Práctica 2. Generación de horarios mediante algoritmos evolutivos

Autores: Miruna Andreea Gheata,
Pablo González Maya,
Mateu Jover Mulet
Segunda práctica de la asignatura de Inteligencia Computacional del Máster Universitario en Sistemas Inteligentes (MUSI)
Universitat de les Illes Balears
07122 Palma, Islas Baleares, España
miruna.gheata1@estudiant.uib.cat
pablo.gonzalez1@estudiant.uib.cat
mateu.jover@gmail.com

I. Definición del problema

La construcción de los horarios de clase dista de ser un problema trivial debido a la gran cantidad de variables que se deben tener en cuenta: horario lectivo, grupos de alumnos, profesores disponibles, asignaturas, etc. Encontrar solución(es) que no generen conflictos entre estas variables es muy costoso para que se pueda realizar manualmente. Por eso, hoy en día este problema se ha abordado empleando técnicas y algoritmos computacionales que pretenden crear el horario considerando todas estas variables y restricciones.

Uno de estos algoritmos es el **algoritmo evolutivo** o genético. Se trata de un algoritmo que se inspira en la evolución darwiniana tal como la conocemos. A partir de una población de individuos (posibles soluciones) se definen una serie de pasos para crear unos elementos de descendencia, que tenderán a ser mejores a cada generación. Los individuos de esta población están caracterizados por una información que denominamos genes. Estos genes representan las características del individuo, al igual que en genética. En el contexto de la computación evolutiva, los genes codifican la adecuación o *fitness* del individuo. En los problemas de optimización, se buscará maximizar *fitness*, por tanto, encontrar el individuo de mejor genética.

El objetivo de este algoritmo es obtener al menos un individuo que cumpla con los requisitos del problema, es decir, que sea la solución cumpliendo las restricciones necesarias sin recurrir a la exploración exhaustiva del espacio de búsqueda. Los problemas a resolver mediante estas técnicas evolutivas se caracterizan por un conjunto de posibles soluciones inabarcable tanto en términos formales como en términos computacionales evaluando todas las posibilidades.

El presente documento aborda los diferentes aspectos de implementación de un algoritmo evolutivo para resolver un problema. No solo es importante reflejar que se encuentra una solución, sino las diversas opciones que aparecen durante el trabajo. El problema a tratar es la creación de un horario para un centro educativo de secundaria y bachillerato. Para modelar la solución se consideran las siguientes variables como *inputs* al algoritmo:

1. Los **cursos** o grupos de alumnos que existen.
2. El número **total de horas semanales de cada clase**.
3. Una **lista de asignaturas** que se impartirán en cada clase.
4. Las **horas semanales dedicadas para cada asignatura**.
5. El **profesor** asignado a cada asignatura.
6. La **disponibilidad horaria** de cada profesor.
7. Las **franjas horarias semanales** disponibles para impartir las asignaturas.

La salida del algoritmo será **el horario semanal de cada curso**. Supondremos que el centro establece un aula fija para cada clase de manera que el algoritmo no tiene que asignar aulas.

I-A. *Modelo matemático*

Las variables enumeradas en el apartado anterior se definen como:

- P es el conjunto de profesores.
- C es el conjunto de cursos.
- A es el conjunto de asignaturas.
- D es el conjunto de días en el horario.
- H es el conjunto de horas lectivas.
- L es el conjunto de asignaturas impartidas por sus correspondientes profesores en una hora de un día lectivo determinado.

Las estructuras de control de los elementos del problema se definen como:

- $Time_Slot[d][h]$ como el *slot* horario que corresponde a la hora h del día lectivo d. Los elementos de este conjunto se identificarán por ts.
- $Clase[a, p]$ define la asignatura a y el profesor que la imparte p que forman la clase $l \in L$.

- $Curso_Clase_Hora_Dia[c][l][ts]$ define la clase l de la asignatura a impartida por el profesor p en el time slot ts (a la hora h el día d) en el curso c.
- $Profesor_Total_Horas[t]$ es el total de horas semanales al profesor $t \in T$.
- $Profesor_Dia_Hora[p][ts]$ define la disponibilidad del profesor t en la hora h del día d.
  1. Si es igual a 1 es que el profesor p tiene una asignatura a en el *slot* horario ts.
  2. Si es igual a 0 es que el profesor p está disponible en en el *slot* horario ts.
  3. Si es igual a -1 es que el profesor p no está disponible (fuera del instituto) en el *slot* horario ts.
- $Curso_Total_Horas[c]$ es el total de horas de clase asignadas al curso $c \in C$.
- $Curso_Asignaturas[c]$ define las asignaturas $a_1, a_2 \ldots a_n$ que el curso c tiene en una semana, donde n es el total de asignaturas impartidas en la clase c.

II. Diseño e implementación del algoritmo evolutivo

Para la implementación del algoritmo evolutivo es necesario definir una serie de conceptos y variables de control para que el algoritmo pueda encontrar una solución. En este apartado se definen las restricciones *hard* y *soft* del problema así como también los diferentes componentes del algoritmo, entre ellos la representación de los individuos, la función de *fitness* y los operadores de mutación y recombinación.

II-A. Restricciones

Las restricciones del problema son las variables de control que el algoritmo tendrá que analizar para determinar si una solución es correcta. Estas restricciones son de 2 tipos:

1. **Restricciones hard**: son aquellas que necesariamente se deben cumplir para que la solución obtenida sea válida.
2. **Restricciones soft**: son unas restricciones que penalizan la solución final pero que no condicionan su validez.

A continuación se enumeran las diferentes restricciones que se han considerado. Cada una de ellas tiene asignada un identificador único para facilitar su referencia en las próximas secciones.

