.16 InitialPopulation

La población inicial utilizada en la siembra de algoritmo AG; puede ser parcial.

- *Matrix*

2.5.17 InitialScores

Los puntajes iniciales utilizados para determinar la condición física; usada al sembrar el algoritmo AG; que puede ser parcial.

- *column vector*

2.5.18 Initial Penalty

Valor escalar positivo inicial del parámetro de penalización.

- *positive scalar*

2.5.19 PenaltyFactor

Parámetro de actualización de penalización.

- *positive scalar*

2.5.20 CreationFcn

Función utilizada para generar la población inicial, en este caso se utiliza *gacreationuniform*.

- *gacreationlinearfeasible*
- *gacreationuniform*

2.5.21 FitnessScalingFcn

En este parámetro se trabaja con *fitscalingrank* que es la función utilizada para escalar puntuaciones de aptitud.

- *fitscalingshiftlinear*
- *fitscalingprop*
- *fitscalingtop*
- *fitscalingrank*

2.5.22 SelectionFcn

Función *selectiontournament*, utilizada en la selección de padres para la próxima generación.

- *selectionremainder*
- *selectionuniform*
- *selectionroulette*
- *selectiontournament*
- *selectionstochunif*

2.5.23 CrossoverFcn

Función *crossoversinglepoint* utilizada para definir el fragmento de genes intercambiados entre individuos.

- *crossoverheuristic*
- *crossoverintermediate*
- *crossoversinglepoint*
- *crossovertwopoint*
- *crossoverarithmetic*
- *crossoverscattered*

2.5.24 MutationFcn

Función *mutationadaptfeasible* utilizada en genes mutantes.

- *mutationuniform*
- *mutationadaptfeasible*
- *mutationgaussian*

2.5.25 DistanceMeasureFcn

Función utilizada para medir la distancia promedio de individuos de sus vecinos.

- *distancecrowding*

2.5.26 Display

Nivel de visualización donde se utiliza la función *iter*.

- *off*
- *iter*
- *diagnose*
- *final*

2.5.27 OutputFcns

Función (es) llamada en cada generación. Esto es más general que *PlotFcns*.

2.5.28 PlotFcns

Función (es) utilizada (s) para trazar varias cantidades durante la simulación. Normalmente no se grafica ninguna para ahorrar o disminuir el costo computacional.

2.5.29 PlotInterval

El número de generaciones entre los resultados de trazado.

- *positive scalar*

2.5.30 Vectorized

La función objetivo está vectorizada y puede evaluar más de un punto en una llamada, pero este parámetro no se utiliza por lo que se selecciona *off*.

- *On*
- *Off*

2.5.31 UseParallel

Parámetro utilizado para evaluar objetivos y funciones no lineales de restricción, que no es necesario en esta programación por lo que se utiliza *never*.

- *always*
- *never*

Conclusiones Parciales

1. Con el previo análisis de los diferentes tipos de armaduras y la definición de las funciones a optimizar y las variables que influyen en ella, se obtiene una programación en el *software MATLAB*, aplicando la técnica heurística seleccionada (Algoritmos Genéticos), que es una herramienta muy útil para los ingenieros, pues permite obtener geometrías óptimas, lo que conlleva a estructuras más ligeras y económicas.
2. Se obtuvo el comportamiento de cada una de las armaduras analizadas a través de los métodos de los nudos y las secciones, definiéndose las variables y restricciones en cada uno de los problemas tratados.

CAPÍTULO 3 CASO DE ESTUDIO

Este capítulo tiene como objetivo fundamental obtener los valores optimizados de las longitudes o incógnitas de cada armadura definidas en el capítulo anterior y compararlos con los obtenidos por experiencia del diseñador a partir de un caso de estudio, para de esta forma validar las programaciones elaboradas demostrando así la garantía de tener una herramienta tan útil que no sólo facilita el trabajo de los ingenieros sino también ofrece los resultados esperados en un corto período de tiempo y sin la necesidad de recurrir a la experiencia y percepción del diseñador.

3.1 Análisis de los resultados obtenidos en cada caso de estudio con las mejores heurísticas.

Se propone validar las programaciones realizadas a partir de un problema aleatorio, en el cual se requiere salvar una longitud de 50 m, con apoyos extremos (simplemente apoyados) y sin apoyos intermedios; usando la menor variedad de perfiles posibles para una mejor facilidad de armado. Para ello se utiliza una carga de 25 kN/m , una altura mínima de 3 m y una máxima de 4 m, los intercolumnios están espaciados a 2,5 m. Estos datos serán los mismos para las tres tipologías analizadas (*Pratt*, *Sloped* y *Vertical*).

3.2 Posibles soluciones.

Este análisis puede llevarse a cabo por dos vías, una por el conocimiento de quien diseñe y la otra utilizando la programación elaborada en el *software MATLAB* en el capítulo anterior. Como se puede apreciar para realizar el análisis de estas tipologías, prescindiendo de dicha programación, es necesario realizar un predimensionamiento (donde los resultados obtenidos dependan de la experiencia del diseñador), lo que no sucede si se analizan las propuestas de las diferentes tipologías a través de la programación elaborada. Para ello se mantiene la longitud a salvar de 50 m, los apoyos extremos (simplemente apoyados) y sin apoyos intermedios y la carga de 25 kN/m , con la diferencia de que en este caso no es necesario el predimensionamiento. Los valores

adquiridos por ambas vías fueron modelados en el software SAP2000 obteniendo las siguientes secciones para cada tipología.

3.2.1 Solución propuesta por el diseñador (Sloped).

Sloped

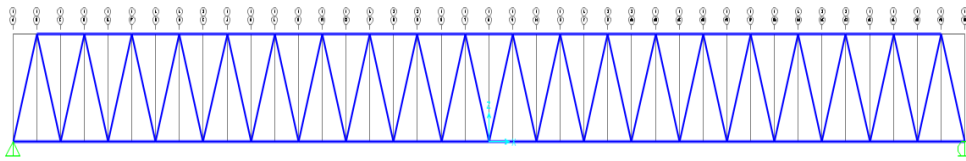


Figura 3.1 Armadura Sloped.

Fuente: Elaborado por la autora en el software SAP2000.

Sección recomendada por el software: S20x96

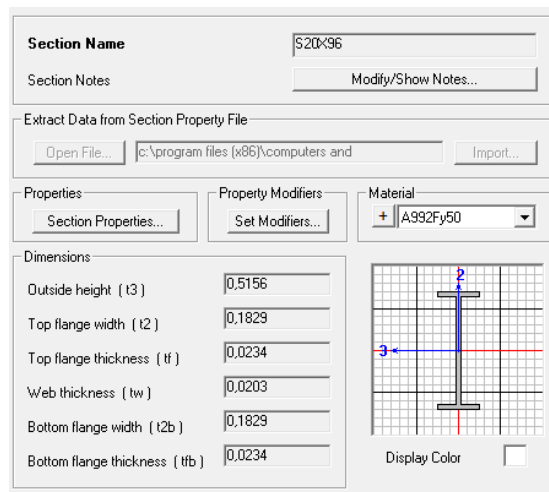


Figura 3.2 Sección S20x96.

Fuente: Software SAP2000

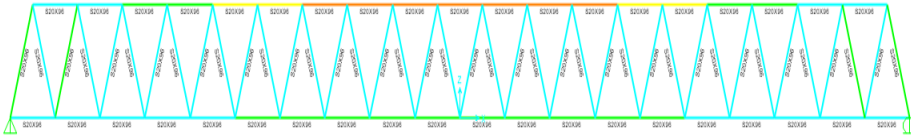


Figura 3.3 Chequeo de la estructura con la sección S20x96.

Fuente: Software SAP2000.

Como se aprecia en la Figura 3.3 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

3.2.2 Solución Optimizada (Sloped).

Sloped Optimizada

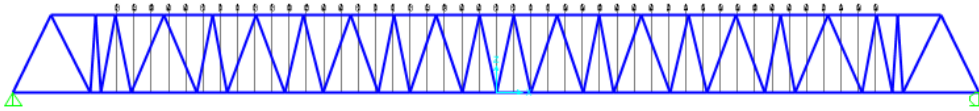


Figura 3.4 Armadura Sloped.

Fuente: Elaborado por la autora en el software SAP2000.

Sección recomendada por el software: S20x86

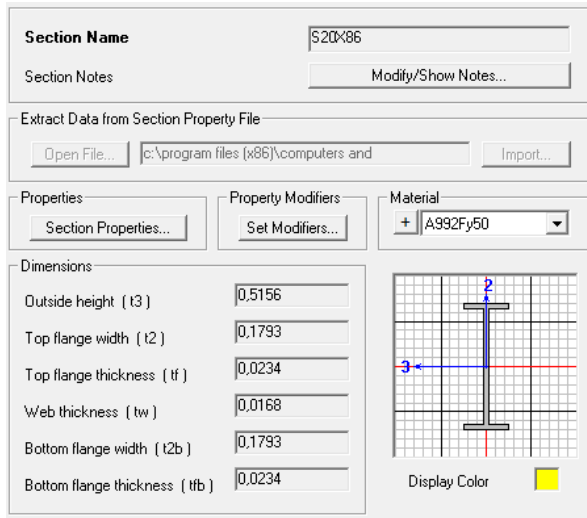


Figura 3.5 Sección S20x86.

Fuente: Software SAP2000.

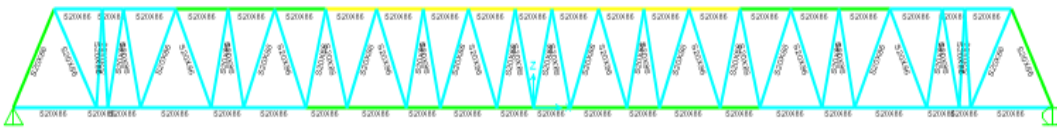


Figura 3.6 Chequeo de la estructura con la sección S20x86.

Fuente: Software SAP2000.

Como se aprecia en la Figura 3.6 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Al comparar ambas soluciones se refleja claramente que en la propuesta optimizada se utiliza una menor sección (S20x86), con un área de 0.0163m^2 , lo que representa una disminución del 10.5% con respecto a la sección propuesta por el diseñador (S20x96) con un área de 0.0182m^2 . En cuanto a la economía los perfiles tienen un precio de 89.96

CUP y 100.42 CUP respectivamente por lo que si se utiliza la armadura optimizada conllevaría a un ahorro de 1591.43 CUP por metro lineal.

3.2.3 Solución propuesta por el diseñador (Pratt).

Pratt

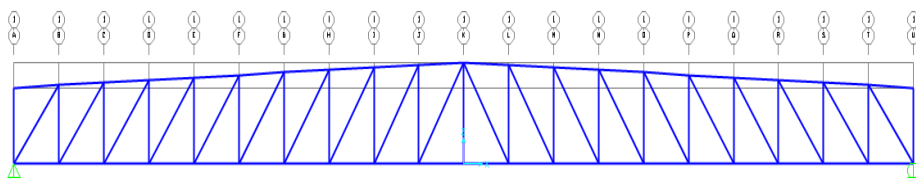


Figura 3.7 Armadura Pratt.

Fuente: Elaborado por la autora en el software SAP2000.

Sección recomendada por el software: S24x106.

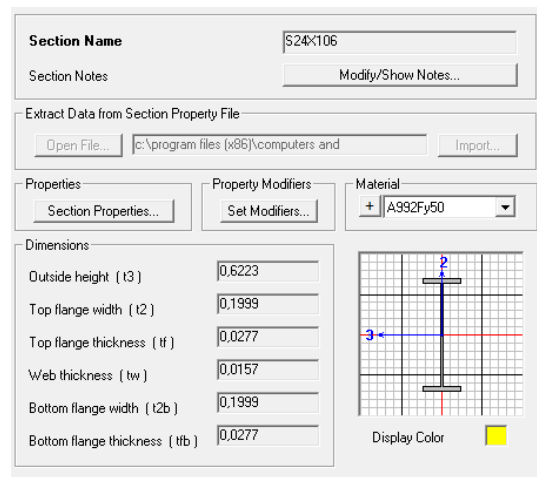


Figura 3.8 Sección S24x106.

Fuente: Software SAP2000

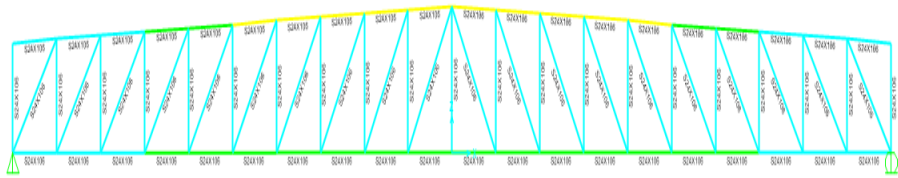


Figura 3.9 Chequeo de la estructura con la sección S24x106.

Fuente: Software SAP2000

Como se aprecia en la Figura 3.9 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

3.2.4 Solución Optimizada (Pratt).

Pratt Optimizada

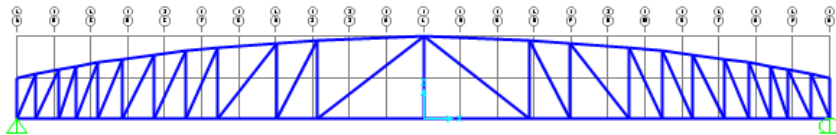


Figura 3.10 Armadura Pratt Optimizada.

Fuente: Elaborado por la autora en el software SAP2000.

Sección recomendada por el software: S20x86

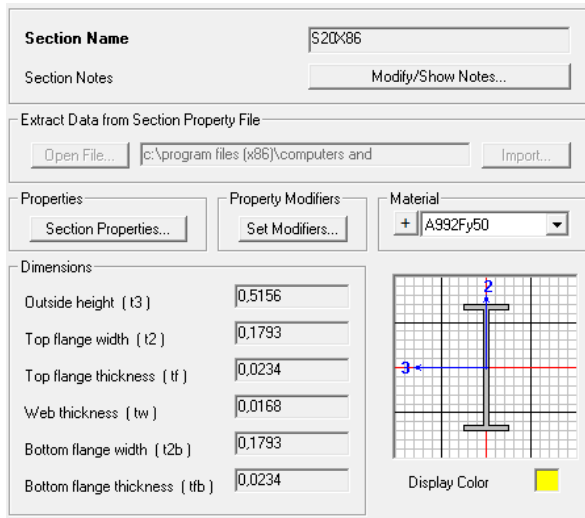


Figura 3.11 Sección S20x86.

Fuente: Software SAP2000



Figura 3.12 Chequeo de la estructura con la sección S20x66.

Fuente: Software SAP2000

Como se aprecia en la Figura 3.12 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Al igual que en la *Sloped* si se comparan ambas soluciones se puede apreciar que en la propuesta optimizada se utiliza una menor sección (S24x106), con un área de 0.0201m^2 , lo que representa una disminución de un 18.9% con respecto a la sección propuesta por el diseñador (S20x86), con un área de 0.0163m^2 . En cuanto a la economía los perfiles tienen

un precio de 110.88 CUP y 89.96 CUP respectivamente por lo que si se utiliza la armadura optimizada conllevaría a un ahorro de 3005.27 CUP por metro lineal.

3.2.5 Solución propuesta por el diseñador (Vertical).

Vertical

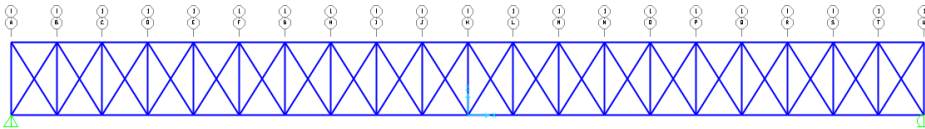


Figura 3.13 Armadura Vertical.

Fuente: Elaborado por la autora en el software SAP2000.

Sección recomendada por el software: S24x106

Figura 3.14 Sección S24x106.

Fuente: Software SAP2000

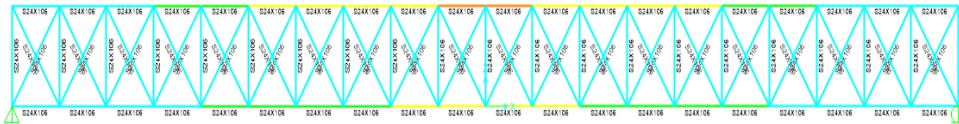


Figura 3.15 Chequeo de la estructura con la sección S24x106.

Fuente: Software SAP2000

Como se aprecia en la Figura 3.15 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

3.2.6 Solución Optimizada (Vertical).

Vertical Optimizada

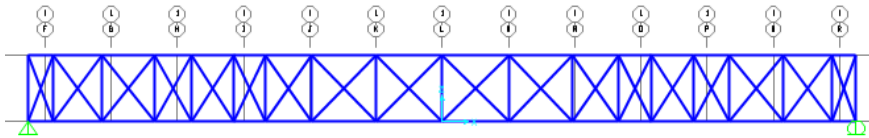


Figura 3.16 Armadura Vertical Optimizada.

Fuente: Elaborado por la autora en el software SAP2000.

Sección recomendada por el software: S20x86

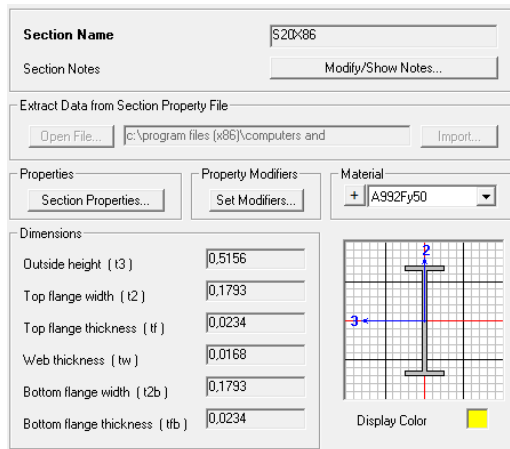


Figura 3.17 Sección S20x86.

Fuente: Software SAP2000.

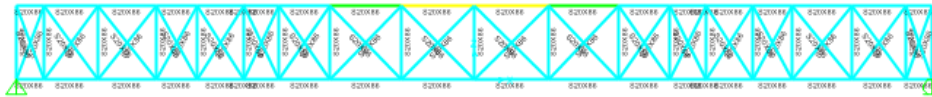


Figura 3.18 Chequeo de la estructura con la sección S20x86.

Fuente: Software SAP2000.

Como se aprecia en la Figura 3.18 no existe ningún elemento sobretensionado por lo que se puede utilizar esta sección en esta tipología.

Se comparan ambas soluciones y se aprecia que en la propuesta optimizada se utiliza una menor sección (S24x106), con un área de 0.0201m^2 , lo que representa una disminución de un 18.9% con respecto a la sección propuesta por el diseñador (S20x86), con un área de 0.0163m^2 . En cuanto a la economía los perfiles tienen un precio de 110.88 CUP y 89.96 CUP respectivamente por lo que si se utiliza la armadura optimizada conllevaría a un ahorro de 9576.66 CUP por metro lineal.

Conclusiones parciales:

- 1- Al analizar los resultados obtenidos en cada caso de estudio con la aplicación de los Algoritmos Genéticos con las programaciones establecidas y los obtenidos por la experiencia del diseñador, se puede apreciar cuan eficiente resulta ser ésta técnica aplicada, pues se reducen las secciones en todos los casos en los que no se realizó un previo dimensionamiento, demostrando que a través de la optimización estructural empleando (AG), se obtienen estructuras más ligeras, económicas y a la vez resistentes.

CONCLUSIONES

1. Las armaduras planas son elementos estructurales simples, compuestas por elementos dispuestos de forma que sean capaces de resistir las cargas y transmitir las a los apoyos actuando como una unidad, su forma más simple es la triangular, trabajan en conjunto como una viga, soportan cargas de flexión, pero sus miembros individualmente soportan cargas de tracción y compresión, se emplean para cubrir grandes luces, se construyen de madera, acero u hormigón armado.
2. Se obtuvo el comportamiento de cada una de las armaduras analizadas a través de los métodos de los nudos y las secciones, definiéndose las variables y restricciones en cada uno de los problemas tratados.
3. Al analizar los resultados obtenidos en cada caso de estudio con la aplicación de los con las programaciones establecidas y los obtenidos por la experiencia del diseñador, se puede apreciar cuan eficiente resulta ser ésta técnica aplicada, pues se reducen las secciones en todos los casos en los que no se realizó un predimensionamiento.
4. La utilización de los algoritmos genéticos en la optimización de funciones es una herramienta muy útil para los ingenieros, pues permite obtener geometrías óptimas, lo que conlleva a estructuras más ligeras y económicas, a la vez que reducen el tiempo de cálculo y disminuyen incertidumbres.

RECOMENDACIONES

1. Extender la investigación a otras tipologías de armaduras planas y espaciales.
2. Emplear otras técnicas heurísticas de optimización que pudieran resultar más eficientes en este tipo de problemas.

REFERENCIAS BIBLIOGRÁFICAS

Metodología basada en Algoritmos Genéticos y Programación en Paralelo para el Diseño Óptimo de Armaduras de Acero.

Optimización. Available: <http://code.pediapress.com/>.

2013. ¿Que es la Computación Evolutiva?

. Available: https://jarroba.com/wpcontent/uploads/2013/05/CE1_Jarroba.com_.png.

ALCAZAR, Y. 2016. *Aplicación de los algoritmos genéticos en la optimización de funciones*. Trabajo de Diploma, Universidad de Matanzas.