Restricciones hard

Para este problema se han definido **7 restricciones hard**:

- `H1`: Un grupo de alumnos no puede tener más de una asignatura a la misma hora.
- `H2`: Un profesor no puede impartir más de una asignatura a la misma hora.
- `H3`: No se debe superar la duración semanal de cada asignatura.
- `H4`: Una asignatura sólo se puede realizar dentro del horario lectivo establecido.
- `H5`: Los profesores sólo pueden dar clase en sus disponibilidades horarias.
- `H6`: Un grupo de alumnos sólo puede realizar las asignaturas que tiene asignadas.
- `H7`: Un grupo de alumnos no puede superar el total de horas semanales establecidas para las clases.

Restricciones soft

Para este problema se han definido **5 restricciones soft**:

- `S1`: Un grupo de alumnos debe tener pocas “horas sueltas“ entre las asignaturas.
- `S2`: Una asignatura no se debe impartir en días consecutivos.
- `S3`: Las “horas sueltas“ de un grupo de alumnos se deben poner al final del día.
- `S4`: Los profesores deben tener el mínimo número de “horas sueltas“ durante el día.
- `S5`: El total de “horas sueltas“ que tiene un profesor se deben distribuir uniformemente entre los días que está disponible.

II-B. Componentes del algoritmo evolutivo

El algoritmo evolutivo consta de diferentes elementos necesarios para su implementación. En este apartado se detalla la implementación de cada elemento para este proyecto en concreto.

Representación de los individuos

La estructura empleada en esta implementación es la de una tabla de $m \times n$ dimensiones, donde m son los diferentes cursos para los cuales se quiere definir un horario y la n son los diferentes *slots* horarios que existen en una semana lectiva. El total de slots viene dado por número de horas diarias de clase $\times$ total de días semanales de clase. Esta estructura se inspira en la implementación propuesta en [2]. En la Fig. 1 se muestra dicha representación.

	Time Slot 1	**Time Slot 2**	...	**Time slot m**
Course 1	$Lesson_{11}$	$Lesson_{12}$	...	$Lesson_{1m}$
Course 2	$Lesson_{21}$	$Lesson_{22}$	...	$Lesson_{2m}$
...	...	...	...	...
Course n	$Lesson_{n1}$	$Lesson_{n2}$	...	$Lesson_{nm}$

Figura 1: Representación de los individuos del problema.

Debido al empleo de esta codificación concreta, muchas de las restricciones *hard* que se han definido ya se cumplen. Estas restricciones son:

1. `H4`: no es posible asignar horas de clase fuera del horario establecido ya que las asignaciones sólo se pueden hacer dentro de los *slots* horarios definidos.
2. `H7`: al igual que la anterior, los *slots* horarios definidos no permiten que una clase haga más horas de clase que las establecidas.

Se referirá a esta estructura como al *timetable* del individuo.

Procedimiento de inicialización

El procedimiento de inicialización sirve para definir la primera población del algoritmo evolutivo. En este caso se generan horarios en los que a cada curso se le asignan todas las horas de clase en posiciones aleatorias de la semana, tantas veces como individuos debe tener la población inicial. En el Algoritmo 1 se muestra el pseudo-código de este proceso de creación de la población inicial.

Algoritmo 1: Inicialización de la población

```
Result: Lista de individuos con una inicialización
        aleatoria de las clases impartidas en cada slot
        horario de cada curso.
Inicializar el total de generaciones calculadas a 0;
for individuo en tamaño_población do
    Inicializar el timetable del individuo con 0 en cada
     celda;
    for curso en total_cursos do
        Seleccionar las clases que se deben realizar en
         este curso;
        for slot in timeslots do
            Asignar de forma aleatoria todas las clases
             hasta que se cumplan todas las horas
             semanales definidas;
        end
    end
end
Incrementar en 1 el total de generaciones calculadas;
```

Debido a esta inicialización se cumplen de inicio y sin ser alterables otras restricciones *hard* propuestas. Estas restricciones son:

1. `H1`: No se puede dar el caso de que en un curso se imparta más de una asignatura en la misma hora.
2. `H3`: Puesto que desde el principio del algoritmo se asignan todas las horas de cada clase, no es posible no alcanzar o superar el número máximo de horas establecidas de las clases.
3. `H6`: No es posible que a un curso se le asignen clases fuera de su plan de estudios ya que sólo se asignan sus clases en la inicialización.

Función de *fitness*

El cálculo de la función de *fitness* consiste en determinar el coste de un individuo concreto a partir de la cantidad de restricciones que incumple.

En los términos del trabajo, es más correcto referirse a la evaluación de los individuos como su coste (ya que se expresa en términos de penalizaciones) que referirse a *fitness*. Se menciona que la función de *fitness* se debe maximizar. Sin embargo, para la definición de este problema en concreto, la función de coste se debe **minimizar** para obtener una buena solución. Esto conceptualmente es lógico: se *maximiza* la calidad de un individuo al *minimizar* su coste de las restricciones incumplidas.

Se asignan pesos para que la relevancia de la restricción se refleje en el cálculo. A las restricciones *hard* se les asigna un peso (*Hard Constraint Weight*, `HCW`) de 1000, y las restricciones *soft* tienen un peso (*Soft Constraint Weight*, `SCW`) de 5. Se les ha otorgado un peso mayor a las restricciones *hard* debido a que determinan en mayor grado que un individuo sea una buena solución, mientras que las restricciones *soft* pueden no cumplirse aún en soluciones válidas; se entienden como imperfecciones de la solución obtenida.

La función de de *fitness* se calcula como la suma de todas los incumplimientos de cada restricción para cada individuo, con su correspondiente peso (`HCW` o `SCW`). El coste de una generación es la suma de los costes individuales. El objetivo del algoritmo es, como se formaliza en la Ecuación 1, la minimización de dicha suma.

$$\begin{aligned}\text{mín}[&\text{profesores_dando_más_de_una_clase_a_la_vez} * \texttt{HCW}\\ &+ \text{profesores_asignados_fuera_de_horario} * \texttt{HCW}\\ &+ \text{número_horas_vacías_por_curso} * \texttt{SCW}\\ &+ \text{clase_impartida_en_días_consecutivos} * \texttt{SCW}\\ &+ \text{horas_vacías_que_no_están_al_final_del_día} * \texttt{SCW}\\ &+ \text{horas_vacías_profesor} * \texttt{SCW}\\ &+ \text{horas_vacías_profesor_no_uniformes} * \texttt{SCW}]\end{aligned} \tag{1}$$

Selección de padres

La selección de padres es el proceso mediante el cual se escogen una parte de los individuos de una generación concreta para poder crear una nueva generación. En la implementación de este proyecto se han considerado dos algoritmos diferentes para la selección de padres: el **algoritmo de la ruleta** (Algoritmo 2) y el **algoritmo por torneo** (Algoritmo 3).

Algoritmo 2: Pseudo-código del algoritmo de selección de padres de la ruleta. (Fuente: [1])

```
/*Dada la probabilidad acumulativa a y teniendo como
  objetivo la selección de λ padres de una generación*/;
miembro = 1;
while miembros ≤ λ do
    Seleccionar un valor aleatorio r entre [0, 1];
    Definir i=0;
    while a_i < r do
        i = i + 1;
    end
    Asignar padres[miembro] = generación[i];
    Asignar miembro = miembro + 1;
end
```

Algoritmo 3: Pseudo-código del algoritmo de selección de padres por torneo. (Fuente: [1])

```
/*Se desea seleccionar λ miembros de una generación
  de μ individuos.*/;
miembro = 1;
while miembros ≤ λ do
    Elegir k individuos aleatorios;
    Comparar los k elementos y seleccionar el mejor
      individuo i;
    Asignar padres[miembro] = i;
    Asignar miembro = miembro + 1;
end
```

Operadores de variación

Los operadores de variación se definen como aquellas modificaciones a los individuos de una generación para obtener una nueva generación. Estos operadores son el operador de **recombinación** y el operador de **mutación**.

El operador de **recombinación** se emplea para poder obtener la descendencia de un conjunto de padres dado. La recombinación consiste en obtener características de una pareja de padres y, al fusionar estas características, se obtiene un nuevo individuo que presenta características de ambos padres. Aplicando este operador a diferentes parejas de padres se consigue obtener la *descendencia* de una generación.

Por otro lado, el operador de **mutación** es una operación que se realiza sobre un individuo de la descendencia. Normalmente, esta operación se realiza de forma aleatoria con una baja probabilidad.

Para esta implementación los operadores se han definido de la siguiente manera:

1. La recombinación consiste en seleccionar los horarios de c cursos del $padre_1$ y los $C - c$ cursos que faltan del $padre_2$. Por lo tanto, el horario resultante del hijo contendrá la organización de los cursos seleccionados de cada uno de los padres. En la Figura 2 se ilustra un ejemplo de recombinación de dos padres.
2. La mutación se realiza al hacer un intercambio de lecciones l_1, l_2 de un curso en concreto. De esta manera, si l_1 se imparte en el *slot* horario s_1 y l_2 en s_2, tras la mutación la lección l_1 se impartiría en s_2 y l_2 en s_1. En la Figura 3 se ilustra un ejemplo de mutación de un descendiente.

II-C. Implementación del programa propuesto

Para llevar a cabo la implementación del proyecto, se ha utilizado Python siguiendo la estructura descrita en el siguiente apartado. El código fuente de la implementación se puede encontrar en [3].

Para la carga de datos, se ha utilizado un fichero de tipo Excel que contiene las horas de clase semanales de cada asignatura, el profesor que imparte cada asignatura, la disponibilidad de cada uno de los profesores, así como los

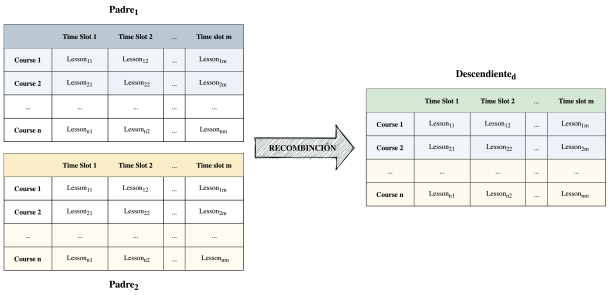


Figura 2: Posible recombinación de dos padres seleccionando 2 cursos del primero y $C - 2$ cursos restantes del segundo.

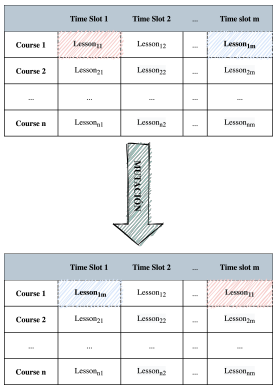


Figura 3: Posible mutación de un descendiente.

datos básicos relativos al problema en cuestión: las horas de clase al día, los días de clase a la semana, las horas de entrada y salida de clase, así como los cursos sobre los que se aplica.

El proyecto se estructura en diversos ficheros, atendiendo a los distintos elementos del problema.