ALI AHRARI, ATAI, A. A. & DEB, K. 2014. Simultaneous Topology, Shape and Size Optimization of Truss Structures by Fully Stressed Design Based on Evolution Strategy.

ALLAIRE, G., GOURNAY, F., JOUVE, F. C. & TOADE, A. M. 2004a. Structural optimization using topological and shape sensitivity via a level set method.

ALLAIRE, G., JOUVE, F. C. & TOADE, A.-M. 2004b. Structural optimization using sensitivity analysis and a level-set method. *Journal Computational Physics*.

ANNICCHIARICO, W. 2007. Una metodología para la optimización estructural de formas usando principios de evolución flexible distribuida. IMME v.45 n.1.

BAQUELA, E. & REDCHUCK, A. 2013. *Optimización Matemática con R* Madrid, España.

BELLIDO, M. C., AYRA, D. & AGUADO, R. 2015. Estructura de puente de armadura Warren. Análisis aplicativo. *Diversificación Productiva y delFortalecimiento de la Educación* [Online].

BENDSOE, M. P. & SIGMUND, O. 2003. *Topology optimization: Theory, methods and applications*, Berlin, Germany.

- BERZAL, F. Computación Evolutiva.
- BIETHAN, J. & NISSEN, V. 1995. *Evolutionary algorithms in management applications*.
- BONELLI, M. 2016. Optimización estructural de armaduras planas por medio de algoritmos genéticos.
- BREMERMANN, H. J. 1962. "Optimization through evolution and recombination". Washington, D.C.
- BROWNE, P. A. 2013. *Topology Optimization of Linear Elastic Structures*. Doctor of Philosophy, University of Bath.
- CASTILLO, J. M. Algoritmos Genéticos. Universidad Internacional de la Rioja.
- CEFERINO, J. 2008. *Técnicas Heurísticas para Problemas de Diseño en Telecomunicacione*. Tesis de Máster, Universidad de Murcia.
- COELLO, RUDNICK & CHRISTIANSEN. 1995. *Using genetic algorithms for optimal design of trusses.*, Tulane University.
- COELLO, C. 2012. *Constraint-Handling techniques used with evolutionary algorithms*.EVOCINV. México D.F.
- COELLO, C., CHRISTIANSEN, C. A. & D., A. 2000. *Multiobjective optimization of trusses using genetic algorithms*.
- COELLO, C. A., LAMONT, G. B. & VAN, D. A. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition*.
- COELLO, C. A., SANTOS, F. & ALONSO, F. A. Diseño óptimo de vigas de concreto reforzado mediante Algoritmos genéticos.
- CHEN, X. 2013. *Learnig with sparsity: Structures, Optimization and Applications*. School of Computer Science Carnegie Mellon University.

- CHEN, Y. & ZOU, X. 2014. Runtime analysis of a multi-objective evolutionary algorithm for obtaining finite approximations of Pareto fronts. *ScienceDirect*.
- CHENG, J. 2010. Optimum design of steel truss arch bridges using a hybrid genetic algorithm. *Journal of Constructional Steel Research*.
- DERRAC, J., GARCÍA, S., MOLINA, D. & HERRERA, F. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *ScienceDirect*, swarm and Evolutionary Computation.
- DOMÍNGUEZ, M. Programación de algoritmos genéticos. Todo Programación.
- FRASER, A. S. 1957a. Simulation of genetic systems by automatic digital computers II: Effects of linkage on rates under selection. *Australian Journal of Biological Sciences*.
- FRASER, A. S. 1957b. "Simulation of genetic systems by automatic digital computers. I. Introduction" *Aust. J. Biol. Sci.*, Vol.10, 484-491.
- FRASER, A. S. 1958. Monte Carlo analyses of genetic models. *Nature*, 181(4603):208–209, Jan. .
- GALILEI, G. 1638. *Discorsi e dimostrazioni matematiche intorno a due nuove scienze: Louis Elseiver*.
- GOLDBERG, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*.
- GUTIERREZ, N. C., PENICHE, R. R., HERRERA, G., ALVARADO, R. & CARRIÓN, F. J. 2012. A Non-Symmetrical Solution Applying a Genetic Algorithm with Natural Crossover for the Structural Optimization of Truss Structures. . *Ingeniería Investigación y Tecnología*, Vol. XIII (número 3).
- HOLLAND, J. H. 1962. "Outline for a logical theory of adaptive systems".

- HOLLAND, J. H. 1975. *“Adaptation in Natural and Artificial Systems”*. Michigan, USA, The University of Michigan Press.
- HULTMAN, M. 2010. *Weight optimization of steel trusses by a genetic algorithm*. Master’s thesis, Lund Institute of Technology Lund Universit.
- ISON, M., SITT, J. & TREVISAN, M. 2005. Algoritmos Genéticos: aplicación en MATLAB. *Guía de la materia de Sistemas Complejos* [Online]. Available: www.df.uba.ar/users/mison/genetico.tar.gz.
- KONA, A., COIT, D. W. & SMITH, A. E. 2006. Multi-objective optimization using genetic algorithms: A tutorial. *ScienceDirect*.
- KOZIEL, S. & YANG, X.-S. 2011. *Computational Optimization, Methods and Algorithms*, Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.
- LÓPEZ, F. C. 2010. *Optimización global con algoritmos genéticos*. Diplomado, Universidad Politécnica de Catalunya.
- MACDONALD, R. C. 2005. *Genetic Algorithms with Steel Structures a Literatura Review*. degree of Master of Science, Brigham Young University.
- MACÍAS, M. 2016. *Estudio Comparativo de Técnicas de Optimización para la Actualización de Modelos de Elementos Finitos. Aplicación de los algoritmos HS y GA para la actualización de una pasarela* Tesis de Máster, Universidad de Sevilla.
- MARLER, R. T. & ARORA, J. S. 2004. Survey of multi-objective optimization methods for engineering. *Struct Multidisc Optim*.
- MARTÍNEZ, G. 2005. Optimización estructural por medio del algoritmo Genético. *Politécnica, Medelín: Politécnico Jaime Isaza Cadavid*, Vol: 1 No. 1.
- MATEOS, A. Algoritmos evolutivos y Algoritmos Genéticos. [Accessed Universidad Carlos III de Madrid].

- MERELO, J. J. 2016. Técnicas heurísticas de resolución de problemas: computación evolutiva y redes neuronales.
- MINARY, P. Advanced optimization and sampling algorithms in structure based biocomputing: From chain closure to variable transformations.
- NICOLÁS, M. V. 2006. "OPTIMIZACIÓN DE FORMA Y TOPOLOGÍA CON MALLA FIJA Y ALGORITMOS GENÉTICOS" TESIS DOCTORAL, UNIVERSIDAD POLITÉCNICA DE CARTAGENA.
- PÉREZ, D. 2016. *Diseño de cerchas planas tubulares de cordones paralelos*. Trabajo de Diploma, Universidad de Matanzas.
- PIRES, J. P. 2014. *Structural Optimization using the Finite Element Method* Dissertação para obtenção do Grau de Mestre Universidade Católica Portuguesa.
- PRENDES, B., BELLO, A. & DELCOZ, J. J. 2002. Entorno Gráfico para la optimización de estructuras metálicas en base a Algoritmos Genéticos. *INGEGRAF* [Online].
- RAMÍREZ, A. F. 2014a. Tutorial de MATLAB. Universidad Nacional de Colombia.
- RAMÍREZ, S. 2014b. *Metodología basada en Algoritmos Genéticos y Programación en Paralelo para el Diseño Óptimo de Armaduras de Acero* Maestría en Ingeniería Civil
Tesis de grado, Universidad Javeriana de Colombia.
- RAMOS, A., SÁNCHEZ, P., FERRER, J. M., BARQUÍN, J. & LINARES, P. 2010. Modelos Matemáticos de Optimización. Available: http://www.doi.icai.upcomillas.es/intro_simio.htm.
- RAMOS, A. & VITORIANO, B. Modelos matemáticos de optimización.
- REZAIEE, M. & SALARI, M. 2001. Shape Optimization Of Plate bending Structure. *International Journal of Engineering Science*, Vol.12 no.3, 165.

- RIVERA, A. 2015. Armaduras Planas. Definición De armaduras.
- SAKA, M. P. & GEEM, Z. W. 2012. Mathematical and metaheuristic application in design optimization of steel frame structures: An extensive review. Available: <http://dx.doi.org/10.1155/2013/271031>.
- SALAZAR, J. J. 2003. Optimización Matemática: Ejemplos y aplicaciones *SCTM03* [Online]. Available: <http://webpages.ull.es/users/jjsalaza>.
- SANCHEZ, S., SELLES, M. A., PLA, R. & PEYDRÓ, M. 2012. Nuevas Técnicas de Optimización de estructuras. *Revista de Investigación*.
- SARKER, R., MOHAMMADIAN, M. & YAO, X. 2003. *Evolutionary Optimization*, ©2003 Kluwer Academic Publishers.
- SECK, J. C., MEDINA, J. & HERNÁNDEZ, N. 2016. Introducción a los Algoritmos Genéticos con Matlab.
- SMITH, J., HODGINS, J., OPPENHEIM, I. & WITKIN, A. 2002. Creating Models of Truss Structures with Optimization.
- SONMEZ, F. O. 2007. Shape optimization of 2d structures using simulated annealing. *ScienceDirect*, Computer methods in applied mechanics and engineering.
- SOUTO, A. *Tutorial de MATLAB*.
- TOLMOS, P. Introducción a los algoritmos genéticos y sus aplicaciones.
- VAMPLEW, P., DAZELEY, R., BERRY, A., ISSABEKOV, R. & DEKKER, E. 2010. Empirical evaluation methods for multiobjective reinforcement learning algorithms.
- VELÁZQUEZ, F. & SANTILLÁN, S. D. 2006. Diseño Ótimo Evolutivo. *SciELO Analytics*, Vol.7 No.3.
- VIDAL, A. 2013. *Algoritmos Heurísticos en optimización*. Tesis de máster, Universidad de Santiago de Compostela.

YANG, X.-S., KOZIEL, S. & LEIFSSON, L. 2014. Computational Optimization, Modelling and Simulation: Past, Present and Future *Procedia Computer Science*, Volume 29, 754-758.

ANEXOS

Anexo 1 Configuración de la Armadura Sloped

```
function y = Sloped(x)

q=20;

h=x(:,1);
l1=x(:,2); l2=x(:,3); l3=x(:,4); l4=x(:,5); l5=x(:,6); l6=x(:,7);
l7=x(:,8); l8=x(:,9); l9=x(:,10); l10=x(:,11); l11=x(:,12);
l12=x(:,13); l13=x(:,14); l14=x(:,15); l15=x(:,16); l16=x(:,17);
l17=x(:,18); l18=x(:,19); l19=x(:,20); l20=x(:,21); l21=x(:,22);
l22=x(:,23); l23=x(:,24); l24=x(:,25); l25=x(:,26);
d1=x(:,27); d2=x(:,28); d3=x(:,29); d4=x(:,30); d5=x(:,31); d6=x(:,32);
d7=x(:,33); d8=x(:,34); d9=x(:,35); d10=x(:,36); d11=x(:,37);
d12=x(:,38); d13=x(:,39); d14=x(:,40); d15=x(:,41); d16=x(:,42);
d17=x(:,43); d18=x(:,44); d19=x(:,45); d20=x(:,46); d21=x(:,47);
d22=x(:,48); d23=x(:,49); d24=x(:,50); d25=x(:,51); d26=x(:,52);
d27=x(:,53); d28=x(:,54); d29=x(:,55); d30=x(:,56); d31=x(:,57);
d32=x(:,58); d33=x(:,59); d34=x(:,60); d35=x(:,61); d36=x(:,62);
d37=x(:,63); d38=x(:,64); d39=x(:,65); d40=x(:,66); d41=x(:,67);
d42=x(:,68); d43=x(:,69); d44=x(:,70); d45=x(:,71); d46=x(:,72);
d47=x(:,73); d48=x(:,74); d49=x(:,75); d50=x(:,76);
x1=x(:,77); x2=x(:,78); x3=x(:,79); x4=x(:,80); x5=x(:,81); x6=x(:,82);
x7=x(:,83); x8=x(:,84); x9=x(:,85); x10=x(:,86); x11=x(:,87);
x12=x(:,88); x13=x(:,89); x14=x(:,90); x15=x(:,91); x16=x(:,92);
x17=x(:,93); x18=x(:,94); x19=x(:,95); x20=x(:,96); x21=x(:,97);
x22=x(:,98); x23=x(:,99); x24=x(:,100); x25=x(:,101); x26=x(:,102);
x27=x(:,103); x28=x(:,104); x29=x(:,105); x30=x(:,106); x31=x(:,107);
x32=x(:,108); x33=x(:,109); x34=x(:,110); x35=x(:,111); x36=x(:,112);
x37=x(:,113); x38=x(:,114); x39=x(:,115); x40=x(:,116); x41=x(:,117);
x42=x(:,118); x43=x(:,119); x44=x(:,120); x45=x(:,121); x46=x(:,122);
x47=x(:,123); x48=x(:,124); x49=x(:,125); x50=x(:,126);

l1= x1+x2
l2= x3+x4
l3= x5+x6
l4= x7+x8
l5= x9+x10
l6= x11+x12
l7= x13+x14
l8= x15+x16
l9= x17+x18
l10= x19+x20
l11= x21+x22
l12= x23+x24
l13= x25+x26
l14= x27+x28
l15= x29+x30
l16= x31+x32
l17= x33+x34
```

l18= x35+x36
 l19= x37+x38
 l20= x39+x40
 l21= x41+x42
 l22= x43+x44
 l23= x45+x46
 l24= x47+x48
 l25= x49+x50

P1=(l1/2)*q;
 P2=((l1+l2)/2)*q;
 P3=((l2+l3)/2)*q;
 P4=((l3+l4)/2)*q;
 P5=((l4+l5)/2)*q;
 P6=((l5+l6)/2)*q;
 P7=((l6+l7)/2)*q;
 P8=((l7+l8)/2)*q;
 P9=((l8+l9)/2)*q;
 P10=((l9+l10)/2)*q;
 P11=((l10+l11)/2)*q;
 P12=((l11+l12)/2)*q;
 P13=((l12+l13)/2)*q;
 P14=((l13+l14)/2)*q;
 P15=((l14+l15)/2)*q;
 P16=((l15+l16)/2)*q;
 P17=((l16+l17)/2)*q;
 P18=((l17+l18)/2)*q;
 P19=((l18+l19)/2)*q;
 P20=((l19+l20)/2)*q;
 P21=((l20+l21)/2)*q;
 P22=((l21+l22)/2)*q;
 P23=((l22+l23)/2)*q;
 P24=((l23+l24)/2)*q;
 P25=((l24+l25)/2)*q;
 P26=((l24+l25)/2)*q;

RA=P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P20+P21+P22+P23+P24+P25+P26;

FA1s=(-RA+P1)/(h/d1);
 FA1a=-FA1s*x1/d1;
 F1a1s=-FA1s*d2/d1;
 F1s2s=(FA1s*x1/d1)-F1a1s*x2/d2;
 F1a2s=(-F1a1s*d3/d2)+P2*d3/h;
 F1a2a=(-F1a2s*x3/d3)+F1a1s*x2/d2+FA1a;
 F2a2s=-F1a2s*d4/d3;
 F2s3s=(F1a2s*x3/d3)-F2a2s*x4/d4+F1s2s;
 F2a3s=(-F2a2s*d5/d4)+P3*d5/h;
 F2a3a=(-F2a3s*x5/d5)+F2a2s*x4/d4+F1a2a;
 F3a3s=-F2a3s*d6/d5;
 F3s4s=(F2a3s*x5/d5)-F3a3s*x6/d6+F2s3s;
 F3a4s=(-F3a3s*d7/d6)+P4*d7/h;
 F3a4a=(-F3a4s*x7/d7)+F3a3s*x6/d6+F2a3a;
 F4a4s=-F3a4s*d8/d7;
 F4s5s=(F3a4s*x7/d7)-F4a4s*x8/d8+F3s4s;

$F4a5s = (-F4a4s*d9/d8) + P5*d9/h;$
 $F4a5a = (-F4a5s*x9/d9) + F4a4s*x8/d8 + F3a4a;$
 $F5a5s = -F4a5s*d10/d9;$
 $F5s6s = (F4a5s*x9/d9) - F5a5s*x10/d10 + F4s5s;$
 $F5a6s = (-F5a5s*d11/d10) + P6*d11/h;$
 $F5a6a = (-F5a6s*x11/d11) + F5a5s*x10/d10 + F4a5a;$
 $F6a6s = -F5a6s*d12/d11;$
 $F6s7s = (F5a6s*x11/d11) - F6a6s*x12/d12 + F5s6s;$
 $F6a7s = (-F6a6s*d13/d12) + P7*d13/h;$
 $F6a7a = (-F6a7s*x13/d13) + F6a6s*x12/d12 + F5a6a;$
 $F7a7s = -F6a7s*d14/d13;$
 $F7s8s = (F6a7s*x13/d13) - F7a7s*x14/d14 + F6s7s;$
 $F7a8s = (-F7a7s*d15/d14) + P8*d15/h;$
 $F7a8a = (-F7a8s*x15/d15) + F7a7s*x14/d14 + F6a7a;$
 $F8a8s = -F7a8s*d16/d15;$
 $F8s9s = (F7a8s*x15/d15) - F8a8s*x16/d16 + F7s8s;$
 $F8a9s = (-F8a8s*d17/d16) + P9*d17/h;$
 $F8a9a = (-F8a9s*x17/d17) + F8a8s*x16/d16 + F7a8a;$
 $F9a9s = -F8a9s*d18/d17;$
 $F9s10s = (F8a9s*x17/d17) - F9a9s*x18/d18 + F8s9s;$
 $F9a10s = (-F9a9s*d19/d18) + P10*d19/h;$
 $F9a10a = (-F9a10s*x19/d19) + F9a9s*x18/d18 + F8a9a;$
 $F10a10s = -F9a10s*d20/d19;$
 $F10s11s = (F9a10s*x19/d19) - F10a10s*x20/d20 + F9s10s;$
 $F10a11s = (-F10a10s*d21/d20) + P11*d21/h;$
 $F10a11a = (-F10a11s*x21/d21) + F10a10s*x20/d20 + F9a10a;$
 $F11a11s = -F10a11s*d22/d21;$
 $F11s12s = (F10a11s*x21/d21) - F11a11s*x22/d22 + F10s11s;$
 $F11a12s = (-F11a11s*d23/d22) + P12*d23/h;$
 $F11a12a = (-F11a12s*x23/d23) + F11a11s*x22/d22 + F10a11a;$
 $F12a12s = -F11a12s*d24/d23;$
 $F12s13s = (F11a12s*x23/d23) - F12a12s*x24/d24 + F11s12s;$
 $F12a13s = (-F12a12s*d25/d24) + P13*d25/h;$
 $F12a13a = (-F12a13s*x25/d25) + F12a12s*x24/d24 + F11a12a;$
 $F13a13s = -F12a13s*d26/d25;$
 $F13s14s = (F12a13s*x25/d25) - F13a13s*x26/d26 + F12s13s;$
 $F13a14s = (-F13a13s*d27/d26) + P14*d27/h;$
 $F13a14a = (-F13a14s*x27/d27) + F13a13s*x26/d26 + F12a13a;$
 $F14a14s = -F13a14s*d28/d27;$
 $F14s15s = (F13a14s*x27/d27) - F14a14s*x28/d28 + F13s14s;$
 $F14a15s = (-F14a14s*d29/d28) + P15*d29/h;$
 $F14a15a = (-F14a15s*x29/d29) + F14a14s*x28/d28 + F13a14a;$
 $F15a15s = -F14a15s*d30/d29;$
 $F15s16s = (F14a15s*x29/d29) - F15a15s*x30/d30 + F14s15s;$
 $F15a16s = (-F15a15s*d31/d30) + P16*d31/h;$
 $F15a16a = (-F15a16s*x31/d31) + F15a15s*x30/d30 + F14a15a;$
 $F16a16s = -F15a16s*d32/d31;$
 $F16s17s = (F15a16s*x31/d31) - F16a16s*x32/d32 + F15s16s;$
 $F16a17s = (-F16a16s*d33/d32) + P17*d33/h;$
 $F16a17a = (-F16a17s*x33/d33) + F16a16s*x32/d32 + F15a16a;$
 $F17a17s = -F16a17s*d34/d33;$
 $F17s18s = (F16a17s*x33/d33) - F17a17s*x34/d34 + F16s17s;$
 $F17a18s = (-F17a17s*d35/d34) + P18*d35/h;$
 $F17a18a = (-F17a18s*x35/d35) + F17a17s*x34/d34 + F16a17a;$
 $F18a18s = -F17a18s*d36/d35;$
 $F18s19s = (F17a18s*x35/d35) - F18a18s*x36/d36 + F17s18s;$
 $F18a19s = (-F18a18s*d37/d36) + P19*d37/h;$

```

F18a19a=(-F18a19s*x37/d37)+F18a18s*x36/d36+ F17a18a;
F19a19s= -F18a19s*d38/d37;
F19s20s=(F18a19s*x37/d37)-F19a19s*x38/d38+ F18s19s;
F19a20s=(-F19a19s*d39/d38)+P20*d39/h;
F19a20a=(-F19a20s*x39/d39)+F19a19s*x38/d38+ F18a19a
F20a20s= -F19a20s*d40/d39;
F20s21s=(F19a20s*x39/d39)-F20a20s*x40/d40+ F19s20s;
F20a21s=(-F20a20s*d41/d40)+P21*d41/h;
F20a21a=(-F20a21s*x41/d41)+F20a20s*x40/d40+ F19a20a
F21a21s= -F20a21s*d42/d41;
F21s22s=(F20a21s*x41/d41)-F21a21s*x42/d42+ F20s21s;
F21a22s=(-F21a21s*d43/d42)+P22*d43/h;
F21a22a=(-F21a22s*x43/d43)+F21a21s*x42/d42+ F20a21a
F22a22s= -F21a22s*d44/d43;
F22s23s=(F21a22s*x43/d43)-F22a22s*x44/d44+ F21s22s;
F22a23s=(-F22a22s*d45/d44)+P23*d45/h;
F22a23a=(-F22a23s*x45/d45)+F22a22s*x44/d44+ F21a22a
F23a23s= -F22a23s*d46/d45;
F23s24s=(F22a23s*x45/d45)-F23a23s*x46/d46+ F22s23s;
F23a24s=(-F23a23s*d47/d46)+P25*d47/h;
F23a24a=(-F23a24s*x47/d47)+F23a23s*x46/d46+ F22a23a
F24a24s= -F23a24s*d48/d47;
F24s25s=(F23a24s*x47/d47)-F24a24s*x48/d48+ F23s24s;
F24a25s=(-F24a24s*d49/d48)+P26*d49/h;
F24a25a=(-F24a25s*x49/d49)+F24a24s*x48/d48+ F23a24a
F25a25s= -F24a25s*d50/d49;

```

```

y=max([FA1a; FA1s; FA1a; F1a1s; F1s2s; F1a2s; F1a2a; F2a2s; F2s3s;
F2a3s; F2a3a; F3a3s; F3s4s; F3a4s; F3a4a; F4a4s; F4s5s; F4a5s; F4a5a;
F5a5s; F5s6s; F5a6s; F5a6a; F6a6s; F6s7s; F6a7s; F6a7a; F7a7s; F7s8s;
F7a8s; F7a8a; F8a8s; F8s9s; F8a9s; F8a9a; F9a9s; F9s10s; F9a10s;
F9a10a; F10a10s; F10s11s; F10a11s; F10a11a; F11a11s; F11s12s; F11a12s;
F11a12a; F12a12s; F12s13s; F12a13s; F12a13a; F13a13s; F13s14s; F13a14s;
F13a14a; F14a14s; F14s15s; F14a15s; F14a15a; F15a15s; F15s16s; F15a16s;
F15a16a; F16a16s; F16s17s; F16a17s; F16a17a; F17a17s; F17s18s; F17a18s;
F17a18a; F18a18s; F18s19s; F18a19s; F18a19a; F19a19s; F19s20s; F19a20s;
F19a20a; F20a20s; F20s21s; F20a21s; F20a21a; F21a21s; F21s22s; F21a22s;
F21a22a; F22a22s; F22s23s; F22a23s; F22a23a; F23a23s; F23s24s; F23a24s;
F23a24a; F24a24s; F24s25s; F24a25s; F24a25a; F25a25s]);

```

Anexo 2 Configuración de la Armadura Pratt

```
function y = Pratt(x)

q=20;

l1=x(:,1);
l2=x(:,2);
l3=x(:,3); l4=x(:,4); l5=x(:,5); l6=x(:,6); l7=x(:,7); l8=x(:,8);
l9=x(:,9); l10=x(:,10); l11=x(:,11); l12=x(:,12); l13=x(:,13);
l14=x(:,14); l15=x(:,15); l16=x(:,16); l17=x(:,17); l18=x(:,18);
l19=x(:,19); l20=x(:,20); l21=x(:,21); l22=x(:,22); l23=x(:,23);
l24=x(:,24); l25=x(:,25); l26=x(:,26);
H1=x(:,27); H2=x(:,28); H3=x(:,29); H4=x(:,30); H5=x(:,31); H6=x(:,32);
H7=x(:,33); H8=x(:,34); H9=x(:,35); H10=x(:,36); H11=x(:,37);
H12=x(:,38); H13=x(:,39); H14=x(:,40); H15=x(:,41); H16=x(:,42);
H17=x(:,43); H18=x(:,44); H19=x(:,45); H20=x(:,46); H21=x(:,47);
H22=x(:,48); H23=x(:,49); H24=x(:,50); H25=x(:,51); H26=x(:,52);
H27=x(:,53);
d1=x(:,54); d2=x(:,55); d3=x(:,56); d4=x(:,57); d5=x(:,58); d6=x(:,59);
d7=x(:,60); d8=x(:,61); d9=x(:,62); d10=x(:,63); d11=x(:,64);
d12=x(:,65); d13=x(:,66); d14=x(:,67); d15=x(:,68); d16=x(:,69);
d17=x(:,70); d18=x(:,71); d19=x(:,72); d20=x(:,73); d21=x(:,74);
d22=x(:,75); d23=x(:,76); d24=x(:,77); d25=x(:,78); d26=x(:,79);
b1=x(:,80); b2=x(:,81); b3=x(:,82); b4=x(:,83); b5=x(:,84); b6=x(:,85);
b7=x(:,86); b8=x(:,87); b9=x(:,88); b10=x(:,89); b11=x(:,90);
b12=x(:,91); b13=x(:,92); b14=x(:,93); b15=x(:,94); b16=x(:,95);
b17=x(:,96); b18=x(:,97); b19=x(:,98); b20=x(:,99); b21=x(:,100);
b22=x(:,101); b23=x(:,102); b24=x(:,103); b25=x(:,104); b26=x(:,105);

c1=H2-H1
c2=H3-H2
c3=H4-H3
c4=H5-H4
c5=H6-H5
c6=H7-H6
c7=H8-H7
c8=H9-H8
c9=H10-H9
c10=H11-H10
c11=H12-H11
c12=H13-H12
c13=H14-H13
c14=H15-H14
c15=H16-H15
c16=H17-H16
c17=H18-H17
c18=H19-H18
c19=H20-H19
c20=H21-H20
c21=H22-H21
c22=H23-H22
c23=H24-H23
```


c24=H25-H24
c25=H26-H25
c26=H27-H26

P1=(11/2)*q;
P2=((11+12)/2)*q;
P3=((12+13)/2)*q;
P4=((13+14)/2)*q;
P5=((14+15)/2)*q;
P6=((15+16)/2)*q;
P7=((16+17)/2)*q;
P8=((17+18)/2)*q;
P9=((18+19)/2)*q;
P10=((19+110)/2)*q;
P11=((110+111)/2)*q;
P12=((111+112)/2)*q;
P13=((112+113)/2)*q;
P14=((113+114)/2)*q;
P15=((114+115)/2)*q;
P16=((115+116)/2)*q;
P17=((116+117)/2)*q;
P18=((117+118)/2)*q;
P19=((118+119)/2)*q;
P20=((119+120)/2)*q;
P21=((120+121)/2)*q;
P22=((120+121)/2)*q;
P23=((121+122)/2)*q;
P24=((122+123)/2)*q;
P25=((123+124)/2)*q;
P26=((124+125)/2)*q;

RA=P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P20+P21+P22+P23+P24+P25+P26;

FA2s=(P1-RA)/(d1/H2);
FA1a=-FA2s/(11/d1);
FA1s=P1-RA-FA2s*H2/d1;
F1s2s=0;
F2s3s=(P1-RA)*11*b2/(H2*12);
F1a3s=(-F2s3s*(c2/b2)-RA+P1+P2)/(d2/H3);
F1a2a=-F1a3s*(12/d2)-F2s3s*12/b2;
F1a2s=P2-F1a3s*H3/d2;
F3s4s=(P1-RA)*(11+12)+P2*12*b3/(H3*13);
F2a4s=(-F3s4s*(c3/b3)-RA+P1+P2+P3)/(d3/H4);
F2a3a=-F2a4s*(13/d3)-F3s4s*13/b3;
F2a3s=P3-F2a4s*H4/d3;
F4s5s=(P1-RA)*(11+12+13)+P2*(12+13)+P3*13*b4/(H4*14);
F3a5s=(-F4s5s*(c4/b4)-RA+P1+P2+P3+P4)/(d4/H5);
F3a4a=-F3a5s*(14/d4)-F4s5s*14/b4;
F3a4s=P4-F3a5s*H5/d4;
F5s6s=(P1-RA)*(11+12+13+14)+P2*(12+13+14)+P3*(13+14)+P4*14*b5/(H5*15);
F4a6s=(-F5s6s*(c5/b5)-RA+P1+P2+P3+P4+P5)/(d5/H6);

$F4a5a = -F4a6s * (15/d5) - F5s6s * 15/b5;$
 $F4a5s = P5 - F4a6s * H6/d5;$
 $F6s7s = (P1 -$
 $RA) * (11+12+13+14+15) + P2 * (12+13+14+15) + P3 * (13+14+15) + P4 * (14+15) + P5 * 15) * b$
 $6 / (H6 * 16);$
 $F5a7s = (-F6s7s * (c6/b6) - RA + P1 + P2 + P3 + P4 + P5 + P6) / (d6 / H7);$
 $F5a6a = -F5a7s * (16/d6) - F6s7s * 16/b6;$
 $F5a6s = P6 - F5a7s * H7/d6;$
 $F7s8s = (P1 -$
 $RA) * (11+12+13+14+15+16) + P2 * (12+13+14+15+16) + P3 * (13+14+15+16) + P4 * (14+15+$
 $16) + P5 * (15+16) + P6 * 16) * b7 / (H7 * 17);$
 $F6a8s = (-F7s8s * (c7/b7) - RA + P1 + P2 + P3 + P4 + P5 + P6 + P7) / (d7 / H8);$
 $F6a7a = -F6a8s * (17/d7) - F7s8s * 17/b7;$
 $F6a7s = P7 - F6a8s * H8/d7;$
 $F8s9s = (P1 -$
 $RA) * (11+12+13+14+15+16+17) + P2 * (12+13+14+15+16+17) + P3 * (13+14+15+16+17) + P$
 $4 * (14+15+16+17) + P5 * (15+16+17) + P6 * (16+17) + P7 * 17) * b8 / (H8 * 18);$
 $F7a9s = (-F8s9s * (c8/b8) - RA + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8) / (d8 / H9);$
 $F7a8a = -F7a9s * (18/d8) - F8s9s * 18/b8;$
 $F7a8s = P8 - F7a9s * H9/d8;$
 $F9s10s = (P1 -$
 $RA) * (11+12+13+14+15+16+17+18) + P2 * (12+13+14+15+16+17+18) + P3 * (13+14+15+16$
 $+17+18) + P4 * (14+15+16+17+18) + P5 * (15+16+17+18) + P6 * (16+17+18) + P7 * (17+18) + P$
 $8 * 18) * b9 / (H9 * 19);$
 $F8a10s = (-F9s10s * (c9/b9) - RA + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9) / (d9 / H10);$
 $F8a9a = -F8a10s * (19/d9) - F9s10s * 19/b9;$
 $F8a9s = P9 - F8a10s * H10/d9;$
 $F10s11s = (P1 -$
 $RA) * (11+12+13+14+15+16+17+18+19) + P2 * (12+13+14+15+16+17+18+19) + P3 * (13+14$
 $+15+16+17+18+19) + P4 * (14+15+16+17+18+19) + P5 * (15+16+17+18+19) + P6 * (16+17+1$
 $8+19) + P7 * (17+18+19) + P8 * (18+19) + P9 * 19) * b10 / (H10 * 110);$
 $F9a11s = (-F10s11s * (c10/b10) - RA + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9 + P10) / (d10 /$
 $H11);$
 $F9a10a = -F9a11s * (110/d10) - F10s11s * 110/b10;$
 $F9a10s = P10 - F9a11s * H11/d10;$
 $F11s12s = (P1 -$
 $RA) * (11+12+13+14+15+16+17+18+19+110) + P2 * (12+13+14+15+16+17+18+19+110) + P$
 $3 * (13+14+15+16+17+18+19+110) + P4 * (14+15+16+17+18+19+110) + P5 * (15+16+17+18$
 $+19+110) + P6 * (16+17+18+19+110) + P7 * (17+18+19+110) + P8 * (18+19+110) + P9 * (19+1$
 $10) + P10 * 110) * b11 / (H11 * 111);$
 $F10a12s = (-F11s12s * (c11/b11) - RA + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9 + P10 + P11) / ($
 $d11 / H12);$
 $F10a11a = -F10a12s * (111/d11) - F11s12s * 111/b11;$
 $F10a11s = P11 - F10a12s * H12/d11;$
 $F12s13s = (P1 -$
 $RA) * (11+12+13+14+15+16+17+18+19+110+111) + P2 * (12+13+14+15+16+17+18+19+11$
 $0+111) + P3 * (13+14+15+16+17+18+19+110+111) + P4 * (14+15+16+17+18+19+110+111)$
 $+ P5 * (15+16+17+18+19+110+111) + P6 * (16+17+18+19+110+111) + P7 * (17+18+19+110+$
 $111) + P8 * (18+19+110+111) + P9 * (19+110+111) + P10 * (110+111) + P11 * 111) * b12 / (H12$
 $* 112);$
 $F11a13s = (-F12s13s * (c12/b12) -$
 $RA + P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9 + P10 + P11 + P12) / (d12 / H13);$
 $F11a12a = -F11a13s * (112/d12) - F12s13s * 112/b12;$
 $F11a12s = P12 - F11a13s * H13/d12;$
 $F13s14s = (P1 -$
 $RA) * (11+12+13+14+15+16+17+18+19+110+111+112) + P2 * (12+13+14+15+16+17+18+1$
 $9+110+111+112) + P3 * (13+14+15+16+17+18+19+110+111+112) + P4 * (14+15+16+17+18$

$+19+110+111+112)+P5*(15+16+17+18+19+110+111+112)+P6*(16+17+18+19+110+111+112)+P7*(17+18+19+110+111+112)+P8*(18+19+110+111+112)+P9*(19+110+111+112)+P10*(110+111+112)+P11*(111+112)+P12*112)*b1./(H13*113);$
 $F12a14s=(-F13s14s*(c13/b13)-$
 $RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13)/(d13/H14);$
 $F12a13a=-F12a14s*(113/d13)-F13s14s*113/b13;$
 $F12a13s=P13-F12a14s*H14/d13;$
 $F14s15s=((P1-$
 $RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113)+P2*(12+13+14+15+16+17+18+19+110+111+112+113)+P3*(13+14+15+16+17+18+19+110+111+112+113)+P4*(14+15+16+17+18+19+110+111+112+113)+P5*(15+16+17+18+19+110+111+112+113)+P6*(16+17+18+19+110+111+112+113)+P7*(17+18+19+110+111+112+113)+P8*(18+19+110+111+112+113)+P9*(19+110+111+112+113)+P10*(110+111+112+113)+P11*(111+112+113)+P12*(112+113)+P13*113)*b14/(H14*114);$
 $F13a15s=(-F14s15s*(c14/b14)-$
 $RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14)/(d14/H15);$
 $F13a14a=-F13a15s*(114/d14)-F14s15s*114/b14;$
 $F13a14s=P14-F13a15s*H15/d14;$
 $F15s16s=((P1-$
 $RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114)+P3*(13+14+15+16+17+18+19+110+111+112+113+114)+P4*(14+15+16+17+18+19+110+111+112+113+114)+P5*(15+16+17+18+19+110+111+112+113+114)+P6*(16+17+18+19+110+111+112+113+114)+P7*(17+18+19+110+111+112+113+114)+P8*(18+19+110+111+112+113+114)+P9*(19+110+111+112+113+114)+P10*(110+111+112+113+114)+P11*(111+112+113+114)+P12*(112+113+114)+P13*(113+114)+P14*114)*b15/(H15*115);$
 $F14a16s=(-F15s16s*(c15/b15)-$
 $RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15)/(d15/H16);$
 $F14a15a=-F14a16s*(115/d15)-F15s16s*115/b15;$
 $F14a15s=P15-F14a16s*H16/d15;$
 $F16s17s=((P1-$
 $RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115)+P3*(13+14+15+16+17+18+19+110+111+112+113+114+115)+P4*(14+15+16+17+18+19+110+111+112+113+114+115)+P5*(15+16+17+18+19+110+111+112+113+114+115)+P6*(16+17+18+19+110+111+112+113+114+115)+P7*(17+18+19+110+111+112+113+114+115)+P8*(18+19+110+111+112+113+114+115)+P9*(19+110+111+112+113+114+115)+P10*(110+111+112+113+114+115)+P11*(111+112+113+114+115)+P12*(112+113+114+115)+P13*(113+114+115)+P14*(114+115)+P15*115)*b16/(H16*116);$
 $F15a17s=(-F16s17s*(c16/b16)-$
 $RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16)/(d16/H17);$
 $F15a16a=-F15a17s*(116/d16)-F16s17s*116/b16;$
 $F15a16s=P16-F15a17s*H17/d16;$
 $F17s18s=((P1-$
 $RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+116)+P3*(13+14+15+16+17+18+19+110+111+112+113+114+115+116)+P4*(14+15+16+17+18+19+110+111+112+113+114+115+116)+P5*(15+16+17+18+19+110+111+112+113+114+115+116)+P6*(16+17+18+19+110+111+112+113+114+115+116)+P7*(17+18+19+110+111+112+113+114+115+116)+P8*(18+19+110+111+112+113+114+115+116)+P9*(19+110+111+112+113+114+115+116)+P10*(110+111+112+113+114+115+116)+P11*(111+112+113+114+115+116)+P12*(112+113+114+115+116)+P13*(113+114+115+116)+P14*(114+115+116)+P15*(115+116)+P16*116)*b17/(H17*117);$
 $F16a18s=(-F17s18s*(c17/b17)-$
 $RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17)/(d17/H18);$

```

F16a17a=-F16a18s*(117/d17)-F17s18s*117/b17;
F16a17s=P17-F16a18s*H18/d17;
F18s19s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117)+P2*(12
+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117)+P3*(13+14+15+16+
17+18+19+110+111+112+113+114+115+116+117)+P4*(14+15+16+17+18+19+110+111
+112+113+114+115+116+117)+P5*(15+16+17+18+19+110+111+112+113+114+115+11
6+117)+P6*(16+17+18+19+110+111+112+113+114+115+116+117)+P7*(17+18+19+11
0+111+112+113+114+115+116+117)+P8*(18+19+110+111+112+113+114+115+116+11
7)+P9*(19+110+111+112+113+114+115+116+117)+P10*(110+111+112+113+114+115
+116+117)+P11*(111+112+113+114+115+116+117)+P12*(112+113+114+115+116+11
7)+P13*(113+114+115+116+117)+P14*(114+115+116+117)+P15*(115+116+117)+P1
6*(116+117)+P17*117)*b18/(H18*118);
F17a19s=(-F18s19s*(c18/b18)-
RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18)/(
d18 / H19);
F17a18a=-F17a19s*(118/d18)-F18s19s*118/b18;
F17a18s=P18-F17a19s*H19/d18;
F19s20s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118)+P2
*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118)+P3*(13+1
4+15+16+17+18+19+110+111+112+113+114+115+116+117+118)+P4*(14+15+16+17+1
8+19+110+111+112+113+114+115+116+117+118)+P5*(15+16+17+18+19+110+111+11
2+113+114+115+116+117+118)+P6*(16+17+18+19+110+111+112+113+114+115+116+
117+118)+P7*(17+18+19+110+111+112+113+114+115+116+117+118)+P8*(18+19+11
0+111+112+113+114+115+116+117+118)+P9*(19+110+111+112+113+114+115+116+1
17+118)+P10*(110+111+112+113+114+115+116+117+118)+P11*(111+112+113+114+
115+116+117+118)+P12*(112+113+114+115+116+117+118)+P13*(113+114+115+116
+117+118)+P14*(114+115+116+117+118)+P15*(115+116+117+118)+P16*(116+117+
118)+P17*(117+118)+P18*118)*b19/(H19*119);
F18a20s=(-F19s20s*(c19/b19)-
RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19)/(
d19 / H20);
F18a19a=-F18a20s*(119/d19)-F19s20s*119/b19;
F18a19s=P19-F18a20s*H20/d19;
F20s21s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119
)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119)+
P3*(13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119)+P4*(1
4+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119)+P5*(15+16+17+
18+19+110+111+112+113+114+115+116+117+118+119)+P6*(16+17+18+19+110+111+
112+113+114+115+116+117+118+119)+P7*(17+18+19+110+111+112+113+114+115+1
16+117+118+119)+P8*(18+19+110+111+112+113+114+115+116+117+118+119)+P9*(
19+110+111+112+113+114+115+116+117+118+119)+P10*(110+111+112+113+114+11
5+116+117+118+119)+P11*(111+112+113+114+115+116+117+118+119)+P12*(112+1
13+114+115+116+117+118+119)+P13*(113+114+115+116+117+118+119)+P14*(114+
115+116+117+118+119)+P15*(115+116+117+118+119)+P16*(116+117+118+119)+P1
7*(117+118+119)+P18*(118+119)+P19*119)*b20/(H20*120);
F19a21s=(-F20s21s*(c20/b20)-
RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P
20)/(d20 / H21);
F19a20a=-F19a21s*(120/d20)-F20s21s*120/b20;
F19a20s=P20-F19a21s*H21/d20;
F21s22s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119
+120)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+1
19+120)+P3*(13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+11