Implementación de los elementos del problema

En primer lugar, se ha creado una clase para cada uno de los elementos del problema: `Clase`, `Curso` y `Profesor`.

Cada `Clase` está compuesta por un nombre, un `Profesor`, los cursos en los que se imparte y las horas por semana.

Adicionalmente, se identificarán las asignaciones de una `Clase` a un *slot* horario concreto como una lección (`Lesson`); los elementos que la componen son el nombre del profesor asignado a la clase, el nombre de la clase y el *slot* horario asignado.

Cada `Curso` está compuesto por un nombre, una lista de clases y las horas semanales en las que se imparte clase.

Cada `Profesor` está compuesto por un nombre, su disponibilidad semanal y las horas a la semana que puede impartir clase. El algoritmo que se utiliza para representar la disponibilidad es el siguiente.

Algoritmo 4: Pseudo-código del algoritmo de disponibilidad de un profesor.

```
Result: La disponibilidad real queda registrada en el
        objeto Profesor.
/*Teniendo la variable disponibilidad formada por 3
  campos: dias_de_la_semana, dias_de_clase y
  horas_por_dia. En este caso, desde el Excel de datos:
  5 días de clase y 7 horas/día*/;
idx = 0;
for disponibilidad_dias_de_clase en disponibilidad do
    for i en [0, horas_por_dia] do
        La disponibilidad real de cada slot es la
          indicada en cada uno de los
          dias_de_la_semana importados previamente
          desde Excel.
        idx = idx + 1
        ;
    end
end
```

Por otro lado, encontramos las clases necesarias para gestionar la construcción del `Cromosoma`, la `Función de fitness` y el `Algoritmo evolutivo`.

El cromosoma

El `Cromosoma` contiene la información de cada individuo del problema. Los parámetros que lo componen son:

- Los cursos para los que se desea computar el horario.
- Las clases disponibles para cada curso.
- Los profesores disponibles en cada curso (cada profesor tiene su disponibilidad a lo largo de la semana).
- El identificador de la generación a la que pertenece el individuo.
- Índice del individuo dentro de esa generación.

Los métodos más importantes de esta clase son la **actualización de la disponibilidad de los profesores** y la **inicialización del *timetable*** (explicado en la Sección II-B).

Al asignar una clase en un *slot* horario ts en concreto, el profesor que imparte dicha clase se puede encontrar en las distintas situaciones:

1. Tiene disponibilidad para dar clase, por lo que actualizamos ts como que está dando clase. A nivel de código, consiste en actualizar $Profesor_Hora_Dia[p][ts] = 0$ a $Profesor_Hora_Dia[p][ts] = 1$.
2. El profesor no está disponible para impartir clase porque está fuera de su horario laboral. Este tipo de situaciones son las que la restricción *hard* `H5` penaliza, por lo que se debe mostrar esta incorrecta asignación para este profesor. A nivel de código, consiste en decrementar $Profesor_Hora_Dia[p][ts]$ en una unidad, de manera que la disponibilidad del profesor en ts será $Profesor_Hora_Dia[p][ts] < -1$. Mencionar que de esta manera se acumula el total de ocasiones en los que al profesor se le ha asignado dar clase en el ts no disponible.

La función de fitness

Para calcular la función de *fitness* se ha calculado el coste de las distintas restricciones indicadas en la Sección II-B.

Coste de `H1`

Para calcular la `H1`, que indica que un profesor no puede impartir más de una clase al mismo tiempo, se comprueba cada ts del cromosoma cada uno de los cursos, y para cada curso se irá recopilando el profesor asignado a cada clase. Si hay profesores que se han asignado más de una vez dentro del asignados dentro de un ts, se sumará una penalización. Matemáticamente, y sabiendo que $l = [a, p]$ lo que se debe cumplir para esta restricción es

$$\forall ts : Clase_Hora_Dia[l][ts, p_1 \neq p_2 \neq \ldots \neq p_l$$

Coste de `H5`

Para calcular la `H5`, que indica que cada profesor sólo puede impartir clases durante sus horas de disponibilidad, se obtienen todas las lecciones que se imparten a lo largo de la semana y de todos los cursos. Para cada lección se comprueba si la disponibilidad del profesor que imparte dicha clase es positiva. En caso contrario, significa que se le ha asignado esta lección fuera de su horario laboral, y se le sumará una penalización. Matemáticamente, lo que se debe cumplir para esta restricción es

$$\forall p, \forall ts : Profesor_Dia_Hora[p][ts] >= 0$$

Coste de `S1`

Para calcular la `S1`, que indica que un curso debe tener el menor número posible de huecos vacíos durante el día, se recorre para cada curso sus ts y, en cada caso en que no haya ninguna clase asignada, se sumará una penalización. Matemáticamente, lo que se debe cumplir para esta restricción es

$$\forall c : Curso_Asignaturas[c] \neq \varnothing$$

Coste de `S2`

Para calcular la `S2`, que indica que una clase no se puede impartir dos días seguidos ni dos veces dentro de un mismo día, para cada curso se selecciona el rango de horas desde el comienzo de un día hasta el final del día siguiente. A lo largo de ese rango de ts, se va haciendo un recuento de las clases impartidas, y si alguna se repite se suma una penalización. Matemáticamente, lo que se debe cumplir para esta restricción es

$$\forall ts_1, ts_2 : ts_2 - ts_1 = horas_dos_dias_lectivos$$

$$Curso_Clase_Hora_Dia[c][l][ts_1] \neq Curso_Clase_Hora_Dia[c][l][ts_2]$$

Coste de `S3`

Para calcular la `S3`, que indica que las horas sueltas de un curso deben ponerse al final del día, se comprueba para cada curso y para cada día, las horas libres (sin lección) que están colocadas entre la primera hora del día y la penúltima. La suma de estos casos será la penalización.