```

$9+120)+P4*(14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120)+P5*(15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120)+P6*(16+17+18+19+110+111+112+113+114+115+116+117+118+119+120)+P7*(17+118+19+110+111+112+113+114+115+116+117+118+119+120)+P8*(18+19+110+111+112+113+114+115+116+117+118+119+120)+P9*(19+110+111+112+113+114+115+116+117+118+119+120)+P10*(110+111+112+113+114+115+116+117+118+119+120)+P11*(111+112+113+114+115+116+117+118+119+120)+P12*(112+113+114+115+116+117+118+119+120)+P13*(113+114+115+116+117+118+119+120)+P14*(114+115+116+117+118+119+120)+P15*(115+116+117+118+119+120)+P16*(116+117+118+119+120)+P17*(117+118+119+120)+P18*(118+119+120)+P19*(119+120)+P20*120)*b21/(H21*121);$
 $F20a22s=(-F21s22s*(c21/b21)-RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P20+P21)/(d21/H22);$
 $F20a21a=-F20a22s*(121/d21)-F21s22s*121/b21;$
 $F20a21s=P21-F20a22s*H22/d21;$
 $F22s23s=((P1-RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+18+119+120+121)+P3*(13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121)+P4*(14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121)+P5*(15+16+17+18+19+110+111+112+113+114+115+116+117+18+119+120+121)+P6*(16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121)+P7*(17+18+19+110+111+112+113+114+115+116+117+118+119+120+121)+P8*(18+19+110+111+112+113+114+115+116+117+118+119+120+121)+P9*(19+110+111+112+113+114+115+116+117+118+119+120+121)+P10*(110+111+112+113+114+115+116+117+118+119+120+121)+P11*(111+112+113+114+115+116+117+118+119+120+121)+P12*(112+113+114+115+116+117+118+119+120+121)+P13*(113+114+115+116+117+118+119+120+121)+P14*(114+115+116+117+118+119+120+121)+P15*(115+116+117+118+119+120+121)+P16*(116+117+118+119+120+121)+P17*(117+118+119+120+121)+P18*(118+119+120+121)+P19*(119+120+121)+P20*(120+121)+P21*121)*b22/(H22*122);$
 $F21a23s=(-F22s23s*(c22/b22)-RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P20+P21+P22)/(d22/H23);$
 $F21a22a=-F21a23s*(122/d22)-F22s23s*122/b22;$
 $F21a22s=P22-F21a23s*H23/d22;$
 $F23s24s=((P1-RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+17+118+119+120+121+122)+P3*(13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P4*(14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P5*(15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P6*(16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P7*(17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P8*(18+19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P9*(19+110+111+112+113+114+115+116+117+118+119+120+121+122)+P10*(110+111+112+113+114+115+116+117+118+119+120+121+122)+P11*(111+112+113+114+115+116+117+118+119+120+121+122)+P12*(112+113+114+115+116+117+118+119+120+121+122)+P13*(113+114+115+116+117+118+119+120+121+122)+P14*(114+115+116+117+118+119+120+121+122)+P15*(115+116+117+118+119+120+121+122)+P16*(116+117+118+119+120+121+122)+P17*(117+118+119+120+121+122)+P18*(118+119+120+121+122)+P19*(119+120+121+122)+P20*(120+121+122)+P21*(121+122)+P22*122)*b23/(H23*123);$
 $F22a24s=(-F23s24s*(c23/b23)-RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P20+P21+P22+P23)/(d23/H24);$
 $F22a23a=-F22a24s*(123/d23)-F23s24s*123/b23;$

```

F22a23s=P23-F22a24s*H24/d23;
F24s25s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119
+120+121+122+123)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+115+1
16+117+118+119+120+121+122+123)+P3*(13+14+15+16+17+18+19+110+111+112+11
3+114+115+116+117+118+119+120+121+122+123)+P4*(14+15+16+17+18+19+110+11
1+112+113+114+115+116+117+118+119+120+121+122+123)+P5*(15+16+17+18+19+1
10+111+112+113+114+115+116+117+118+119+120+121+122+123)+P6*(16+17+18+19
+110+111+112+113+114+115+116+117+118+119+120+121+122+123)+P7*(17+18+19+
110+111+112+113+114+115+116+117+118+119+120+121+122+123)+P8*(18+19+110+
111+112+113+114+115+116+117+118+119+120+121+122+123)+P9*(19+110+111+112
+113+114+115+116+117+118+119+120+121+122+123)+P10*(110+111+112+113+114+
115+116+117+118+119+120+121+122+123)+P11*(111+112+113+114+115+116+117+1
18+119+120+121+122+123)+P12*(112+113+114+115+116+117+118+119+120+121+12
2+123)+P13*(113+114+115+116+117+118+119+120+121+122+123)+P14*(114+115+1
16+117+118+119+120+121+122+123)+P15*(115+116+117+118+119+120+121+122+12
3)+P16*(116+117+118+119+120+121+122+123)+P17*(117+118+119+120+121+122+1
23)+P18*(118+119+120+121+122+123)+P19*(119+120+121+122+123)+P20*(120+12
1+122+123)+P21*(121+122+123)+P22*(122+123)+P23*123)*b24/(H24*124);
F23a25s=(-F24s25s*(c24/b24)-
RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P
20+P21+P22+P23+P24)/(d24/H25);
F23a24a=-F23a25s*(124/d24)-F24s25s*124/b24;
F23a24s=P24-F23a25s*H25/d24;
F25s26s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119
+120+121+122+123+124)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+114+1
15+116+117+118+119+120+121+122+123+124)+P3*(13+14+15+16+17+18+19+110+11
1+112+113+114+115+116+117+118+119+120+121+122+123+124)+P4*(14+15+16+17+
18+19+110+111+112+113+114+115+116+117+118+119+120+121+122+123+124)+P5*(
15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122+123+
124)+P6*(16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+12
2+123+124)+P7*(17+18+19+110+111+112+113+114+115+116+117+118+119+120+121
+122+123+124)+P8*(18+19+110+111+112+113+114+115+116+117+118+119+120+121
+122+123+124)+P9*(19+110+111+112+113+114+115+116+117+118+119+120+121+12
2+123+124)+P10*(110+111+112+113+114+115+116+117+118+119+120+121+122+123
+124)+P11*(111+112+113+114+115+116+117+118+119+120+121+122+123+124)+P12
*(112+113+114+115+116+117+118+119+120+121+122+123+124)+P13*(113+114+115
+116+117+118+119+120+121+122+123+124)+P14*(114+115+116+117+118+119+120+
121+122+123+124)+P15*(115+116+117+118+119+120+121+122+123+124)+P16*(116
+117+118+119+120+121+122+123+124)+P17*(117+118+119+120+121+122+123+124)
+P18*(118+119+120+121+122+123+124)+P19*(119+120+121+122+123+124)+P20*(1
20+121+122+123+124)+P21*(121+122+123+124)+P22*(122+123+124)+P23*(123+12
4)+P24*124)*b25/(H25*125);
F24a26s=(-F25s26s*(c25/b25)-
RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P
20+P21+P22+P23+P24+P25)/(d25/H26);
F24a25a=-F24a26s*(125/d25)-F25s26s*125/b25;
F24a25s=P25-F24a26s*H26/d25;
F26s27s=((P1-
RA)*(11+12+13+14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119
+120+121+122+123+124+125)+P2*(12+13+14+15+16+17+18+19+110+111+112+113+1
14+115+116+117+118+119+120+121+122+123+124+125)+P3*(13+14+15+16+17+18+1
9+110+111+112+113+114+115+116+117+118+119+120+121+122+123+124+125)+P4*(
14+15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+120+121+122+1
23+124+125)+P5*(15+16+17+18+19+110+111+112+113+114+115+116+117+118+119+
120+121+122+123+124+125)+P6*(16+17+18+19+110+111+112+113+114+115+116+11

```

```

7+118+119+120+121+122+123+124+125)+P7*(17+18+19+110+111+112+113+114+115
+116+117+118+119+120+121+122+123+124+125)+P8*(18+19+110+111+112+113+114
+115+116+117+118+119+120+121+122+123+124+125)+P9*(19+110+111+112+113+11
4+115+116+117+118+119+120+121+122+123+124+125)+P10*(110+111+112+113+114
+115+116+117+118+119+120+121+122+123+124+125)+P11*(111+112+113+114+115+
116+117+118+119+120+121+122+123+124+125)+P12*(112+113+114+115+116+117+1
18+119+120+121+122+123+124+125)+P13*(113+114+115+116+117+118+119+120+12
1+122+123+124+125)+P14*(114+115+116+117+118+119+120+121+122+123+124+125
)+P15*(115+116+117+118+119+120+121+122+123+124+125)+P16*(116+117+118+11
9+120+121+122+123+124+125)+P17*(117+118+119+120+121+122+123+124+125)+P1
8*(118+119+120+121+122+123+124+125)+P19*(119+120+121+122+123+124+125)+P
20*(120+121+122+123+124+125)+P21*(121+122+123+124+125)+P22*(122+123+124
+125)+P23*(123+124+125)+P24*(124+125)+P25*125)*b26/(H26*126);
F25a27s=(-F26s27s*(c26/b26)-
RA+P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P
20+P21+P22+P23+P24+P25+P26)/(d26/H27);
F25a26a=-F25a27s*(126/d26)-F26s27s*126/b26;
F25a26s=P26-F25a27s*H27/d26;

```

```

y=max([FA1a; FA2s; FA1s; F1s2s; F2s3s; F1a3s; F1a2a; F1a2s; F3s4s;
F2a4s; F2a3a; F2a3s; F4s5s; F3a5s; F3a4a; F3a4s; F5s6s; F4a6s; F4a5a;
F4a5s; F6s7s; F5a7s; F5a6a; F5a6s; F7s8s; F6a8s; F6a7a; F6a7s; F8s9s;
F7a9s; F7a8a; F9s10s; F8a10s; F8a9a; F8a9s; F10s11s; F9a11s; F9a10a;
F9a10s; F11s12s; F10a12s; F10a11a; F10a11s; F12s13s; F11a13s; F11a12a;
F11a12s; F13s14s; F12a13a; F12a13s; F14s15s; F13a15s; F13a14a; F13a14s;
F15s16s; F14a16s; F14a15a; F14a15s; F16s17s; F15a17s; F15a16a; F15a16s;
F17s18s; F16a18s; F16a17a; F16a17s; F18s19s; F17a19s; F17a18a; F17a18s;
F19s20s; F18a20s; F18a19a; F18a19s; F20s21s; F19a21s; F19a20a; F19a20s;
F21s22s; F20a22s; F20a21a; F20a21s; F22s23s; F21a23s; F21a22a; F21a22s;
F23s24s; F22a24s; F22a23a; F22a23s; F24s25s; F23a25s; F23a24a; F23a24s;
F25s26s; F24a25a; F24a25s; F26s27s; F25a26a; F25a26s]);

```

Anexo 3 Configuración de la Armadura Vertical

```
function y = Vertical(x)

q=20;

H=x(:,1);
l1=x(:,2); l2=x(:,3); l3=x(:,4); l4=x(:,5); l5=x(:,6); l6=x(:,7);
l7=x(:,8); l8=x(:,9); l9=x(:,10); l10=x(:,11); l11=x(:,12);
l12=x(:,13); l13=x(:,14); l14=x(:,15); l15=x(:,16); l16=x(:,17);
l17=x(:,18); l18=x(:,19); l19=x(:,20); l20=x(:,21); l21=x(:,22);
l22=x(:,23); l23=x(:,24); l24=x(:,25); l25=x(:,26);
d1=x(:,27); d2=x(:,28); d3=x(:,29); d4=x(:,30); d5=x(:,31); d6=x(:,32);
d7=x(:,33); d8=x(:,34); d9=x(:,35); d10=x(:,36); d11=x(:,37);
d12=x(:,38); d13=x(:,39); d14=x(:,40); d15=x(:,41); d16=x(:,42);
d17=x(:,43); d18=x(:,44); d19=x(:,45); d20=x(:,46); d21=x(:,47);
d22=x(:,48); d23=x(:,49); d24=x(:,50); d25=x(:,51); d26=x(:,52);
d27=x(:,53); d28=x(:,54); d29=x(:,55); d30=x(:,56); d31=x(:,57);
d32=x(:,58); d33=x(:,59); d34=x(:,60); d35=x(:,61); d36=x(:,62);
d37=x(:,63); d38=x(:,64); d39=x(:,65); d40=x(:,66); d41=x(:,67);
d42=x(:,68); d43=x(:,69); d44=x(:,70); d45=x(:,71); d46=x(:,72);
d47=x(:,73); d48=x(:,74); d49=x(:,75); d50=x(:,76);
x1=x(:,77)

P1=(l1/2)*q;
P2=((l1+l2)/2)*q;
P3=((l2+l3)/2)*q;
P4=((l3+l4)/2)*q;
P5=((l4+l5)/2)*q;
P6=((l5+l6)/2)*q;
P7=((l6+l7)/2)*q;
P8=((l7+l8)/2)*q;
P9=((l8+l9)/2)*q;
P10=((l9+l10)/2)*q;
P11=((l10+l11)/2)*q;
P12=((l11+l12)/2)*q;
P13=((l12+l13)/2)*q;
P14=((l13+l14)/2)*q;
P15=((l14+l15)/2)*q;
P16=((l15+l16)/2)*q;
P17=((l16+l17)/2)*q;
P18=((l17+l18)/2)*q;
P19=((l18+l19)/2)*q;
P20=((l19+l20)/2)*q;
P21=((l20+l21)/2)*q;
P22=((l21+l22)/2)*q;
P23=((l22+l23)/2)*q;
P24=((l23+l24)/2)*q;
P25=((l24+l25)/2)*q;

RA=P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11+P12+P13+P14+P15+P16+P17+P18+P19+P
20+P21+P22+P23+P24+P25;
```



```

F1s1a=(d2*(RA*d1 - P1*d1 + H*x))/(H*d1)
F1s2s=- (l1*(RA*d1 - P1*d1 + H*x))/(H*d1)
FA2s=x
FA1a=- (l1*x)/d1
FA1s=- (RA*d1 - P1*d1 + H*x)/d1
F2s2a=- (d4*(RA*l1 - P1*l1 + H*x))/(H*l2)
F2s3s=x
F1a3s=(d3*(P1*l1 + P1*l2 + P2*l2 - RA*l1 - RA*l2 - H*x))/(H*l2)
F1a2a=- (2*P1*l1 + P1*l2 + P2*l2 - 2*RA*l1 - RA*l2 - H*x)/H
F3s3a=(d6*(P1*l1 + P1*l2 + P2*l2 - RA*l1 - RA*l2 - H*x))/(H*l3)
F3s4s=x
F2a4s=(d5*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P2*l3 + P3*l3 - RA*l1 -
RA*l2 - RA*l3 - H*x))/(H*l3)
F2a3a=- (2*P1*l1 + 2*P1*l2 + P1*l3 + 2*P2*l2 + P2*l3 + P3*l3 - 2*RA*l1 -
2*RA*l2 - RA*l3 - H*x)/H
F4s4a=(d8*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P2*l3 + P3*l3 - RA*l1 -
RA*l2 - RA*l3 - H*x))/(H*l4)
F4s5s=x
F3a5s=(d7*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P2*l4 +
P3*l3 + P3*l4 + P4*l4 - RA*l1 - RA*l2 - RA*l3 - RA*l4 - H*x))/(H*l4)
F3a4a=- (2*P1*l1 + 2*P1*l2 + 2*P1*l3 + 2*P2*l2 + P1*l4 + 2*P2*l3 + P2*l4
+ 2*P3*l3 + P3*l4 + P4*l4 - 2*RA*l1 - 2*RA*l2 - 2*RA*l3 - RA*l4 -
H*x)/H
F5s5a=(d10*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P2*l4 +
P3*l3 + P3*l4 + P4*l4 - RA*l1 - RA*l2 - RA*l3 - RA*l4 - H*x))/(H*l5)
F5s6s=x
F4a6s=(d9*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P1*l5 +
P2*l4 + P3*l3 + P2*l5 + P3*l4 + P3*l5 + P4*l4 + P4*l5 + P5*l5 - RA*l1 -
RA*l2 - RA*l3 - RA*l4 - RA*l5 - H*x))/(H*l5)
F4a5a=- (2*P1*l1 + 2*P1*l2 + 2*P1*l3 + 2*P2*l2 + 2*P1*l4 + 2*P2*l3 +
P1*l5 + 2*P2*l4 + 2*P3*l3 + P2*l5 + 2*P3*l4 + P3*l5 + 2*P4*l4 + P4*l5 +
P5*l5 - 2*RA*l1 - 2*RA*l2 - 2*RA*l3 - 2*RA*l4 - RA*l5 - H*x)/H
F6s6a=(d12*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P1*l5 +
P2*l4 + P3*l3 + P2*l5 + P3*l4 + P3*l5 + P4*l4 + P4*l5 + P5*l5 - RA*l1 -
RA*l2 - RA*l3 - RA*l4 - RA*l5 - H*x))/(H*l6)
F6s7s=x
F5a7s=(d11*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P1*l5 +
P2*l4 + P3*l3 + P1*l6 + P2*l5 + P3*l4 + P2*l6 + P3*l5 + P4*l4 + P3*l6 +
P4*l5 + P4*l6 + P5*l5 + P5*l6 + P6*l6 - RA*l1 - RA*l2 - RA*l3 - RA*l4 -
RA*l5 - RA*l6 - H*x))/(H*l6)
F5a6a=- (2*P1*l1 + 2*P1*l2 + 2*P1*l3 + 2*P2*l2 + 2*P1*l4 + 2*P2*l3 +
2*P1*l5 + 2*P2*l4 + 2*P3*l3 + P1*l6 + 2*P2*l5 + 2*P3*l4 + P2*l6 +
2*P3*l5 + 2*P4*l4 + P3*l6 + 2*P4*l5 + P4*l6 + 2*P5*l5 + P5*l6 + P6*l6 -
2*RA*l1 - 2*RA*l2 - 2*RA*l3 - 2*RA*l4 - 2*RA*l5 - RA*l6 - H*x)/H
F7s7a=(d14*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P1*l5 +
P2*l4 + P3*l3 + P1*l6 + P2*l5 + P3*l4 + P2*l6 + P3*l5 + P4*l4 + P3*l6 +
P4*l5 + P4*l6 + P5*l5 + P5*l6 + P6*l6 - RA*l1 - RA*l2 - RA*l3 - RA*l4 -
RA*l5 - RA*l6 - H*x))/(H*l7)
F7s8s=x
F6a8s=(d13*(P1*l1 + P1*l2 + P1*l3 + P2*l2 + P1*l4 + P2*l3 + P1*l5 +
P2*l4 + P3*l3 + P1*l6 + P2*l5 + P3*l4 + P1*l7 + P2*l6 + P3*l5 + P4*l4 +
P2*l7 + P3*l6 + P4*l5 + P3*l7 + P4*l6 + P5*l5 + P4*l7 + P5*l6 + P5*l7 +
P6*l6 + P6*l7 + P7*l7 - RA*l1 - RA*l2 - RA*l3 - RA*l4 - RA*l5 - RA*l6 -
RA*l7 - H*x))/(H*l7)
F6a7a=- (2*P1*l1 + 2*P1*l2 + 2*P1*l3 + 2*P2*l2 + 2*P1*l4 + 2*P2*l3 +
2*P1*l5 + 2*P2*l4 + 2*P3*l3 + 2*P1*l6 + 2*P2*l5 + 2*P3*l4 + P1*l7 +

```

$$\frac{2*P2*16 + 2*P3*15 + 2*P4*14 + P2*17 + 2*P3*16 + 2*P4*15 + P3*17 + 2*P4*16 + 2*P5*15 + P4*17 + 2*P5*16 + P5*17 + 2*P6*16 + P6*17 + P7*17 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - RA*17 - H*x}{H}$$

$$F8s8a=(d16*(17*P6 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P2*17 + P3*16 + P4*15 + P3*17 + P4*16 + P5*15 + P4*17 + P5*16 + P5*17 + P6*16 + P7*17 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - H*x))/(H*18)$$

$$F8s9s=x$$

$$F7a9s=(d15*(17*P6 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P2*18 + P3*17 + P4*16 + P5*15 + P3*18 + P4*17 + P5*16 + P4*18 + P5*17 + P6*16 + P5*18 + P6*18 + P7*17 + P7*18 + P8*18 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - H*x))/(H*18)$$

$$F7a8a=- (34*P6 + 2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + 2*P2*16 + 2*P3*15 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*14 + P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + P3*18 + 2*P4*17 + 2*P5*16 + P4*18 + 2*P5*17 + 2*P6*16 + P5*18 + P6*18 + 2*P7*17 + P7*18 + P8*18 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - RA*18 - H*x)/H$$

$$F9s9a=(d18*(17*P5 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P2*18 + P3*17 + P4*16 + P5*15 + P3*18 + P4*17 + P5*16 + P4*18 + P6*16 + P5*18 + P6*17 + P6*18 + P7*17 + P7*18 + P8*18 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - H*x))/(H*19)$$

$$F9s10s=x$$

$$F8a10s=(d17*(17*P5 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P2*19 + P3*18 + P4*17 + P5*16 + P3*19 + P4*18 + P6*16 + P4*19 + P5*18 + P6*17 + P5*19 + P6*18 + P7*17 + P6*19 + P7*18 + P7*19 + P8*18 + P8*19 + P9*19 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - H*x))/(H*19)$$

$$F8a9a=- (34*P5 + 2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + 2*P2*16 + 2*P3*15 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + P2*19 + 2*P3*18 + 2*P4*17 + 2*P5*16 + P3*19 + 2*P4*18 + 2*P6*16 + P4*19 + 2*P5*18 + 2*P6*17 + P5*19 + 2*P6*18 + 2*P7*17 + P6*19 + 2*P7*18 + P7*19 + 2*P8*18 + P8*19 + P9*19 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - RA*19 - H*x)/H$$

$$F10s10a=- (d20*(P1*d19 + P2*d19 + P3*d19 + P4*d19 + P5*d19 + P6*d19 + P7*d19 + P8*d19 + P9*d19 + P10*d19 - RA*d19 - H*x))/(H*d19)$$

$$F10s11s=(17*P4*d19 + P1*d19*11 + P1*d19*12 + P1*d19*13 + P2*d19*12 + P1*d19*14 + P2*d19*13 + P1*d19*15 + P2*d19*14 + P3*d19*13 + P1*d19*16 + P2*d19*15 + P3*d19*14 + P1*d19*17 + P2*d19*16 + P3*d19*15 + P4*d19*14 + P1*d19*18 + P2*d19*17 + P3*d19*16 + P4*d19*15 + P1*d19*19 + P2*d19*18 + P3*d19*17 + P4*d19*16 + P5*d19*15 + P1*d19*110 + P2*d19*19 + P3*d19*18 + P5*d19*16 + P2*d19*110 + P3*d19*19 + P4*d19*18 + P5*d19*17 + P6*d19*16 + P3*d19*110 + P4*d19*19 + P5*d19*18 + P6*d19*17 + P4*d19*110 + P5*d19*19 + P6*d19*18 + P7*d19*17 + P5*d19*110 + P6*d19*19 + P7*d19*18 + P6*d19*110 + P7*d19*19 + P8*d19*18 + P7*d19*110 + P8*d19*19 + P8*d19*110 + P9*d19*19 + P9*d19*110 + P10*d19*110 - RA*d19*11 -$$

$$\frac{RA*d19*12 - RA*d19*13 - RA*d19*14 - RA*d19*15 - RA*d19*16 - RA*d19*17 - RA*d19*18 - RA*d19*19 - RA*d19*110 - H*110*x}{(H*d19)}$$

$$F9a11s=x$$

$$F9a10a=-\frac{(17*P4*d19 + P1*d19*11 + P1*d19*12 + P1*d19*13 + P2*d19*12 + P1*d19*14 + P2*d19*13 + P1*d19*15 + P2*d19*14 + P3*d19*13 + P1*d19*16 + P2*d19*15 + P3*d19*14 + P1*d19*17 + P2*d19*16 + P3*d19*15 + P4*d19*14 + P1*d19*18 + P2*d19*17 + P3*d19*16 + P4*d19*15 + P1*d19*19 + P2*d19*18 + P3*d19*17 + P4*d19*16 + P5*d19*15 + P2*d19*19 + P3*d19*18 + P5*d19*16 + P3*d19*19 + P4*d19*18 + P5*d19*17 + P6*d19*16 + P4*d19*19 + P5*d19*18 + P6*d19*17 + P5*d19*19 + P6*d19*18 + P7*d19*17 + P6*d19*19 + P7*d19*18 + P7*d19*19 + P8*d19*18 + P8*d19*19 + P9*d19*19 - RA*d19*11 - RA*d19*12 - RA*d19*13 - RA*d19*14 - RA*d19*15 - RA*d19*16 - RA*d19*17 - RA*d19*18 - RA*d19*19 + H*110*x)}{(H*d19)}$$

$$F11s11a=(d22*(17*P3 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P3*110 + P4*19 + P5*18 + P6*17 + P4*110 + P5*19 + P6*18 + P7*17 + P5*110 + P6*19 + P7*18 + P6*110 + P7*19 + P8*18 + P7*110 + P8*19 + P8*110 + P9*19 + P9*110 + P10*110 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - H*x))/(H*111)$$

$$F11s12s=x$$

$$F10a12s=(d21*(17*P3 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P4*111 + P5*110 + P6*19 + P7*18 + P5*111 + P6*110 + P7*19 + P8*18 + P6*111 + P7*110 + P8*19 + P7*111 + P8*110 + P9*19 + P8*111 + P9*110 + P9*111 + P10*110 + P10*111 + P11*111 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - H*x))/(H*111)$$

$$F10a11a=-\frac{(34*P3 + 2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + 2*P2*16 + 2*P3*15 + 2*P4*15 + 2*P1*19 + 2*P2*18 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + 2*P3*18 + 2*P4*17 + 2*P5*16 + P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + 2*P5*17 + 2*P6*16 + P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 + 2*P6*17 + P3*111 + 2*P4*110 + 2*P5*19 + 2*P6*18 + 2*P7*17 + P4*111 + 2*P5*110 + 2*P6*19 + 2*P7*18 + P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + P6*111 + 2*P7*110 + 2*P8*19 + P7*111 + 2*P8*110 + 2*P9*19 + P8*111 + 2*P9*110 + P9*111 + 2*P10*110 + P10*111 + P11*111 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - RA*111 - H*x)/H}$$

$$F12s12a=(d24*(17*P2 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P4*111 + P5*110 + P6*19 + P7*18 + P5*111 + P6*110 + P7*19 + P8*18 + P6*111 + P7*110 + P8*19 + P7*111 + P8*110 + P9*19 + P8*111 + P9*110 + P9*111 + P10*110 + P10*111 + P11*111 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - H*x))/(H*112)$$

$$F12s13s=x$$

$$F11a13s=(d23*(17*P2 + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 +$$

$$\frac{P4*14 + P1*18 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P5*112 + P6*111 + P7*110 + P8*19 + P6*112 + P7*111 + P8*110 + P9*19 + P7*112 + P8*111 + P9*110 + P8*112 + P9*111 + P10*110 + P9*112 + P10*111 + P10*112 + P11*111 + P11*112 + P12*112 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - H*x)}{(H*112)}$$

$$F11a12a = -(34*P2 + 2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P3*16 + 2*P4*15 + 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + 2*P5*17 + 2*P6*16 + P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 + 2*P6*17 + P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + 2*P6*18 + 2*P7*17 + P3*112 + 2*P4*111 + 2*P5*110 + 2*P6*19 + 2*P7*18 + P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + P5*112 + 2*P8*19 + P5*112 + 2*P6*111 + 2*P7*110 + 2*P8*19 + P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + P7*112 + 2*P8*111 + 2*P9*110 + P8*112 + 2*P9*111 + 2*P10*110 + P9*112 + 2*P10*111 + P10*112 + 2*P11*111 + P11*112 + P12*112 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - RA*112 - H*x)/H$$

$$F13s13a = (d26*(17*P1 - 17*RA + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P5*112 + P6*111 + P7*110 + P8*19 + P6*112 + P7*111 + P8*110 + P9*19 + P7*112 + P8*111 + P9*110 + P8*112 + P9*111 + P10*110 + P9*112 + P10*111 + P10*112 + P11*111 + P11*112 + P12*112 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - H*x))/(H*113)$$

$$F13s14s = x$$

$$F12a14s = (d25*(17*P1 - 17*RA + P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P6*113 + P7*112 + P8*111 + P9*110 + P7*113 + P8*112 + P9*111 + P10*110 + P8*113 + P9*112 + P10*111 + P9*113 + P10*112 + P11*111 + P10*113 + P11*112 + P11*113 + P12*112 + P12*113 + P13*113 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - H*x))/(H*113)$$

$$F12a13a = -(34*P1 - 34*RA + 2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P2*17 + 2*P3*16 + 2*P4*15 + 2*P1*18 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*19 + 2*P2*19 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19$$

```

+ 2*P5*18 + 2*P6*17 + P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19
+ 2*P6*18 + 2*P7*17 + P2*113 + 2*P3*112 + 2*P4*111 + 2*P5*110 + 2*P6*19
+ 2*P7*18 + P3*113 + 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18
+ P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + 2*P8*19 + P5*113 + 2*P6*112
+ 2*P7*111 + 2*P8*110 + 2*P9*19 + P6*113 + 2*P7*112 + 2*P8*111 +
2*P9*110 + P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + P8*113 + 2*P9*112
+ 2*P10*111 + P9*113 + 2*P10*112 + 2*P11*111 + P10*113 + 2*P11*112 +
P11*113 + 2*P12*112 + P12*113 + P13*113 - 2*RA*11 - 2*RA*12 - 2*RA*13 -
2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 -
2*RA*112 - RA*113 - H*x)/H
F14s14a=(d28*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +
P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +
P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +
P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +
P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +
P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P3*113 + P4*112 +
P5*111 + P6*110 + P7*19 + P8*18 + P4*113 + P5*112 + P6*111 + P7*110 +
P8*19 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P6*113 + P7*112 +
P8*111 + P9*110 + P7*113 + P8*112 + P9*111 + P10*110 + P8*113 + P9*112
+ P10*111 + P9*113 + P10*112 + P11*111 + P10*113 + P11*112 + P11*113 +
P12*112 + P12*113 + P13*113 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 -
RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 -
H*x))/(H*114)
F14s15s=x
F13a15s=(d27*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +
P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +
P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +
P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +
P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +
P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P2*114 +
P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P3*114 + P4*113 +
P5*112 + P6*111 + P7*110 + P8*19 + P4*114 + P5*113 + P6*112 + P7*111 +
P8*110 + P9*19 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 + P6*114 +
P7*113 + P8*112 + P9*111 + P10*110 + P7*114 + P8*113 + P9*112 + P10*111
+ P8*114 + P9*113 + P10*112 + P11*111 + P9*114 + P10*113 + P11*112 +
P10*114 + P11*113 + P12*112 + P11*114 + P12*113 + P12*114 + P13*113 +
P13*114 + P14*114 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 -
RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 -
H*x))/(H*114)
F13a14a=- (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 +
2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 +
2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 +
2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 +
2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 +
2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18
+ 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 +
2*P6*18 + 2*P7*17 + P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + 2*P5*110
+ 2*P6*19 + 2*P7*18 + P2*114 + 2*P3*113 + 2*P4*112 + 2*P5*111 +
2*P6*110 + 2*P7*19 + 2*P8*18 + P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111
+ 2*P7*110 + 2*P8*19 + P4*114 + 2*P5*113 + 2*P6*112 + 2*P7*111 +
2*P8*110 + 2*P9*19 + P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + 2*P9*110
+ P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + P7*114 +
2*P8*113 + 2*P9*112 + 2*P10*111 + P8*114 + 2*P9*113 + 2*P10*112 +
2*P11*111 + P9*114 + 2*P10*113 + 2*P11*112 + P10*114 + 2*P11*113 +
2*P12*112 + P11*114 + 2*P12*113 + P12*114 + 2*P13*113 + P13*114 +

```

$$\frac{P14*114 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - 2*RA*112 - 2*RA*113 - RA*114 - H*x}{H}$$

$$F15s15a=(d30*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 + P6*114 + P7*113 + P8*112 + P9*111 + P10*110 + P7*114 + P8*113 + P9*112 + P10*111 + P8*114 + P9*113 + P10*112 + P11*111 + P9*114 + P10*113 + P11*112 + P10*114 + P11*113 + P12*112 + P11*114 + P12*113 + P12*114 + P13*113 + P13*114 + P14*114 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - H*x))/(H*115)$$

$$F15s16s=x$$

$$F14a16s=(d29*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 + P5*115 + P6*114 + P7*113 + P8*112 + P9*111 + P10*110 + P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111 + P8*115 + P9*114 + P10*113 + P11*112 + P9*115 + P10*114 + P11*113 + P12*112 + P10*115 + P11*114 + P12*113 + P11*115 + P12*114 + P13*113 + P12*115 + P13*114 + P13*115 + P14*114 + P14*115 + P15*115 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - H*x))/(H*115)$$

$$F14a15a=- (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 + 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + 2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + 2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + 2*P8*19 + P3*115 + 2*P4*114 + 2*P5*113 + 2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + 2*P9*110 + P5*115 + 2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + P6*115 + 2*P7*114 + 2*P8*113 + 2*P9*112 + 2*P10*111 + P7*115 + 2*P8*114 + 2*P9*113 + 2*P10*112 + 2*P11*111 + P8*115 + 2*P9*114 + 2*P10*113 + 2*P11*112 + P9*115 + 2*P10*114 + 2*P11*113 + 2*P12*112 + P10*115 + 2*P11*114 + 2*P12*113 + P11*115 + 2*P13*113 + P12*115 + 2*P13*114 + P13*115 - 2*RA*11 - 2*RA*12$$

$$\begin{aligned}
& - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - \\
& 2*RA*110 - 2*RA*111 - 2*RA*112 - 2*RA*113 - 2*RA*114 - RA*115 - H*x)/H \\
F16s16a=& (d32*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P2*115 + \\
& P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P3*115 + P4*114 + \\
& P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P4*115 + P5*114 + P6*113 + \\
& P7*112 + P8*111 + P9*110 + P5*115 + P6*114 + P7*113 + P8*112 + P9*111 + \\
& P10*110 + P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P7*115 + P8*114 \\
& + P9*113 + P10*112 + P11*111 + P8*115 + P9*114 + P10*113 + P11*112 + \\
& P9*115 + P10*114 + P11*113 + P12*112 + P10*115 + P11*114 + P12*113 + \\
& P11*115 + P12*114 + P13*113 + P12*115 + P13*114 + P13*115 + P14*114 + \\
& P14*115 + P15*115 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - \\
& RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - \\
& RA*115 - H*x))/(H*116) \\
F16s17s=& x \\
F15a17s=& (d31*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
& P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P2*116 + \\
& P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P3*116 + \\
& P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 + P4*116 + P5*115 + \\
& P6*114 + P7*113 + P8*112 + P9*111 + P10*110 + P5*116 + P6*115 + P7*114 \\
& + P8*113 + P9*112 + P10*111 + P6*116 + P7*115 + P8*114 + P9*113 + \\
& P10*112 + P11*111 + P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + \\
& P8*116 + P9*115 + P10*114 + P11*113 + P12*112 + P9*116 + P10*115 + \\
& P11*114 + P12*113 + P10*116 + P11*115 + P12*114 + P13*113 + P11*116 + \\
& P12*115 + P13*114 + P12*116 + P13*115 + P14*114 + P13*116 + P14*115 + \\
& P14*116 + P15*115 + P15*116 + P16*116 - RA*11 - RA*12 - RA*13 - RA*14 - \\
& RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - \\
& RA*113 - RA*114 - RA*115 - RA*116 - H*x))/(H*116) \\
F15a16a=& -(2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
& 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
& 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
& 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
& 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + \\
& 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 \\
& + 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + \\
& 2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + \\
& 2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + \\
& 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + P1*116 + 2*P2*115 \\
& + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + 2*P8*19 + \\
& P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + 2*P6*112 + 2*P7*111 + \\
& 2*P8*110 + 2*P9*19 + P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 \\
& + 2*P8*111 + 2*P9*110 + P4*116 + 2*P5*115 + 2*P6*114 + 2*P7*113 + \\
& 2*P8*112 + 2*P9*111 + 2*P10*110 + P5*116 + 2*P6*115 + 2*P7*114 + \\
& 2*P8*113 + 2*P9*112 + 2*P10*111 + P6*116 + 2*P7*115 + 2*P8*114 +
\end{aligned}$$

$$\begin{aligned}
& 2*P9*113 + 2*P10*112 + 2*P11*111 + P7*116 + 2*P8*115 + 2*P9*114 + \\
& 2*P10*113 + 2*P11*112 + P8*116 + 2*P9*115 + 2*P10*114 + 2*P11*113 + \\
& 2*P12*112 + P9*116 + 2*P10*115 + 2*P11*114 + 2*P12*113 + P10*116 + \\
& 2*P11*115 + 2*P12*114 + 2*P13*113 + P11*116 + 2*P12*115 + 2*P13*114 + \\
& P12*116 + 2*P13*115 + 2*P14*114 + P13*116 + 2*P14*115 + P14*116 + \\
& 2*P15*115 + P15*116 + P16*116 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - \\
& 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - \\
& 2*RA*112 - 2*RA*113 - 2*RA*114 - 2*RA*115 - RA*116 - H*x)/H \\
F17s17a=& (d34*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
& P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P2*116 + \\
& P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + P3*116 + \\
& P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 + P4*116 + P5*115 + \\
& P6*114 + P7*113 + P8*112 + P9*111 + P10*110 + P5*116 + P6*115 + P7*114 \\
& + P8*113 + P9*112 + P10*111 + P6*116 + P7*115 + P8*114 + P9*113 + \\
& P10*112 + P11*111 + P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + \\
& P8*116 + P9*115 + P10*114 + P11*113 + P12*112 + P9*116 + P10*115 + \\
& P11*114 + P12*113 + P10*116 + P11*115 + P12*114 + P13*113 + P11*116 + \\
& P12*115 + P13*114 + P12*116 + P13*115 + P14*114 + P13*116 + P14*115 + \\
& P14*116 + P15*115 + P15*116 + P16*116 - RA*11 - RA*12 - RA*13 - RA*14 - \\
& RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - \\
& RA*113 - RA*114 - RA*115 - RA*116 - H*x))/(H*117) \\
F17s18s=& x \\
F16a18s=& (d33*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
& P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 + \\
& P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + \\
& P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 + \\
& P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + P8*112 + P9*111 + P10*110 \\
& + P4*117 + P5*116 + P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + \\
& P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111 + P6*117 \\
& + P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P7*117 + P8*116 + \\
& P9*115 + P10*114 + P11*113 + P12*112 + P8*117 + P9*116 + P10*115 + \\
& P11*114 + P12*113 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + \\
& P10*117 + P11*116 + P12*115 + P13*114 + P11*117 + P12*116 + P13*115 + \\
& P14*114 + P12*117 + P13*116 + P14*115 + P13*117 + P14*116 + P15*115 + \\
& P14*117 + P15*116 + P15*117 + P16*116 + P16*117 + P17*117 - RA*11 - \\
& RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 \\
& - RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - \\
& H*x))/(H*117) \\
F16a17a=& -(2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
& 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
& 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
& 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
& 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 +
\end{aligned}$$

$2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18$
 $+ 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 +$
 $2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 +$
 $2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 +$
 $2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 +$
 $2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 +$
 $2*P8*19 + P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + 2*P6*112$
 $+ 2*P7*111 + 2*P8*110 + 2*P9*19 + P2*117 + 2*P3*116 + 2*P4*115 +$
 $2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + 2*P9*110 + P3*117 +$
 $2*P4*116 + 2*P5*115 + 2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 +$
 $2*P10*110 + P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 + 2*P8*113 +$
 $2*P9*112 + 2*P10*111 + P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 +$
 $2*P9*113 + 2*P10*112 + 2*P11*111 + P6*117 + 2*P7*116 + 2*P8*115 +$
 $2*P9*114 + 2*P10*113 + 2*P11*112 + P7*117 + 2*P8*116 + 2*P9*115 +$
 $2*P10*114 + 2*P11*113 + 2*P12*112 + P8*117 + 2*P9*116 + 2*P10*115 +$
 $2*P11*114 + 2*P12*113 + P9*117 + 2*P10*116 + 2*P11*115 + 2*P12*114 +$
 $2*P13*113 + P10*117 + 2*P11*116 + 2*P12*115 + 2*P13*114 + P11*117 +$
 $2*P12*116 + 2*P13*115 + 2*P14*114 + P12*117 + 2*P13*116 + 2*P14*115 +$
 $P13*117 + 2*P14*116 + 2*P15*115 + P14*117 + 2*P15*116 + P15*117 +$
 $2*P16*116 + P16*117 + P17*117 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 -$
 $2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 -$
 $2*RA*112 - 2*RA*113 - 2*RA*114 - 2*RA*115 - 2*RA*116 - RA*117 - H*x)/H$
 $F18s18a=(d36*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + P9*110 +$
 $P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + P8*112 + P9*111 + P10*110$
 $+ P4*117 + P5*116 + P6*115 + P7*114 + P8*113 + P9*112 + P10*111 +$
 $P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111 + P6*117$
 $+ P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P7*117 + P8*116 +$
 $P9*115 + P10*114 + P11*113 + P12*112 + P8*117 + P9*116 + P10*115 +$
 $P11*114 + P12*113 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 +$
 $P10*117 + P11*116 + P12*115 + P13*114 + P11*117 + P12*116 + P13*115 +$
 $P14*114 + P12*117 + P13*116 + P14*115 + P13*117 + P14*116 + P15*115 +$
 $P14*117 + P15*116 + P15*117 + P16*116 + P16*117 + P17*117 - RA*11 -$
 $RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110$
 $- RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 -$
 $H*x)/(H*118)$
 $F18s19s=x$
 $F17a19s=(d35*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +$