Coste de `S4`

Para calcular la `S4`, que indica que las horas sueltas de un profesor a lo largo del día deben ser las mínimas posibles, se establece un umbral del 25 % de sus horas de trabajo semanales para indicar el máximo de huecos libres que puede tener un profesor. Para cada profesor del cromosoma, se añadirá una penalización si el total de horas libres que tiene a la semana supera este umbral de horas libres permitidas.

Coste de `S5`

Para calcular la `S5`, que indica que el total de horas sueltas de un profesor debe distribuirse uniformemente entre los días que está disponible, se comprueba para cada profesor las horas que tiene libres a lo largo de la semana y se dividen entre las horas por día para obtener los días a la semana en los que está dando clases. A raíz de esto, se calcula cuántas horas libres al día puede tener un profesor de forma óptima. Finalmente, se calcula para cada día de la semana las horas que tiene libres y, si estas sobrepasan las establecidas como óptimas, se sumará una penalización.

El algoritmo evolutivo

Como bien se ha ido mencionando en las diferentes secciones anteriores, este algoritmo es aquel que se aplica para poder construir un horario. Para su implementación se ha creado una clase `GeneticAlgorithm`. Los métodos relevantes de esta clase son explican a continuación.

Encontrar solución

Aplica el algoritmo evolutivo para encontrar el mejor horario para los datos introducidos. El pseudo-código se puede visualizar en el Algoritmo 5. Los diferentes pasos a seguir son:

1. **Inicialización del algoritmo:** se genera la población inicial y se ordenan los individuos según su coste en orden descendiente.
2. **Añadir costes iniciales:** se utilizan las variables *generation_cost_evolution* y *best_individual_constraints_evolution* para ir acumulando los costes y poder analizar la evolución de la solución. Inicialmente, se introducen en ellas los costes de la primera generación. Estos costes son los que quedan reflejados en las gráficas presentes en la Sección V donde se muestra la evolución de las generaciones y del mejor individuo de cada una de estas.
3. **Definición de los datos del individuo aleatorio**: este individuo se explica en detalle en la Sección II-D.
4. **Iterar tantas veces como máximas generaciones se hayan indicado:**
   - Calcular el coste de la generación anterior para poder compararlo con el coste de la generación actual y determinar cuánta mejor hay en el coste de la nueva generación.
   - Seleccionar padres usando el tipo de selección elegido (Roulette o Tournament).
   - Generar la descendencia.
   - Mutar la descendencia.
   - Evaluar la función de fitness para cada uno de los individuos generados.
   - Seleccionar individuos para la siguiente generación. La siguiente generación contendrá los padres seleccionados y la descendencia creada a partir de estos padres. Cabe destacar que si un padre ya se encuentra en la nueva generación, no se vuelve a añadir.
   - Calcular *fitness* para esta nueva generación.
   - Calcular la mejora comparando el antiguo *fitness* respecto del nuevo.

Algoritmo 5: Pseudo-código del algoritmo evolutivo.

```
Inicializar población inicial
Evaluar cada individuo
while no se alcance el total de generaciones máximas
 do
    Seleccionar padres
    Recombinación de los padres
    Mutación de la descendencia
    Evaluación de los nuevos individuos
    Selección de la nueva descendencia
end
```

Generación de la población inicial

Inicializa tantos individuos como se hayan indicado utilizando la función de generación individuos iniciales (explicada en apartado del `Cromosoma`), calculando, además, el coste de cada uno de ellos mediante la función de *fitness*.

Selección de padres

Selección de padres acorde a una de las técnicas propuestas en la Sección II-B.

Creación de la descendencia

Obtiene descendencia de cada pareja de padres. Para ello, selecciona padres aleatorios cada vez (deben ser diferentes) y genera su descendencia con la función de Generar individuo.

Mutación de la descendencia

Modifica aleatoriamente los descendientes intercambiando clases de al menos un curso. Inicialmente, se calcula un número aleatorio entre 0 y 1 para decidir si se ha de mutar un individuo. Esta decisión viene determinada por el umbral establecido en la ejecución que se esté realizando. Si se debe mutar, se calculará un número aleatorio entre 1 y 3 para decidir

cuántas mutaciones se van a realizar. Para cada una de esas mutaciones, se escogen aleatoriamente dos lecciones (que no sean iguales) y se intercambian.

II-D. Modificación del algoritmo

Por último, una propuesta de modificación en el algoritmo es la **creación de un individuo aleatorio** en aquellas situaciones cuando el coste de la generación no mejore en tres generaciones (es decir, que tres generaciones seguidas tengan el mismo coste). De esta manera, se consigue salir del mínimo local en el que se ha quedado estancado el algoritmo evolutivo al obligar a un individuo totalmente nuevo a entrar dentro de la lista de posibles padres.

III. Evaluación del algoritmo implementado

Una evaluación de la implementación del algoritmo evolutivo representativa requerirá la prueba de diversas combinaciones de parámetros y de *inputs*. Por tanto, se definirán una serie de valores para cada parámetro e *input* y así reflejar el comportamiento del algoritmo bajo diferentes condiciones.

Se propone la variación del procedimiento de selección de padres y de la probabilidad de mutación en cada generación para cada individuo. Con el objetivo de reflejar que el algoritmo obtiene mejores poblaciones a medida que se suceden las generaciones se acota, a dos límites diferentes, el numero de generaciones calculadas. Todas estas variaciones de parámetros se probarán sobre tres diferentes conjuntos de datos de entrada.

1. Selección de padres: ruleta probabilística o selección por torneo.
2. Probabilidad de mutación: 0.01, 0.15, 0.35.
3. Límite de generaciones: 200 ó 500 generaciones.
4. Conjuntos de datos: tres dificultades para el algoritmo *teacher_hour*, *teacher_hour_easy*, *teacher_hour_hard*.

III-A. Formato de los datos de entrada

De acuerdo con la definición del problema, los datos de entrada se caracterizan por:

- Una lista de grupos de alumnos, es decir, de cursos. Por ejemplo, tercero de ESO o primero de bachillerato.
- Un lista representado las asignaturas particulares para cada grupo.
- La disponibilidad para cada profesor a lo largo de una semana en días disponibles y ocupados.

No se tienen en cuenta las horas disponibles para impartir clase ya que la representación de los datos condiciona este aspecto, como se comenta en la Sección II-B.

También es menester comentar algunas de las elipsis realizadas respecto de problemas que consideren todos sus aspectos reales. Se han abstraído los siguientes aspectos atendiendo a que el objetivo no es, estrictamente, dar solución a un problema de generación de horarios sino también familiarizarse de una forma empírica con los algoritmos evolutivos.

- Solo se calcula una semana, donde cada día que aparece es laborable.
- Cada asignatura tiene un profesor asignado.
- No se consideran desdoblamientos de grupos ni más de una línea por curso.
- El tamaño de las aulas y de los grupos de alumnos no es relevante. Se supone que se dispone de los recursos necesarios para cada caso.
- Una asignatura sólo se puede impartir por un único profesor.

III-B. Inputs evaluados

Las tablas de disponibilidad con las que se realizan las pruebas del algoritmo están definidas en las páginas del fichero Excel mencionado en la Sección II-C. Cada página que sirve de input está marcada con los nombres *teacher_hours_easy*, *teacher_hours*, *teacher_hours_hard*, ordenadas en función de su supuesta dificultad para encontrar solución.

Cada hoja del Excel tiene una tabla de disponibilidades con diferentes densidades de no disponibilidad. Para el caso con más holgura, todos los profesores tienen toda la semana disponible. Intuitivamente, será más difícil encontrar horarios posibles como mayor sea la no disponibilidad de los profesores, ya que las combinaciones de posibles horarios finales serán más reducidas.

Por otro lado, en el caso más complicado se han restringido las disponibilidades horarias de los profesores que imparten pocas clases a la semana; por ejemplo, si un profesor sólo debe impartir una clase de una hora de duración y esta clase se debe dar dos veces por semana, este profesor sólo estará disponible dos días a la semana. Puesto que esta restricción afecta a 8 de los 23 profesores existentes del problema, se espera que sea más complicado encontrar una solución de cumpla las restricciones *hard*.

La secuencia de los tres niveles de dificultad espera verse reflejada en el valor del coste de cada generación y en el coste del mejor individuo de cada iteración. De esta forma se espera que el coste en los casos sencillos decrezca más rápidamente. Al tener menos posibles celdas horarias en conflicto para cada restricción esta añadirá un menor número de penalizaciones total para el individuo.

Del punto de vista de la variación de los parámetros del algoritmo, se espera que usando una reducida probabilidad de mutación el algoritmo no tenga una evolución muy lenta hacia un horario que sea solución. Esto se debe a que la mutación permite salir de singularidades locales de forma que rompe

la monotonía que ofrece la recombinación. Por otro lado, una probabilidad de mutación elevada añade una componente de aleatoriedad grande. Esto puede repercutir sobre los mejores individuos de cada generación en repetidas ocasiones, con lo que se pueden revertir los cambios que llevan al programa a alcanzar una solución óptima.

IV. Resultados

Las ejecuciones del algoritmo sobre los diferentes *inputs* y con las variaciones del algoritmo arrojan una gran cantidad de datos iteración tras iteración. Esta aparece en diversas gráficas en la Sección V. En ellas se muestra la evolución de los diferentes costes a lo largo de las iteraciones para cada ejecución.

Condensando la información de los resultados se puede centrar la atención en si la ejecución obtiene una solución válida para el problema. Recordar que, en este caso, una solución válida es aquella que no incumple ninguna de las restricciones *hard* marcadas al inicio. Se espera que el mejor individuo de la última generación sea una solución válida.

teacher_hours_easy			
Selección padres	Generaciones	Prob. Mutación	Cumple rest. hard
Roulette	200	0.01	No
Roulette	200	0.15	No
Roulette	200	0.35	No
Roulette	500	0.01	No
Roulette	500	0.15	**Sí**
Roulette	500	0.35	**Sí**
Tournament	200	0.01	No
Tournament	200	0.15	**Sí**
Tournament	200	0.35	**Sí**
Tournament	500	0.01	No
Tournament	500	0.15	**Sí**
Tournament	500	0.35	**Sí**

Cuadro I: Tabla de resultados para *inputs teacher_hours_easy.*

teacher_hours			
Selección padres	Generaciones	Prob. Mutación	Cumple rest. hard
Roulette	200	0.01	No
Roulette	200	0.15	No
Roulette	200	0.35	No
Roulette	500	0.01	No
Roulette	500	0.15	No
Roulette	500	0.35	No
Tournament	200	0.01	No
Tournament	200	0.15	No
Tournament	200	0.35	**Sí**
Tournament	500	0.01	No
Tournament	500	0.15	**Sí**
Tournament	500	0.35	**Sí**

Cuadro II: Tabla de resultados para *inputs teacher_hours.*

De entre los casos que no cumplen todas las restricciones, destaca el quinto caso del Cuadro II. En este caso, el coste del individuo mejor al finalizar la ejecución es de 2060 con el incumplimiento de una sola restricción *hard*, con lo que previsiblemente este caso en generaciones próximas sería resuelto como confirman algunos otros casos.