$$\begin{aligned}
& P9*110 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + P8*112 + \\
& P9*111 + P10*110 + P3*118 + P4*117 + P5*116 + P6*115 + P7*114 + P8*113 \\
& + P9*112 + P10*111 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + \\
& P9*113 + P10*112 + P11*111 + P5*118 + P6*117 + P7*116 + P8*115 + P9*114 \\
& + P10*113 + P11*112 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 + \\
& P11*113 + P12*112 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + \\
& P12*113 + P8*118 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + \\
& P9*118 + P10*117 + P11*116 + P12*115 + P13*114 + P10*118 + P11*117 + \\
& P12*116 + P13*115 + P14*114 + P11*118 + P12*117 + P13*116 + P14*115 + \\
& P12*118 + P13*117 + P14*116 + P15*115 + P13*118 + P14*117 + P15*116 + \\
& P14*118 + P15*117 + P16*116 + P15*118 + P16*117 + P16*118 + P17*117 + \\
& P17*118 + P18*118 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - \\
& RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - \\
& RA*115 - RA*116 - RA*117 - RA*118 - H*x))/ (H*118) \\
F17a18a = & -(2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
& 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
& 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
& 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
& 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + \\
& 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 \\
& + 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + \\
& 2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + \\
& 2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + \\
& 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 + \\
& 2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + \\
& 2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + \\
& 2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + P1*118 + 2*P2*117 + 2*P3*116 \\
& + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + 2*P9*110 + \\
& P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 + 2*P6*114 + 2*P7*113 + \\
& 2*P8*112 + 2*P9*111 + 2*P10*110 + P3*118 + 2*P4*117 + 2*P5*116 + \\
& 2*P6*115 + 2*P7*114 + 2*P8*113 + 2*P9*112 + 2*P10*111 + P4*118 + \\
& 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 + 2*P10*112 + \\
& 2*P11*111 + P5*118 + 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + \\
& 2*P10*113 + 2*P11*112 + P6*118 + 2*P7*117 + 2*P8*116 + 2*P9*115 + \\
& 2*P10*114 + 2*P11*113 + 2*P12*112 + P7*118 + 2*P8*117 + 2*P9*116 + \\
& 2*P10*115 + 2*P11*114 + 2*P12*113 + P8*118 + 2*P9*117 + 2*P10*116 + \\
& 2*P11*115 + 2*P12*114 + 2*P13*113 + P9*118 + 2*P10*117 + 2*P11*116 + \\
& 2*P12*115 + 2*P13*114 + P10*118 + 2*P11*117 + 2*P12*116 + 2*P13*115 + \\
& 2*P14*114 + P11*118 + 2*P12*117 + 2*P13*116 + 2*P14*115 + P12*118 + \\
& 2*P13*117 + 2*P14*116 + 2*P15*115 + P13*118 + 2*P14*117 + 2*P15*116 + \\
& P14*118 + 2*P15*117 + 2*P16*116 + P15*118 + 2*P16*117 + P16*118 + \\
& 2*P17*117 + P17*118 + P18*118 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - \\
& 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - \\
& 2*RA*112 - 2*RA*113 - 2*RA*114 - 2*RA*115 - 2*RA*116 - 2*RA*117 - \\
& RA*118 - H*x)/H \\
F19s19a = & (d38*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
& P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 + \\
& P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + \\
& P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + \\
& P9*110 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + P8*112 +
\end{aligned}$$

$$\begin{aligned}
&P9*111 + P10*110 + P3*118 + P4*117 + P5*116 + P6*115 + P7*114 + P8*113 \\
&+ P9*112 + P10*111 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + \\
&P9*113 + P10*112 + P11*111 + P5*118 + P6*117 + P7*116 + P8*115 + P9*114 + \\
&+ P10*113 + P11*112 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 + \\
&P11*113 + P12*112 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + \\
&P12*113 + P8*118 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + \\
&P9*118 + P10*117 + P11*116 + P12*115 + P13*114 + P10*118 + P11*117 + \\
&P12*116 + P13*115 + P14*114 + P11*118 + P12*117 + P13*116 + P14*115 + \\
&P12*118 + P13*117 + P14*116 + P15*115 + P13*118 + P14*117 + P15*116 + \\
&P14*118 + P15*117 + P16*116 + P15*118 + P16*117 + P16*118 + P17*117 + \\
&P17*118 + P18*118 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - \\
&RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - \\
&RA*115 - RA*116 - RA*117 - RA*118 - H*x)/(H*119) \\
F19s20s=x \\
F18a20s=(d37*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 + \\
P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + \\
P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + \\
P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + \\
P8*112 + P9*111 + P10*110 + P2*119 + P3*118 + P4*117 + P5*116 + P6*115 \\
+ P7*114 + P8*113 + P9*112 + P10*111 + P3*119 + P4*118 + P5*117 + \\
P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111 + P4*119 + P5*118 \\
+ P6*117 + P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P5*119 + \\
P6*118 + P7*117 + P8*116 + P9*115 + P10*114 + P11*113 + P12*112 + \\
P6*119 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + P12*113 + \\
P7*119 + P8*118 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + \\
P8*119 + P9*118 + P10*117 + P11*116 + P12*115 + P13*114 + P9*119 + \\
P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P10*119 + P11*118 + \\
P12*117 + P13*116 + P14*115 + P11*119 + P12*118 + P13*117 + P14*116 + \\
P15*115 + P12*119 + P13*118 + P14*117 + P15*116 + P13*119 + P14*118 + \\
P15*117 + P16*116 + P14*119 + P15*118 + P16*117 + P15*119 + P16*118 + \\
P17*117 + P16*119 + P17*118 + P17*119 + P18*118 + P18*119 + P19*119 - \\
RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - \\
RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - \\
RA*118 - RA*119 - H*x)/(H*119) \\
F18a19a=- (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + \\
2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 \\
+ 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + \\
2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + \\
2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + \\
2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 + \\
2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + \\
2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + \\
2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 + \\
2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + \\
2*P9*110 + P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 +
\end{aligned}$$

$2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + P2*119 +$
 $2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 + 2*P8*113 +$
 $2*P9*112 + 2*P10*111 + P3*119 + 2*P4*118 + 2*P5*117 + 2*P6*116 +$
 $2*P7*115 + 2*P8*114 + 2*P9*113 + 2*P10*112 + 2*P11*111 + P4*119 +$
 $2*P5*118 + 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + 2*P10*113 +$
 $2*P11*112 + P5*119 + 2*P6*118 + 2*P7*117 + 2*P8*116 + 2*P9*115 +$
 $2*P10*114 + 2*P11*113 + 2*P12*112 + P6*119 + 2*P7*118 + 2*P8*117 +$
 $2*P9*116 + 2*P10*115 + 2*P11*114 + 2*P12*113 + P7*119 + 2*P8*118 +$
 $2*P9*117 + 2*P10*116 + 2*P11*115 + 2*P12*114 + 2*P13*113 + P8*119 +$
 $2*P9*118 + 2*P10*117 + 2*P11*116 + 2*P12*115 + 2*P13*114 + P9*119 +$
 $2*P10*118 + 2*P11*117 + 2*P12*116 + 2*P13*115 + 2*P14*114 + P10*119 +$
 $2*P11*118 + 2*P12*117 + 2*P13*116 + 2*P14*115 + P11*119 + 2*P12*118 +$
 $2*P13*117 + 2*P14*116 + 2*P15*115 + P12*119 + 2*P13*118 + 2*P14*117 +$
 $2*P15*116 + P13*119 + 2*P14*118 + 2*P15*117 + 2*P16*116 + P14*119 +$
 $2*P15*118 + 2*P16*117 + P15*119 + 2*P16*118 + 2*P17*117 + P16*119 +$
 $2*P17*118 + P17*119 + 2*P18*118 + P18*119 + P19*119 - 2*RA*11 - 2*RA*12$
 $- 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 -$
 $2*RA*110 - 2*RA*111 - 2*RA*112 - 2*RA*113 - 2*RA*114 - 2*RA*115 -$
 $2*RA*116 - 2*RA*117 - 2*RA*118 - RA*119 - H*x)/H$
 $F20s20a=(d40*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +$
 $P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +$
 $P8*112 + P9*111 + P10*110 + P2*119 + P3*118 + P4*117 + P5*116 + P6*115$
 $+ P7*114 + P8*113 + P9*112 + P10*111 + P3*119 + P4*118 + P5*117 +$
 $P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111 + P4*119 + P5*118$
 $+ P6*117 + P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P5*119 +$
 $P6*118 + P7*117 + P8*116 + P9*115 + P10*114 + P11*113 + P12*112 +$
 $P6*119 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + P12*113 +$
 $P7*119 + P8*118 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 +$
 $P8*119 + P9*118 + P10*117 + P11*116 + P12*115 + P13*114 + P9*119 +$
 $P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P10*119 + P11*118 +$
 $P12*117 + P13*116 + P14*115 + P11*119 + P12*118 + P13*117 + P14*116 +$
 $P15*115 + P12*119 + P13*118 + P14*117 + P15*116 + P13*119 + P14*118 +$
 $P15*117 + P16*116 + P14*119 + P15*118 + P16*117 + P15*119 + P16*118 +$
 $P17*117 + P16*119 + P17*118 + P17*119 + P18*118 + P18*119 + P19*119 -$
 $RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 -$
 $RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 -$
 $RA*118 - RA*119 - H*x))/(H*120)$
 $F20s21s=x$
 $F19a21s=(d39*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$

$$\begin{aligned}
& P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*119 + \\
& P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + \\
& P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + \\
& P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116 + \\
& P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P2*120 + P3*119 + \\
& P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111 + \\
& P3*120 + P4*119 + P5*118 + P6*117 + P7*116 + P8*115 + P9*114 + \\
& P10*113 + P11*112 + P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + \\
& P10*114 + P11*113 + P12*112 + P5*120 + P6*119 + P7*118 + P8*117 + \\
& P9*116 + P10*115 + P11*114 + P12*113 + P6*120 + P7*119 + P8*118 + \\
& P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + P7*120 + P8*119 + \\
& P9*118 + P10*117 + P11*116 + P12*115 + P13*114 + P8*120 + P9*119 + \\
& P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P9*120 + P10*119 + \\
& P11*118 + P12*117 + P13*116 + P14*115 + P10*120 + P11*119 + P12*118 + \\
& P13*117 + P14*116 + P15*115 + P11*120 + P12*119 + P13*118 + P14*117 + \\
& P15*116 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P13*120 + \\
& P14*119 + P15*118 + P16*117 + P14*120 + P15*119 + P16*118 + P17*117 + \\
& P15*120 + P16*119 + P17*118 + P16*120 + P17*119 + P18*118 + P17*120 + \\
& P18*119 + P18*120 + P19*119 + P19*120 + P20*120 - RA*11 - RA*12 - RA*13 \\
& - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - \\
& RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 - \\
& RA*120 - H*x) / (H*120) \\
F19a20a = & (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
& 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
& 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
& 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
& 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + \\
& 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 \\
& + 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + \\
& 2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + \\
& 2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + \\
& 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 + \\
& 2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + \\
& 2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + \\
& 2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 + \\
& 2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + \\
& 2*P9*110 + 2*P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 + \\
& 2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + P1*120 + \\
& 2*P2*119 + 2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 + \\
& 2*P8*113 + 2*P9*112 + 2*P10*111 + P2*120 + 2*P3*119 + 2*P4*118 + \\
& 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 + 2*P10*112 + \\
& 2*P11*111 + P3*120 + 2*P4*119 + 2*P5*118 + 2*P6*117 + 2*P7*116 + \\
& 2*P8*115 + 2*P9*114 + 2*P10*113 + 2*P11*112 + P4*120 + 2*P5*119 + \\
& 2*P6*118 + 2*P7*117 + 2*P8*116 + 2*P9*115 + 2*P10*114 + 2*P11*113 + \\
& 2*P12*112 + P5*120 + 2*P6*119 + 2*P7*118 + 2*P8*117 + 2*P9*116 + \\
& 2*P10*115 + 2*P11*114 + 2*P12*113 + P6*120 + 2*P7*119 + 2*P8*118 + \\
& 2*P9*117 + 2*P10*116 + 2*P11*115 + 2*P12*114 + 2*P13*113 + P7*120 + \\
& 2*P8*119 + 2*P9*118 + 2*P10*117 + 2*P11*116 + 2*P12*115 + 2*P13*114 + \\
& P8*120 + 2*P9*119 + 2*P10*118 + 2*P11*117 + 2*P12*116 + 2*P13*115 + \\
& 2*P14*114 + P9*120 + 2*P10*119 + 2*P11*118 + 2*P12*117 + 2*P13*116 + \\
& 2*P14*115 + P10*120 + 2*P11*119 + 2*P12*118 + 2*P13*117 + 2*P14*116 + \\
& 2*P15*115 + P11*120 + 2*P12*119 + 2*P13*118 + 2*P14*117 + 2*P15*116 + \\
& P12*120 + 2*P13*119 + 2*P14*118 + 2*P15*117 + 2*P16*116 + P13*120 + \\
& 2*P14*119 + 2*P15*118 + 2*P16*117 + P14*120 + 2*P15*119 + 2*P16*118 + \\
& 2*P17*117 + P15*120 + 2*P16*119 + 2*P17*118 + P16*120 + 2*P17*119 + \\
& 2*P18*118 + P17*120 + 2*P18*119 + P18*120 + 2*P19*119 + P19*120 + \\
& P20*120 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 -
\end{aligned}$$

$2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - 2*RA*112 - 2*RA*113$
 $- 2*RA*114 - 2*RA*115 - 2*RA*116 - 2*RA*117 - 2*RA*118 - 2*RA*119 -$
 $RA*120 - H*x)/H$
 $F21s21a=(d42*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +$
 $P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +$
 $P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116$
 $+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P2*120 + P3*119 +$
 $P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + P11*111$
 $+ P3*120 + P4*119 + P5*118 + P6*117 + P7*116 + P8*115 + P9*114 +$
 $P10*113 + P11*112 + P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115$
 $+ P10*114 + P11*113 + P12*112 + P5*120 + P6*119 + P7*118 + P8*117 +$
 $P9*116 + P10*115 + P11*114 + P12*113 + P6*120 + P7*119 + P8*118 +$
 $P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + P7*120 + P8*119 +$
 $P9*118 + P10*117 + P11*116 + P12*115 + P13*114 + P8*120 + P9*119 +$
 $P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P9*120 + P10*119 +$
 $P11*118 + P12*117 + P13*116 + P14*115 + P10*120 + P11*119 + P12*118 +$
 $P13*117 + P14*116 + P15*115 + P11*120 + P12*119 + P13*118 + P14*117 +$
 $P15*116 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P13*120 +$
 $P14*119 + P15*118 + P16*117 + P14*120 + P15*119 + P16*118 + P17*117 +$
 $P15*120 + P16*119 + P17*118 + P16*120 + P17*119 + P18*118 + P17*120 +$
 $P18*119 + P18*120 + P19*119 + P19*120 + P20*120 - RA*11 - RA*12 - RA*13$
 $- RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 -$
 $RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 -$
 $RA*120 - H*x))/(H*121)$
 $F21s22s=x$
 $F20a22s=(d41*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +$
 $P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +$
 $P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116$
 $+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +$
 $P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112$
 $+ P11*111 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 + P7*116 +$
 $P8*115 + P9*114 + P10*113 + P11*112 + P3*121 + P4*120 + P5*119 + P6*118$
 $+ P7*117 + P8*116 + P9*115 + P10*114 + P11*113 + P12*112 + P4*121 +$
 $P5*120 + P6*119 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 +$
 $P12*113 + P5*121 + P6*120 + P7*119 + P8*118 + P9*117 + P10*116 +$
 $P11*115 + P12*114 + P13*113 + P6*121 + P7*120 + P8*119 + P9*118 +$
 $P10*117 + P11*116 + P12*115 + P13*114 + P7*121 + P8*120 + P9*119 +$

$$\begin{aligned}
& P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P8*121 + P9*120 + \\
& P10*119 + P11*118 + P12*117 + P13*116 + P14*115 + P9*121 + P10*120 + \\
& P11*119 + P12*118 + P13*117 + P14*116 + P15*115 + P10*121 + P11*120 + \\
& P12*119 + P13*118 + P14*117 + P15*116 + P11*121 + P12*120 + P13*119 + \\
& P14*118 + P15*117 + P16*116 + P12*121 + P13*120 + P14*119 + P15*118 + \\
& P16*117 + P13*121 + P14*120 + P15*119 + P16*118 + P17*117 + P14*121 + \\
& P15*120 + P16*119 + P17*118 + P15*121 + P16*120 + P17*119 + P18*118 + \\
& P16*121 + P17*120 + P18*119 + P17*121 + P18*120 + P19*119 + P18*121 + \\
& P19*120 + P19*121 + P20*120 + P20*121 + P21*121 - RA*11 - RA*12 - RA*13 \\
& - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - \\
& RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 - \\
& RA*120 - RA*121 - H*x) / (H*121) \\
F20a21a = & -(2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
& 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
& 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
& 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
& 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + \\
& 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 \\
& + 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + \\
& 2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + \\
& 2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + \\
& 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 + \\
& 2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + \\
& 2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + \\
& 2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 + \\
& 2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + \\
& 2*P9*110 + 2*P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 + \\
& 2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + 2*P1*120 + \\
& 2*P2*119 + 2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 + \\
& 2*P8*113 + 2*P9*112 + 2*P10*111 + P1*121 + 2*P2*120 + 2*P3*119 + \\
& 2*P4*118 + 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 + \\
& 2*P10*112 + 2*P11*111 + P2*121 + 2*P3*120 + 2*P4*119 + 2*P5*118 + \\
& 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + 2*P10*113 + 2*P11*112 + \\
& P3*121 + 2*P4*120 + 2*P5*119 + 2*P6*118 + 2*P7*117 + 2*P8*116 + \\
& 2*P9*115 + 2*P10*114 + 2*P11*113 + 2*P12*112 + P4*121 + 2*P5*120 + \\
& 2*P6*119 + 2*P7*118 + 2*P8*117 + 2*P9*116 + 2*P10*115 + 2*P11*114 + \\
& 2*P12*113 + P5*121 + 2*P6*120 + 2*P7*119 + 2*P8*118 + 2*P9*117 + \\
& 2*P10*116 + 2*P11*115 + 2*P12*114 + 2*P13*113 + P6*121 + 2*P7*120 + \\
& 2*P8*119 + 2*P9*118 + 2*P10*117 + 2*P11*116 + 2*P12*115 + 2*P13*114 + \\
& P7*121 + 2*P8*120 + 2*P9*119 + 2*P10*118 + 2*P11*117 + 2*P12*116 + \\
& 2*P13*115 + 2*P14*114 + P8*121 + 2*P9*120 + 2*P10*119 + 2*P11*118 + \\
& 2*P12*117 + 2*P13*116 + 2*P14*115 + P9*121 + 2*P10*120 + 2*P11*119 + \\
& 2*P12*118 + 2*P13*117 + 2*P14*116 + 2*P15*115 + P10*121 + 2*P11*120 + \\
& 2*P12*119 + 2*P13*118 + 2*P14*117 + 2*P15*116 + P11*121 + 2*P12*120 + \\
& 2*P13*119 + 2*P14*118 + 2*P15*117 + 2*P16*116 + P12*121 + 2*P13*120 + \\
& 2*P14*119 + 2*P15*118 + 2*P16*117 + P13*121 + 2*P14*120 + 2*P15*119 + \\
& 2*P16*118 + 2*P17*117 + P14*121 + 2*P15*120 + 2*P16*119 + 2*P17*118 + \\
& P15*121 + 2*P16*120 + 2*P17*119 + 2*P18*118 + P16*121 + 2*P17*120 + \\
& 2*P18*119 + P17*121 + 2*P18*120 + 2*P19*119 + P18*121 + 2*P19*120 + \\
& P19*121 + 2*P20*120 + P20*121 + P21*121 - 2*RA*11 - 2*RA*12 - 2*RA*13 - \\
& 2*RA*14 - 2*RA*15 - 2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - \\
& 2*RA*111 - 2*RA*112 - 2*RA*113 - 2*RA*114 - 2*RA*115 - 2*RA*116 - \\
& 2*RA*117 - 2*RA*118 - 2*RA*119 - 2*RA*120 - RA*121 - H*x) / H \\
F22s22a = & (d44*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
\end{aligned}$$

$$\begin{aligned}
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
& P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 + \\
& P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + \\
& P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + \\
& P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + \\
& P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116 + \\
& P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 + \\
& P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + \\
& P11*111 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 + P7*116 + \\
& P8*115 + P9*114 + P10*113 + P11*112 + P3*121 + P4*120 + P5*119 + P6*118 + \\
& P7*117 + P8*116 + P9*115 + P10*114 + P11*113 + P12*112 + P4*121 + \\
& P5*120 + P6*119 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + \\
& P12*113 + P5*121 + P6*120 + P7*119 + P8*118 + P9*117 + P10*116 + \\
& P11*115 + P12*114 + P13*113 + P6*121 + P7*120 + P8*119 + P9*118 + \\
& P10*117 + P11*116 + P12*115 + P13*114 + P7*121 + P8*120 + P9*119 + \\
& P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P8*121 + P9*120 + \\
& P10*119 + P11*118 + P12*117 + P13*116 + P14*115 + P9*121 + P10*120 + \\
& P11*119 + P12*118 + P13*117 + P14*116 + P15*115 + P10*121 + P11*120 + \\
& P12*119 + P13*118 + P14*117 + P15*116 + P11*121 + P12*120 + P13*119 + \\
& P14*118 + P15*117 + P16*116 + P12*121 + P13*120 + P14*119 + P15*118 + \\
& P16*117 + P13*121 + P14*120 + P15*119 + P16*118 + P17*117 + P14*121 + \\
& P15*120 + P16*119 + P17*118 + P15*121 + P16*120 + P17*119 + P18*118 + \\
& P16*121 + P17*120 + P18*119 + P17*121 + P18*120 + P19*119 + P18*121 + \\
& P19*120 + P19*121 + P20*120 + P20*121 + P21*121 - RA*11 - RA*12 - RA*13 - \\
& RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - \\
& RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 - \\
& RA*120 - RA*121 - H*x)/(H*122) \\
& F22s23s=x \\
& F21a23s=(d43*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 + \\
& P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 + \\
& P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 + \\
& P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 + \\
& P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 + \\
& P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 + \\
& P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 + \\
& P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 + \\
& P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 + \\
& P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 + \\
& P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 + \\
& P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 + \\
& P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116 + \\
& P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 + \\
& P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112 + \\
& P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 + \\
& P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P2*122 + P3*121 + P4*120 + \\
& P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 + P11*113 + \\
& P12*112 + P3*122 + P4*121 + P5*120 + P6*119 + P7*118 + P8*117 + P9*116 + \\
& P10*115 + P11*114 + P12*113 + P4*122 + P5*121 + P6*120 + P7*119 + \\
& P8*118 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + P5*122 + \\
& P6*121 + P7*120 + P8*119 + P9*118 + P10*117 + P11*116 + P12*115 + \\
& P13*114 + P6*122 + P7*121 + P8*120 + P9*119 + P10*118 + P11*117 + \\
& P12*116 + P13*115 + P14*114 + P7*122 + P8*121 + P9*120 + P10*119 + \\
& P11*118 + P12*117 + P13*116 + P14*115 + P8*122 + P9*121 + P10*120 + \\
& P11*119 + P12*118 + P13*117 + P14*116 + P15*115 + P9*122 + P10*121 +
\end{aligned}$$