Por último, en el Cuadro III se muestran los resultados de la ejecución con el conjunto de datos más complicado

teacher_hours_hard			
Selección padres	Generaciones	Prob. Mutación	Cumple rest. hard
Roulette	200	0.01	No
Roulette	200	0.15	No
Roulette	200	0.35	No
Roulette	500	0.01	No
Roulette	500	0.15	No
Roulette	500	0.35	No
Tournament	200	0.01	No
Tournament	200	0.15	**Sí**
Tournament	200	0.35	**Sí**
Tournament	500	0.01	No
Tournament	500	0.15	**Sí**
Tournament	500	0.35	**Sí**

Cuadro III: Tabla de resultados para *inputs teacher_hours_hard.*

que se ha tratado. Sorprendentemente, el algoritmo evolutivo ha sido capaz de encontrar soluciones válidas en 4 de las 12 ejecuciones realizadas. Destacar que en la ejecución de la Figura 85 (algoritmo de la ruleta con mutación 0.35) se muestra que los costes de las *hard* bajan considerablemente, por lo que se puede pensar que aumentando el número de generaciones podría haber llegado a una solución válida.

IV-A. Tendencias observadas

Tras la ejecución del algoritmo para todos los casos planteados, las gráficas generadas permiten observar ciertas tendencias o realizar comparaciones para intuir la influencia de los parámetros modificados.

En primer lugar, pueden compararse los métodos de selección de padres (mirando siempre sobre el mismo ratio de mutación, para mantener mínimas las variaciones). El sistema de ruleta probabilística aparentemente es más neutro ya que se parece más a una selección aleatoria con un peso hacia aquellos elementos de mejor *fitness*. En cambio, la selección por torneo tiende a tomar los mejores individuos de los subconjuntos para cada selección de padres. A priori, la selección de progenitores por torneo parece mejor al combinar el carácter aleatorio en la elección (al tomarse un subconjunto aleatorio de los padres disponibles para elegir al progenitor) y entre esos elegidos aleatoriamente, se toma el de mejor. Se pretende favorecer a los mejores individuos más allá de las probabilidades en la elección.

Dicho detalle de selección creemos que se refleja en el coste de las generaciones a lo largo de las iteraciones del algoritmo. Y así lo confirmamos en la evolución del valor de la función de coste para toda la generación. Por ejemplo, en la evolución del coste de las generaciones en las parejas de Figuras 45a - 63a o 54a - 72a.

En segundo lugar, los diferentes datos de entrada suponen un diferente aspecto en los resultados de las ejecuciones. Para ilustrarlo, se propone la comparación entre los tres niveles de dificultad y comparando las ejecuciones seleccionando padres por torneo, 200 generaciones y una probabilidad de mutación de 0.15, reflejadas en las Figuras 36a, 72a

y 108a. Visualmente se buscan el codo donde el coste de la generación pasa a decrecer a menor ritmo y la iteración aproximada en la que se obtiene un individuo solución con $H1 = 0$ y $H5 = 0$. Se emplean valores aproximados ya que entre ejecuciones pueden variar los valores.

Dificultad datos	Coste del codo	Iteración codo	Iteración solución
Easy	100000	50	375
Medium	300000	75	240
Hard	500000	75	450

Cuadro IV: Comparación de resultados según dificultad de los datos de entrada

De la tabla, comprobamos que a mayor dificultad es mayor el coste en el que situamos el codo, aunque no significativamente alejados en el eje horizontal. También se podría extraer que a menor dificultad, antes se encuentra un individuo que cumple todas las restricciones *hard*. Las ejecuciones mostradas en este caso no lo cumplen estrictamente. El conjunto de datos sencillo encuentra soluciones de bajo coste muy rápidamente, pero no encuentra una válida pasadas las 350 iteraciones. Se atribuye esto a la inicialización y a las acciones con componente aleatoria.

Otro aspecto a destacar es la influencia de la probabilidad de mutación. El resultado esperado para una probabilidad baja de mutación es que las generaciones puedan quedar estancadas en ciertos valores de *fitness*. A pesar de añadir generaciones, el algoritmo no obtiene un resultado cercano a los deseados, con costes por encima de 25000 (solo para las penalizaciones *hard*), como es el caso de los *inputs* de complejidad media (*teacher_hours*).

Las gráficas muestran como las variaciones de los costes son discretos, de acuerdo con su definición. Se aprecia mejor en las gráficas de las propias restricciones, o en los individuos, donde un incumplimiento de una restricción provoca un salto igual al peso asociado a la restricción.

Para valorar en términos generales el comportamiento del algoritmo es mejor observarlo sobre el coste de la población completa para las diferentes combinaciones. Este coste incluye todas las modificaciones que se han realizado en los individuos, por lo que la tendencia de la población muestra qué configuraciones llevan a este problema alcanzar más rápidamente los mínimos óptimos; además, es una comprobación algo más inmune a la posible generación de un individuo muy bueno por razones de aleatoriedad.

Por contra, para valorar si el programa avanza hacia una solución de forma más controlada, sin dar muchos pasos atrás u oscilar en un rango de *fitness* de la generación, es posible verlo reflejado en la evolución del mejor individuo de cada generación. Además, también se visualiza el estancamiento en los tramos dónde no se mejora el individuo de menor coste (tramos horizontales).