P11*120 + P12*119 + P13*118 + P14*117 + P15*116 + P10*122 + P11*121 +
P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P11*122 + P12*121 +
P13*120 + P14*119 + P15*118 + P16*117 + P12*122 + P13*121 + P14*120 +
P15*119 + P16*118 + P17*117 + P13*122 + P14*121 + P15*120 + P16*119 +
P17*118 + P14*122 + P15*121 + P16*120 + P17*119 + P18*118 + P15*122 +
P16*121 + P17*120 + P18*119 + P16*122 + P17*121 + P18*120 + P19*119 +
P17*122 + P18*121 + P19*120 + P18*122 + P19*121 + P20*120 + P19*122 +
P20*121 + P20*122 + P21*121 + P21*122 + P22*122 - RA*11 - RA*12 - RA*13
- RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 -
RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 -
RA*120 - RA*121 - RA*122 - H*x)/(H*122)
F21a22a=- (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 +
2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 +
2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 +
2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 +
2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 +
2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18
+ 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 +
2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 +
2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 +
2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 +
2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 +
2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 +
2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 +
2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 +
2*P9*110 + 2*P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 +
2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + 2*P1*120 +
2*P2*119 + 2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 +
2*P8*113 + 2*P9*112 + 2*P10*111 + 2*P1*121 + 2*P2*120 + 2*P3*119 +
2*P4*118 + 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 +
2*P10*112 + 2*P11*111 + P1*122 + 2*P2*121 + 2*P3*120 + 2*P4*119 +
2*P5*118 + 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + 2*P10*113 +
2*P11*112 + P2*122 + 2*P3*121 + 2*P4*120 + 2*P5*119 + 2*P6*118 +
2*P7*117 + 2*P8*116 + 2*P9*115 + 2*P10*114 + 2*P11*113 + 2*P12*112 +
P3*122 + 2*P4*121 + 2*P5*120 + 2*P6*119 + 2*P7*118 + 2*P8*117 +
2*P9*116 + 2*P10*115 + 2*P11*114 + 2*P12*113 + P4*122 + 2*P5*121 +
2*P6*120 + 2*P7*119 + 2*P8*118 + 2*P9*117 + 2*P10*116 + 2*P11*115 +
2*P12*114 + 2*P13*113 + P5*122 + 2*P6*121 + 2*P7*120 + 2*P8*119 +
2*P9*118 + 2*P10*117 + 2*P11*116 + 2*P12*115 + 2*P13*114 + P6*122 +
2*P7*121 + 2*P8*120 + 2*P9*119 + 2*P10*118 + 2*P11*117 + 2*P12*116 +
2*P13*115 + 2*P14*114 + P7*122 + 2*P8*121 + 2*P9*120 + 2*P10*119 +
2*P11*118 + 2*P12*117 + 2*P13*116 + 2*P14*115 + P8*122 + 2*P9*121 +
2*P10*120 + 2*P11*119 + 2*P12*118 + 2*P13*117 + 2*P14*116 + 2*P15*115 +
P9*122 + 2*P10*121 + 2*P11*120 + 2*P12*119 + 2*P13*118 + 2*P14*117 +
2*P15*116 + P10*122 + 2*P11*121 + 2*P12*120 + 2*P13*119 + 2*P14*118 +
2*P15*117 + 2*P16*116 + P11*122 + 2*P12*121 + 2*P13*120 + 2*P14*119 +
2*P15*118 + 2*P16*117 + P12*122 + 2*P13*121 + 2*P14*120 + 2*P15*119 +
2*P16*118 + 2*P17*117 + P13*122 + 2*P14*121 + 2*P15*120 + 2*P16*119 +
2*P17*118 + P14*122 + 2*P15*121 + 2*P16*120 + 2*P17*119 + 2*P18*118 +
P15*122 + 2*P16*121 + 2*P17*120 + 2*P18*119 + P16*122 + 2*P17*121 +
2*P18*120 + 2*P19*119 + P17*122 + 2*P18*121 + 2*P19*120 + P18*122 +
2*P19*121 + 2*P20*120 + P19*122 + 2*P20*121 + P20*122 + 2*P21*121 +
P21*122 + P22*122 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 -
2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - 2*RA*112
- 2*RA*113 - 2*RA*114 - 2*RA*115 - 2*RA*116 - 2*RA*117 - 2*RA*118 -
2*RA*119 - 2*RA*120 - 2*RA*121 - RA*122 - H*x)/H

```

F23s23a=(d46*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +
P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +
P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +
P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +
P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +
P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +
P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +
P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +
P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +
P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +
P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +
P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116
+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +
P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112
+ P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 +
P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P2*122 + P3*121 + P4*120
+ P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 + P11*113 +
P12*112 + P3*122 + P4*121 + P5*120 + P6*119 + P7*118 + P8*117 + P9*116
+ P10*115 + P11*114 + P12*113 + P4*122 + P5*121 + P6*120 + P7*119 +
P8*118 + P9*117 + P10*116 + P11*115 + P12*114 + P13*113 + P5*122 +
P6*121 + P7*120 + P8*119 + P9*118 + P10*117 + P11*116 + P12*115 +
P13*114 + P6*122 + P7*121 + P8*120 + P9*119 + P10*118 + P11*117 +
P12*116 + P13*115 + P14*114 + P7*122 + P8*121 + P9*120 + P10*119 +
P11*118 + P12*117 + P13*116 + P14*115 + P8*122 + P9*121 + P10*120 +
P11*119 + P12*118 + P13*117 + P14*116 + P15*115 + P9*122 + P10*121 +
P11*120 + P12*119 + P13*118 + P14*117 + P15*116 + P10*122 + P11*121 +
P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P11*122 + P12*121 +
P13*120 + P14*119 + P15*118 + P16*117 + P12*122 + P13*121 + P14*120 +
P15*119 + P16*118 + P17*117 + P13*122 + P14*121 + P15*120 + P16*119 +
P17*118 + P14*122 + P15*121 + P16*120 + P17*119 + P18*118 + P15*122 +
P16*121 + P17*120 + P18*119 + P16*122 + P17*121 + P18*120 + P19*119 +
P17*122 + P18*121 + P19*120 + P18*122 + P19*121 + P20*120 + P19*122 +
P20*121 + P20*122 + P21*121 + P21*122 + P22*122 - RA*11 - RA*12 - RA*13
- RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 -
RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 -
RA*120 - RA*121 - RA*122 - H*x))/(H*123)
F23s24s=x
F22a24s=(d45*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +
P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +
P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +
P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +
P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +
P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +
P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +
P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +
P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +
P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +
P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +
P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116
+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +
P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112
+ P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 +
P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P1*123 + P2*122 + P3*121
+ P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 +
P11*113 + P12*112 + P2*123 + P3*122 + P4*121 + P5*120 + P6*119 + P7*118

```

+ P8*117 + P9*116 + P10*115 + P11*114 + P12*113 + P3*123 + P4*122 +
P5*121 + P6*120 + P7*119 + P8*118 + P9*117 + P10*116 + P11*115 +
P12*114 + P13*113 + P4*123 + P5*122 + P6*121 + P7*120 + P8*119 + P9*118
+ P10*117 + P11*116 + P12*115 + P13*114 + P5*123 + P6*122 + P7*121 +
P8*120 + P9*119 + P10*118 + P11*117 + P12*116 + P13*115 + P14*114 +
P6*123 + P7*122 + P8*121 + P9*120 + P10*119 + P11*118 + P12*117 +
P13*116 + P14*115 + P7*123 + P8*122 + P9*121 + P10*120 + P11*119 +
P12*118 + P13*117 + P14*116 + P15*115 + P8*123 + P9*122 + P10*121 +
P11*120 + P12*119 + P13*118 + P14*117 + P15*116 + P9*123 + P10*122 +
P11*121 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P10*123 +
P11*122 + P12*121 + P13*120 + P14*119 + P15*118 + P16*117 + P11*123 +
P12*122 + P13*121 + P14*120 + P15*119 + P16*118 + P17*117 + P12*123 +
P13*122 + P14*121 + P15*120 + P16*119 + P17*118 + P13*123 + P14*122 +
P15*121 + P16*120 + P17*119 + P18*118 + P14*123 + P15*122 + P16*121 +
P17*120 + P18*119 + P15*123 + P16*122 + P17*121 + P18*120 + P19*119 +
P16*123 + P17*122 + P18*121 + P19*120 + P17*123 + P18*122 + P19*121 +
P20*120 + P18*123 + P19*122 + P20*121 + P19*123 + P20*122 + P21*121 +
P20*123 + P21*122 + P21*123 + P22*122 + P22*123 + P23*123 - RA*11 -
RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110
- RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118
- RA*119 - RA*120 - RA*121 - RA*122 - RA*123 - H*x))/ (H*123)
F22a23a=- (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 +
2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 +
2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 +
2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 +
2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 +
2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18
+ 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 +
2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 +
2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 +
2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 +
2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 +
2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 +
2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 +
2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 +
2*P9*110 + 2*P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 +
2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + 2*P1*120 +
2*P2*119 + 2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 +
2*P8*113 + 2*P9*112 + 2*P10*111 + 2*P1*121 + 2*P2*120 + 2*P3*119 +
2*P4*118 + 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 +
2*P10*112 + 2*P11*111 + 2*P1*122 + 2*P2*121 + 2*P3*120 + 2*P4*119 +
2*P5*118 + 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + 2*P10*113 +
2*P11*112 + P1*123 + 2*P2*122 + 2*P3*121 + 2*P4*120 + 2*P5*119 +
2*P6*118 + 2*P7*117 + 2*P8*116 + 2*P9*115 + 2*P10*114 + 2*P11*113 +
2*P12*112 + P2*123 + 2*P3*122 + 2*P4*121 + 2*P5*120 + 2*P6*119 +
2*P7*118 + 2*P8*117 + 2*P9*116 + 2*P10*115 + 2*P11*114 + 2*P12*113 +
P3*123 + 2*P4*122 + 2*P5*121 + 2*P6*120 + 2*P7*119 + 2*P8*118 +
2*P9*117 + 2*P10*116 + 2*P11*115 + 2*P12*114 + 2*P13*113 + P4*123 +
2*P5*122 + 2*P6*121 + 2*P7*120 + 2*P8*119 + 2*P9*118 + 2*P10*117 +
2*P11*116 + 2*P12*115 + 2*P13*114 + P5*123 + 2*P6*122 + 2*P7*121 +
2*P8*120 + 2*P9*119 + 2*P10*118 + 2*P11*117 + 2*P12*116 + 2*P13*115 +
2*P14*114 + P6*123 + 2*P7*122 + 2*P8*121 + 2*P9*120 + 2*P10*119 +
2*P11*118 + 2*P12*117 + 2*P13*116 + 2*P14*115 + P7*123 + 2*P8*122 +
2*P9*121 + 2*P10*120 + 2*P11*119 + 2*P12*118 + 2*P13*117 + 2*P14*116 +
2*P15*115 + P8*123 + 2*P9*122 + 2*P10*121 + 2*P11*120 + 2*P12*119 +
2*P13*118 + 2*P14*117 + 2*P15*116 + P9*123 + 2*P10*122 + 2*P11*121 +
2*P12*120 + 2*P13*119 + 2*P14*118 + 2*P15*117 + 2*P16*116 + P10*123 +

$2*P11*122 + 2*P12*121 + 2*P13*120 + 2*P14*119 + 2*P15*118 + 2*P16*117 +$
 $P11*123 + 2*P12*122 + 2*P13*121 + 2*P14*120 + 2*P15*119 + 2*P16*118 +$
 $2*P17*117 + P12*123 + 2*P13*122 + 2*P14*121 + 2*P15*120 + 2*P16*119 +$
 $2*P17*118 + P13*123 + 2*P14*122 + 2*P15*121 + 2*P16*120 + 2*P17*119 +$
 $2*P18*118 + P14*123 + 2*P15*122 + 2*P16*121 + 2*P17*120 + 2*P18*119 +$
 $P15*123 + 2*P16*122 + 2*P17*121 + 2*P18*120 + 2*P19*119 + P16*123 +$
 $2*P17*122 + 2*P18*121 + 2*P19*120 + P17*123 + 2*P18*122 + 2*P19*121 +$
 $2*P20*120 + P18*123 + 2*P19*122 + 2*P20*121 + P19*123 + 2*P20*122 +$
 $2*P21*121 + P20*123 + 2*P21*122 + P21*123 + 2*P22*122 + P22*123 +$
 $P23*123 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 -$
 $2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - 2*RA*112 - 2*RA*113$
 $- 2*RA*114 - 2*RA*115 - 2*RA*116 - 2*RA*117 - 2*RA*118 - 2*RA*119 -$
 $2*RA*120 - 2*RA*121 - 2*RA*122 - RA*123 - H*x)/H$
 $F24s24a=(d48*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +$
 $P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +$
 $P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116$
 $+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +$
 $P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112$
 $+ P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 +$
 $P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P1*123 + P2*122 + P3*121$
 $+ P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 +$
 $P11*113 + P12*112 + P2*123 + P3*122 + P4*121 + P5*120 + P6*119 + P7*118$
 $+ P8*117 + P9*116 + P10*115 + P11*114 + P12*113 + P3*123 + P4*122 +$
 $P5*121 + P6*120 + P7*119 + P8*118 + P9*117 + P10*116 + P11*115 +$
 $P12*114 + P13*113 + P4*123 + P5*122 + P6*121 + P7*120 + P8*119 + P9*118$
 $+ P10*117 + P11*116 + P12*115 + P13*114 + P5*123 + P6*122 + P7*121 +$
 $P8*120 + P9*119 + P10*118 + P11*117 + P12*116 + P13*115 + P14*114 +$
 $P6*123 + P7*122 + P8*121 + P9*120 + P10*119 + P11*118 + P12*117 +$
 $P13*116 + P14*115 + P7*123 + P8*122 + P9*121 + P10*120 + P11*119 +$
 $P12*118 + P13*117 + P14*116 + P15*115 + P8*123 + P9*122 + P10*121 +$
 $P11*120 + P12*119 + P13*118 + P14*117 + P15*116 + P9*123 + P10*122 +$
 $P11*121 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P10*123 +$
 $P11*122 + P12*121 + P13*120 + P14*119 + P15*118 + P16*117 + P11*123 +$
 $P12*122 + P13*121 + P14*120 + P15*119 + P16*118 + P17*117 + P12*123 +$
 $P13*122 + P14*121 + P15*120 + P16*119 + P17*118 + P13*123 + P14*122 +$
 $P15*121 + P16*120 + P17*119 + P18*118 + P14*123 + P15*122 + P16*121 +$
 $P17*120 + P18*119 + P15*123 + P16*122 + P17*121 + P18*120 + P19*119 +$
 $P16*123 + P17*122 + P18*121 + P19*120 + P17*123 + P18*122 + P19*121 +$
 $P20*120 + P18*123 + P19*122 + P20*121 + P19*123 + P20*122 + P21*121 +$
 $P20*123 + P21*122 + P21*123 + P22*122 + P22*123 + P23*123 - RA*11 -$
 $RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110$
 $- RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118$
 $- RA*119 - RA*120 - RA*121 - RA*122 - RA*123 - H*x))/(H*124)$
 $F24s25s=x$
 $F23a25s=(d47*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$

P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +
P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +
P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +
P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +
P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +
P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +
P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +
P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +
P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116
+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +
P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112
+ P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 +
P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P1*123 + P2*122 + P3*121
+ P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 +
P11*113 + P12*112 + P1*124 + P2*123 + P3*122 + P4*121 + P5*120 + P6*119
+ P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + P12*113 + P2*124 +
P3*123 + P4*122 + P5*121 + P6*120 + P7*119 + P8*118 + P9*117 + P10*116
+ P11*115 + P12*114 + P13*113 + P13*113 + P3*124 + P4*123 + P5*122 + P6*121 +
P7*120 + P8*119 + P9*118 + P10*117 + P11*116 + P12*115 + P13*114 +
P4*124 + P5*123 + P6*122 + P7*121 + P8*120 + P9*119 + P10*118 + P11*117
+ P12*116 + P13*115 + P14*114 + P5*124 + P6*123 + P7*122 + P8*121 +
P9*120 + P10*119 + P11*118 + P12*117 + P13*116 + P14*115 + P6*124 +
P7*123 + P8*122 + P9*121 + P10*120 + P11*119 + P12*118 + P13*117 +
P14*116 + P15*115 + P7*124 + P8*123 + P9*122 + P10*121 + P11*120 +
P12*119 + P13*118 + P14*117 + P15*116 + P8*124 + P9*123 + P10*122 +
P11*121 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P9*124 +
P10*123 + P11*122 + P12*121 + P13*120 + P14*119 + P15*118 + P16*117 +
P10*124 + P11*123 + P12*122 + P13*121 + P14*120 + P15*119 + P16*118 +
P17*117 + P11*124 + P12*123 + P13*122 + P14*121 + P15*120 + P16*119 +
P17*118 + P12*124 + P13*123 + P14*122 + P15*121 + P16*120 + P17*119 +
P18*118 + P13*124 + P14*123 + P15*122 + P16*121 + P17*120 + P18*119 +
P14*124 + P15*123 + P16*122 + P17*121 + P18*120 + P19*119 + P15*124 +
P16*123 + P17*122 + P18*121 + P19*120 + P16*124 + P17*123 + P18*122 +
P19*121 + P20*120 + P17*124 + P18*123 + P19*122 + P20*121 + P18*124 +
P19*123 + P20*122 + P21*121 + P19*124 + P20*123 + P21*122 + P20*124 +
P21*123 + P22*122 + P21*124 + P22*123 + P22*124 + P23*123 + P23*124 +
P24*124 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18
- RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116
- RA*117 - RA*118 - RA*119 - RA*120 - RA*121 - RA*122 - RA*123 - RA*124
- H*x) / (H*124)
F23a24a=- (2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 +
2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 +
2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 +
2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 +
2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 +
2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18
+ 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 +
2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 +
2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 +
2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 +
2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 +
2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 +
2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 +
2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 +
2*P9*110 + 2*P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 +
2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + 2*P1*120 +

$2*P2*119 + 2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 +$
 $2*P8*113 + 2*P9*112 + 2*P10*111 + 2*P1*121 + 2*P2*120 + 2*P3*119 +$
 $2*P4*118 + 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 +$
 $2*P10*112 + 2*P11*111 + 2*P1*122 + 2*P2*121 + 2*P3*120 + 2*P4*119 +$
 $2*P5*118 + 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + 2*P10*113 +$
 $2*P11*112 + 2*P1*123 + 2*P2*122 + 2*P3*121 + 2*P4*120 + 2*P5*119 +$
 $2*P6*118 + 2*P7*117 + 2*P8*116 + 2*P9*115 + 2*P10*114 + 2*P11*113 +$
 $2*P12*112 + P1*124 + 2*P2*123 + 2*P3*122 + 2*P4*121 + 2*P5*120 +$
 $2*P6*119 + 2*P7*118 + 2*P8*117 + 2*P9*116 + 2*P10*115 + 2*P11*114 +$
 $2*P12*113 + P2*124 + 2*P3*123 + 2*P4*122 + 2*P5*121 + 2*P6*120 +$
 $2*P7*119 + 2*P8*118 + 2*P9*117 + 2*P10*116 + 2*P11*115 + 2*P12*114 +$
 $2*P13*113 + P3*124 + 2*P4*123 + 2*P5*122 + 2*P6*121 + 2*P7*120 +$
 $2*P8*119 + 2*P9*118 + 2*P10*117 + 2*P11*116 + 2*P12*115 + 2*P13*114 +$
 $P4*124 + 2*P5*123 + 2*P6*122 + 2*P7*121 + 2*P8*120 + 2*P9*119 +$
 $2*P10*118 + 2*P11*117 + 2*P12*116 + 2*P13*115 + 2*P14*114 + P5*124 +$
 $2*P6*123 + 2*P7*122 + 2*P8*121 + 2*P9*120 + 2*P10*119 + 2*P11*118 +$
 $2*P12*117 + 2*P13*116 + 2*P14*115 + P6*124 + 2*P7*123 + 2*P8*122 +$
 $2*P9*121 + 2*P10*120 + 2*P11*119 + 2*P12*118 + 2*P13*117 + 2*P14*116 +$
 $2*P15*115 + P7*124 + 2*P8*123 + 2*P9*122 + 2*P10*121 + 2*P11*120 +$
 $2*P12*119 + 2*P13*118 + 2*P14*117 + 2*P15*116 + P8*124 + 2*P9*123 +$
 $2*P10*122 + 2*P11*121 + 2*P12*120 + 2*P13*119 + 2*P14*118 + 2*P15*117 +$
 $2*P16*116 + P9*124 + 2*P10*123 + 2*P11*122 + 2*P12*121 + 2*P13*120 +$
 $2*P14*119 + 2*P15*118 + 2*P16*117 + P10*124 + 2*P11*123 + 2*P12*122 +$
 $2*P13*121 + 2*P14*120 + 2*P15*119 + 2*P16*118 + 2*P17*117 + P11*124 +$
 $2*P12*123 + 2*P13*122 + 2*P14*121 + 2*P15*120 + 2*P16*119 + 2*P17*118 +$
 $P12*124 + 2*P13*123 + 2*P14*122 + 2*P15*121 + 2*P16*120 + 2*P17*119 +$
 $2*P18*118 + P13*124 + 2*P14*123 + 2*P15*122 + 2*P16*121 + 2*P17*120 +$
 $2*P18*119 + P14*124 + 2*P15*123 + 2*P16*122 + 2*P17*121 + 2*P18*120 +$
 $2*P19*119 + P15*124 + 2*P16*123 + 2*P17*122 + 2*P18*121 + 2*P19*120 +$
 $P16*124 + 2*P17*123 + 2*P18*122 + 2*P19*121 + 2*P20*120 + P17*124 +$
 $2*P18*123 + 2*P19*122 + 2*P20*121 + P18*124 + 2*P19*123 + 2*P20*122 +$
 $2*P21*121 + P19*124 + 2*P20*123 + 2*P21*122 + P20*124 + 2*P21*123 +$
 $2*P22*122 + P21*124 + 2*P22*123 + P22*124 + 2*P23*123 + P23*124 +$
 $P24*124 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 - 2*RA*16 -$
 $2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - 2*RA*112 - 2*RA*113$
 $- 2*RA*114 - 2*RA*115 - 2*RA*116 - 2*RA*117 - 2*RA*118 - 2*RA*119 -$
 $2*RA*120 - 2*RA*121 - 2*RA*122 - 2*RA*123 - RA*124 - H*x)/H$
 $F25s25a=(d50*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +$
 $P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +$
 $P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +$
 $P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +$
 $P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +$
 $P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +$
 $P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +$
 $P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +$
 $P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +$
 $P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +$
 $P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +$
 $P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +$
 $P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116$
 $+ P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +$
 $P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112$
 $+ P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 +$
 $P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P1*123 + P2*122 + P3*121$
 $+ P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 +$
 $P11*113 + P12*112 + P1*124 + P2*123 + P3*122 + P4*121 + P5*120 + P6*119$
 $+ P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + P12*113 + P2*124 +$

P3*123 + P4*122 + P5*121 + P6*120 + P7*119 + P8*118 + P9*117 + P10*116
 + P11*115 + P12*114 + P13*113 + P3*124 + P4*123 + P5*122 + P6*121 +
 P7*120 + P8*119 + P9*118 + P10*117 + P11*116 + P12*115 + P13*114 +
 P4*124 + P5*123 + P6*122 + P7*121 + P8*120 + P9*119 + P10*118 + P11*117
 + P12*116 + P13*115 + P14*114 + P5*124 + P6*123 + P7*122 + P8*121 +
 P9*120 + P10*119 + P11*118 + P12*117 + P13*116 + P14*115 + P6*124 +
 P7*123 + P8*122 + P9*121 + P10*120 + P11*119 + P12*118 + P13*117 +
 P14*116 + P15*115 + P7*124 + P8*123 + P9*122 + P10*121 + P11*120 +
 P12*119 + P13*118 + P14*117 + P15*116 + P8*124 + P9*123 + P10*122 +
 P11*121 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P9*124 +
 P10*123 + P11*122 + P12*121 + P13*120 + P14*119 + P15*118 + P16*117 +
 P10*124 + P11*123 + P12*122 + P13*121 + P14*120 + P15*119 + P16*118 +
 P17*117 + P11*124 + P12*123 + P13*122 + P14*121 + P15*120 + P16*119 +
 P17*118 + P12*124 + P13*123 + P14*122 + P15*121 + P16*120 + P17*119 +
 P18*118 + P13*124 + P14*123 + P15*122 + P16*121 + P17*120 + P18*119 +
 P14*124 + P15*123 + P16*122 + P17*121 + P18*120 + P19*119 + P15*124 +
 P16*123 + P17*122 + P18*121 + P19*120 + P16*124 + P17*123 + P18*122 +
 P19*121 + P20*120 + P17*124 + P18*123 + P19*122 + P20*121 + P18*124 +
 P19*123 + P20*122 + P21*121 + P19*124 + P20*123 + P21*122 + P20*124 +
 P21*123 + P22*122 + P21*124 + P22*123 + P22*124 + P23*123 + P23*124 +
 P24*124 - RA*11 - RA*12 - RA*13 - RA*14 - RA*15 - RA*16 - RA*17 - RA*18
 - RA*19 - RA*110 - RA*111 - RA*112 - RA*113 - RA*114 - RA*115 - RA*116
 - RA*117 - RA*118 - RA*119 - RA*120 - RA*121 - RA*122 - RA*123 - RA*124
 - H*x)/(H*125)

F25s26s=x

F24a26s=(d49*(P1*11 + P1*12 + P1*13 + P2*12 + P1*14 + P2*13 + P1*15 +
 P2*14 + P3*13 + P1*16 + P2*15 + P3*14 + P1*17 + P2*16 + P3*15 + P4*14 +
 P1*18 + P2*17 + P3*16 + P4*15 + P1*19 + P2*18 + P3*17 + P4*16 + P5*15 +
 P1*110 + P2*19 + P3*18 + P4*17 + P5*16 + P1*111 + P2*110 + P3*19 +
 P4*18 + P5*17 + P6*16 + P1*112 + P2*111 + P3*110 + P4*19 + P5*18 +
 P6*17 + P1*113 + P2*112 + P3*111 + P4*110 + P5*19 + P6*18 + P7*17 +
 P1*114 + P2*113 + P3*112 + P4*111 + P5*110 + P6*19 + P7*18 + P1*115 +
 P2*114 + P3*113 + P4*112 + P5*111 + P6*110 + P7*19 + P8*18 + P1*116 +
 P2*115 + P3*114 + P4*113 + P5*112 + P6*111 + P7*110 + P8*19 + P1*117 +
 P2*116 + P3*115 + P4*114 + P5*113 + P6*112 + P7*111 + P8*110 + P9*19 +
 P1*118 + P2*117 + P3*116 + P4*115 + P5*114 + P6*113 + P7*112 + P8*111 +
 P9*110 + P1*119 + P2*118 + P3*117 + P4*116 + P5*115 + P6*114 + P7*113 +
 P8*112 + P9*111 + P10*110 + P1*120 + P2*119 + P3*118 + P4*117 + P5*116
 + P6*115 + P7*114 + P8*113 + P9*112 + P10*111 + P1*121 + P2*120 +
 P3*119 + P4*118 + P5*117 + P6*116 + P7*115 + P8*114 + P9*113 + P10*112
 + P11*111 + P1*122 + P2*121 + P3*120 + P4*119 + P5*118 + P6*117 +
 P7*116 + P8*115 + P9*114 + P10*113 + P11*112 + P1*123 + P2*122 + P3*121
 + P4*120 + P5*119 + P6*118 + P7*117 + P8*116 + P9*115 + P10*114 +
 P11*113 + P12*112 + P1*124 + P2*123 + P3*122 + P4*121 + P5*120 + P6*119
 + P7*118 + P8*117 + P9*116 + P10*115 + P11*114 + P12*113 + P1*125 +
 P2*124 + P3*123 + P4*122 + P5*121 + P6*120 + P7*119 + P8*118 + P9*117 +
 P10*116 + P11*115 + P12*114 + P13*113 + P2*125 + P3*124 + P4*123 +
 P5*122 + P6*121 + P7*120 + P8*119 + P9*118 + P10*117 + P11*116 +
 P12*115 + P13*114 + P3*125 + P4*124 + P5*123 + P6*122 + P7*121 + P8*120
 + P9*119 + P10*118 + P11*117 + P12*116 + P13*115 + P14*114 + P4*125 +
 P5*124 + P6*123 + P7*122 + P8*121 + P9*120 + P10*119 + P11*118 +
 P12*117 + P13*116 + P14*115 + P5*125 + P6*124 + P7*123 + P8*122 +
 P9*121 + P10*120 + P11*119 + P12*118 + P13*117 + P14*116 + P15*115 +
 P6*125 + P7*124 + P8*123 + P9*122 + P10*121 + P11*120 + P12*119 +
 P13*118 + P14*117 + P15*116 + P7*125 + P8*124 + P9*123 + P10*122 +
 P11*121 + P12*120 + P13*119 + P14*118 + P15*117 + P16*116 + P8*125 +
 P9*124 + P10*123 + P11*122 + P12*121 + P13*120 + P14*119 + P15*118 +

$$\begin{aligned}
& P16*117 + P9*125 + P10*124 + P11*123 + P12*122 + P13*121 + P14*120 + \\
& P15*119 + P16*118 + P17*117 + P10*125 + P11*124 + P12*123 + P13*122 + \\
& P14*121 + P15*120 + P16*119 + P17*118 + P11*125 + P12*124 + P13*123 + \\
& P14*122 + P15*121 + P16*120 + P17*119 + P18*118 + P12*125 + P13*124 + \\
& P14*123 + P15*122 + P16*121 + P17*120 + P18*119 + P13*125 + P14*124 + \\
& P15*123 + P16*122 + P17*121 + P18*120 + P19*119 + P14*125 + P15*124 + \\
& P16*123 + P17*122 + P18*121 + P19*120 + P15*125 + P16*124 + P17*123 + \\
& P18*122 + P19*121 + P20*120 + P16*125 + P17*124 + P18*123 + P19*122 + \\
& P20*121 + P17*125 + P18*124 + P19*123 + P20*122 + P21*121 + P18*125 + \\
& P19*124 + P20*123 + P21*122 + P19*125 + P20*124 + P21*123 + P22*122 + \\
& P20*125 + P21*124 + P22*123 + P21*125 + P22*124 + P23*123 + P22*125 + \\
& P23*124 + P23*125 + P24*124 + P24*125 + P25*125 - RA*11 - RA*12 - RA*13 \\
& - RA*14 - RA*15 - RA*16 - RA*17 - RA*18 - RA*19 - RA*110 - RA*111 - \\
& RA*112 - RA*113 - RA*114 - RA*115 - RA*116 - RA*117 - RA*118 - RA*119 - \\
& RA*120 - RA*121 - RA*122 - RA*123 - RA*124 - RA*125 - H*x)) / (H*125) \\
& F24a25a = -(2*P1*11 + 2*P1*12 + 2*P1*13 + 2*P2*12 + 2*P1*14 + 2*P2*13 + \\
& 2*P1*15 + 2*P2*14 + 2*P3*13 + 2*P1*16 + 2*P2*15 + 2*P3*14 + 2*P1*17 + \\
& 2*P2*16 + 2*P3*15 + 2*P4*14 + 2*P1*18 + 2*P2*17 + 2*P3*16 + 2*P4*15 + \\
& 2*P1*19 + 2*P2*18 + 2*P3*17 + 2*P4*16 + 2*P5*15 + 2*P1*110 + 2*P2*19 + \\
& 2*P3*18 + 2*P4*17 + 2*P5*16 + 2*P1*111 + 2*P2*110 + 2*P3*19 + 2*P4*18 + \\
& 2*P5*17 + 2*P6*16 + 2*P1*112 + 2*P2*111 + 2*P3*110 + 2*P4*19 + 2*P5*18 \\
& + 2*P6*17 + 2*P1*113 + 2*P2*112 + 2*P3*111 + 2*P4*110 + 2*P5*19 + \\
& 2*P6*18 + 2*P7*17 + 2*P1*114 + 2*P2*113 + 2*P3*112 + 2*P4*111 + \\
& 2*P5*110 + 2*P6*19 + 2*P7*18 + 2*P1*115 + 2*P2*114 + 2*P3*113 + \\
& 2*P4*112 + 2*P5*111 + 2*P6*110 + 2*P7*19 + 2*P8*18 + 2*P1*116 + \\
& 2*P2*115 + 2*P3*114 + 2*P4*113 + 2*P5*112 + 2*P6*111 + 2*P7*110 + \\
& 2*P8*19 + 2*P1*117 + 2*P2*116 + 2*P3*115 + 2*P4*114 + 2*P5*113 + \\
& 2*P6*112 + 2*P7*111 + 2*P8*110 + 2*P9*19 + 2*P1*118 + 2*P2*117 + \\
& 2*P3*116 + 2*P4*115 + 2*P5*114 + 2*P6*113 + 2*P7*112 + 2*P8*111 + \\
& 2*P9*110 + 2*P1*119 + 2*P2*118 + 2*P3*117 + 2*P4*116 + 2*P5*115 + \\
& 2*P6*114 + 2*P7*113 + 2*P8*112 + 2*P9*111 + 2*P10*110 + 2*P1*120 + \\
& 2*P2*119 + 2*P3*118 + 2*P4*117 + 2*P5*116 + 2*P6*115 + 2*P7*114 + \\
& 2*P8*113 + 2*P9*112 + 2*P10*111 + 2*P1*121 + 2*P2*120 + 2*P3*119 + \\
& 2*P4*118 + 2*P5*117 + 2*P6*116 + 2*P7*115 + 2*P8*114 + 2*P9*113 + \\
& 2*P10*112 + 2*P11*111 + 2*P1*122 + 2*P2*121 + 2*P3*120 + 2*P4*119 + \\
& 2*P5*118 + 2*P6*117 + 2*P7*116 + 2*P8*115 + 2*P9*114 + 2*P10*113 + \\
& 2*P11*112 + 2*P1*123 + 2*P2*122 + 2*P3*121 + 2*P4*120 + 2*P5*119 + \\
& 2*P6*118 + 2*P7*117 + 2*P8*116 + 2*P9*115 + 2*P10*114 + 2*P11*113 + \\
& 2*P12*112 + 2*P1*124 + 2*P2*123 + 2*P3*122 + 2*P4*121 + 2*P5*120 + \\
& 2*P6*119 + 2*P7*118 + 2*P8*117 + 2*P9*116 + 2*P10*115 + 2*P11*114 + \\
& 2*P12*113 + P1*125 + 2*P2*124 + 2*P3*123 + 2*P4*122 + 2*P5*121 + \\
& 2*P6*120 + 2*P7*119 + 2*P8*118 + 2*P9*117 + 2*P10*116 + 2*P11*115 + \\
& 2*P12*114 + 2*P13*113 + P2*125 + 2*P3*124 + 2*P4*123 + 2*P5*122 + \\
& 2*P6*121 + 2*P7*120 + 2*P8*119 + 2*P9*118 + 2*P10*117 + 2*P11*116 + \\
& 2*P12*115 + 2*P13*114 + P3*125 + 2*P4*124 + 2*P5*123 + 2*P6*122 + \\
& 2*P7*121 + 2*P8*120 + 2*P9*119 + 2*P10*118 + 2*P11*117 + 2*P12*116 + \\
& 2*P13*115 + 2*P14*114 + P4*125 + 2*P5*124 + 2*P6*123 + 2*P7*122 + \\
& 2*P8*121 + 2*P9*120 + 2*P10*119 + 2*P11*118 + 2*P12*117 + 2*P13*116 + \\
& 2*P14*115 + P5*125 + 2*P6*124 + 2*P7*123 + 2*P8*122 + 2*P9*121 + \\
& 2*P10*120 + 2*P11*119 + 2*P12*118 + 2*P13*117 + 2*P14*116 + 2*P15*115 + \\
& P6*125 + 2*P7*124 + 2*P8*123 + 2*P9*122 + 2*P10*121 + 2*P11*120 + \\
& 2*P12*119 + 2*P13*118 + 2*P14*117 + 2*P15*116 + P7*125 + 2*P8*124 + \\
& 2*P9*123 + 2*P10*122 + 2*P11*121 + 2*P12*120 + 2*P13*119 + 2*P14*118 + \\
& 2*P15*117 + 2*P16*116 + P8*125 + 2*P9*124 + 2*P10*123 + 2*P11*122 + \\
& 2*P12*121 + 2*P13*120 + 2*P14*119 + 2*P15*118 + 2*P16*117 + P9*125 + \\
& 2*P10*124 + 2*P11*123 + 2*P12*122 + 2*P13*121 + 2*P14*120 + 2*P15*119 + \\
& 2*P16*118 + 2*P17*117 + P10*125 + 2*P11*124 + 2*P12*123 + 2*P13*122 +
\end{aligned}$$


```

2*P14*121 + 2*P15*120 + 2*P16*119 + 2*P17*118 + P11*125 + 2*P12*124 +
2*P13*123 + 2*P14*122 + 2*P15*121 + 2*P16*120 + 2*P17*119 + 2*P18*118 +
P12*125 + 2*P13*124 + 2*P14*123 + 2*P15*122 + 2*P16*121 + 2*P17*120 +
2*P18*119 + P13*125 + 2*P14*124 + 2*P15*123 + 2*P16*122 + 2*P17*121 +
2*P18*120 + 2*P19*119 + P14*125 + 2*P15*124 + 2*P16*123 + 2*P17*122 +
2*P18*121 + 2*P19*120 + P15*125 + 2*P16*124 + 2*P17*123 + 2*P18*122 +
2*P19*121 + 2*P20*120 + P16*125 + 2*P17*124 + 2*P18*123 + 2*P19*122 +
2*P20*121 + P17*125 + 2*P18*124 + 2*P19*123 + 2*P20*122 + 2*P21*121 +
P18*125 + 2*P19*124 + 2*P20*123 + 2*P21*122 + P19*125 + 2*P20*124 +
2*P21*123 + 2*P22*122 + P20*125 + 2*P21*124 + 2*P22*123 + P21*125 +
2*P22*124 + 2*P23*123 + P22*125 + 2*P23*124 + P23*125 + 2*P24*124 +
P24*125 + P25*125 - 2*RA*11 - 2*RA*12 - 2*RA*13 - 2*RA*14 - 2*RA*15 -
2*RA*16 - 2*RA*17 - 2*RA*18 - 2*RA*19 - 2*RA*110 - 2*RA*111 - 2*RA*112
- 2*RA*113 - 2*RA*114 - 2*RA*115 - 2*RA*116 - 2*RA*117 - 2*RA*118 -
2*RA*119 - 2*RA*120 - 2*RA*121 - 2*RA*122 - 2*RA*123 - 2*RA*124 -
RA*125 - H*x)/H

```

```

y=max([F1s1a; F1s2s; FA2s; FA1a; FA1s; F2s2a; F2s3s; F1a3s; F1a2a;
F3s3a; F3s4s; F2a4s; F2a3a; F4s4a; F4s5s; F3a5s; F3a4a; F5s5a; F5s6s;
F4a6s; F4a5a; F6s6a; F6s7s; F5a7s; F5a6a; F7s7a; F7s8s; F6a8s; F6a7a;
F8s8a; F8s9s; F7a9s; F7a8a; F9s9a; F9s10s; F8a10s; F8a9a; F10s10a;
F10s11s; F9a11s; F9a10a; F11s11a; F11s12s; F10a12s; F10a11a; F12s12a;
F12s13s; F11a13s; F11a12a; F13s13a; F13s14s; F12a14s; F12a13a; F14s14a;
F14s15s; F13a15s; F13a14a; F15s15a; F15s16s; F14a16s; F14a15a; F16s16a;
F16s17s; F15a17s; F15a16a; F17s17a; F17s18s; F16a18s; F16a17a; F18s18a;
F18s19s; F17a19s; F17a18a; F19s19a; F19s20s; F18a20s; F18a19a; F20s20a;
F20s21s; F19a21s; F19a20a; F21s21a; F21s22s; F20a22s; F20a21a; F22s22a;
F22s23s; F21a23s; F21a22a; F23s23a; F23s24s; F22a24s; F22a23a; F24s24a;
F24s25s; F23a25s; F23a24a; F25s25a; F25s26s; F24a26s; F24a25a]);

```


Anexo 5 Programación (Pratt)

```
%Borrando la consola y la memoria
clear all; clc;

% Estableciendo las opciones
options = gaoptimset(@ga);
options = gaoptimset(options,...
    'CrossoverFcn', @crossoversinglepoint, ...
    'Display', 'iter', ...
    'Generations', 100, ...
    'MutationFcn', @mutationadaptfeasible, ...
    'PopulationSize', 50, ...
    'SelectionFcn', @selectiontournament);

% Ejecutando el proceso de optimización
time = now;

[x_opt, f_val] =gamultiobj(@Pratt,105 , [], [], [], [], [1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20], options);
time = (now - time)*24*60*60;
```

Anexo 6 Programación (Vertical)

```
%Borrando la consola y la memoria
clear all; clc;

% Estableciendo las opciones
options = gaoptimset(@ga);
options = gaoptimset(options,...
    'CrossoverFcn', @crossoversinglepoint, ...
    'Display', 'iter', ...
    'Generations', 100, ...
    'MutationFcn', @mutationadaptfeasible, ...
    'PopulationSize', 50, ...
    'SelectionFcn', @selectiontournament);

% Ejecutando el proceso de optimización
time = now;

[x_opt, f_val] = gamultiobj(@Vertical,77 , [], [], [], [], [1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1],
[20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20], options);
time = (now - time)*24*60*60
```