En este sentido, las configuraciones con un criterio de selección por torneo tienden a reducir mucho el coste en las primeras iteraciones tanto de la población ya que evitan escoger a los elementos de mayor función de coste. Así se refleja en las Figuras 63b y 48b. Las gráficas análogas para el procedimiento de la ruleta probabilística muestran un comportamiento más oscilatorio en el valor del mejor individuo de cada generación.

IV-B. Horarios resultantes

Finalmente, en este apartado se comentan algunas de las soluciones obtenidas a lo largo de las diferentes ejecuciones.

La representación de los horarios obtenidos está en forma de tabla de Excel, donde las columnas se corresponden con los cursos y las filas con los diferentes *slots* horarios que existen en el problema. Las casillas tienen el color:

- Azul: en caso de que se trate de una hora libre.
- Rosa: en caso de que se incumpla la restricción `H1`.
- Rojo: en caso de que se incumpla la restricción `H5`.

En la Figura 4 se muestra el horario construido para los datos de $teacher_hours_hard$ mediante el algoritmo de selección de padres de la ruleta, con 500 generaciones y con 0.15 de mutación. El individuo al que corresponde el horario tenía los siguientes costes:

- `H1`: 2000. El profesor Carlos C. tiene horas solapadas entre cursos diferentes.
- `H5`: 14000. Diferentes profesores tienen asignados días en los que no están disponibles para impartir clases.
- `S1`: 150.
- `S2`: 470.
- `S3`: 95.
- `S4`: 115.
- `S5`: 160.
- **Coste total:** 16990.

En comparación, la Figura 5 se muestra el horario construido para los datos de $teacher_hours_hard$ manteniendo la configuración anterior pero empleando el algoritmo de selección de padres por torneo. El individuo al que corresponde el horario tenía los siguientes costes:

- `H1`: 0.
- `H5`: 0.
- `S1`: 150.
- `S2`: 400.
- `S3`: 80.
- `S4`: 115.
- `S5`: 125.
- **Coste total:** 870.

La diferencia en la calidad de estos dos resultados es apreciable; en concreto, hay una diferencia de coste de **16120**.

1ESO	2ESO	3ESO	4ESO	1BATX	2BATX
LUNES 8 Matemáticas with Eva M.	LUNES 8 Inglés with Miquel B.	LUNES 8 Religión with Silvia S.	LUNES 8 Matemáticas B with Llorenç S.		LUNES 8 Lengua Castellana y literatura with Carlos C.
		LUNES 9 Educación física with Gloria F.	LUNES 9 Informática with Llorenç S.	LUNES 9 Lengua Catalana y literatura with Esther C.	LUNES 9 Inglés with Miquel B.
LUNES 10 Ciencia de la naturaleza with Andreu F.	LUNES 10 Lengua Catalana y literatura with Esther C.	LUNES 10 Educación plástica y visual with Pere V.	LUNES 10 Tecnología with Xim H.	LUNES 10 Educación física with Gloria F.	LUNES 10 Lengua Castellana y literatura with Carlos C.
LUNES 11 Religión with Silvia S.	LUNES 11 Lengua Castellana y literatura with Carlos C.	LUNES 11 Tecnología with Xim H.	LUNES 11 Matemáticas B with Llorenç S.	LUNES 11 Inglés with Miquel B.	LUNES 11 Física with Ángel F.
LUNES 12 Matemáticas with Eva M.	LUNES 12 Lengua Castellana y literatura with Carlos C.	LUNES 12 Música with María P.	LUNES 12 Educación eticocívica with Pedro P.	LUNES 12 Matemáticas 1 with Cristina O.	LUNES 12 Historia de la filosofía with Carlos S.
	LUNES 13 Francés with Agathe N.	LUNES 13 Lengua Catalana y literatura with Esther C.		LUNES 13 Tecnología industrial with Yolanda M.	LUNES 13 Matemáticas 2 with Eva M.
LUNES 14 Francés with Agathe N.	LUNES 14 Ciencia de la naturaleza with Andreu F.	LUNES 14 Lengua Castellana y literatura with Carlos C.	LUNES 14 Informática with Llorenç S.	LUNES 14 Lengua Catalana y literatura with Esther C.	
		MARTES 8 Educación plástica y visual with Pere V.		MARTES 8 Lengua Castellana y literatura with Carlos C.	MARTES 8 Lengua Catalana y literatura with Esther C.
MARTES 9 Lengua Catalana y literatura with Esther C.	MARTES 9 Educación para la ciudadanía y los derechos humanos with Leonor A.	MARTES 9 Inglés with Miquel B.	MARTES 9 Educación física with Gloria F.	MARTES 9 Filosofía y ciutadania with Maria Teresa L.	MARTES 9 Historia de España with Alexandre S.
MARTES 10 Ciencia de la naturaleza with Andreu F.	MARTES 10 Matemáticas with Eva M.	MARTES 10 Lengua Catalana y literatura with Esther C.	MARTES 10 Lengua Castellana y literatura with Carlos C.		MARTES 10 Tecnología industrial with Yolanda M.
MARTES 11 Ciencias sociales, geografía e historia with Alexandre S.	MARTES 11 Educación física with Gloria F.	MARTES 11 Taller de ciéncias experimentales with Yolanda M.	MARTES 11 Inglés with Miquel B.	MARTES 11 Lengua Castellana y literatura with Carlos C.	MARTES 11 Lengua Catalana y literatura with Esther C.
MARTES 12 Lengua Castellana y literatura with Carlos C.	MARTES 12 Religión with Silvia S.	MARTES 12 Ciencia de la naturaleza with Andreu F.	MARTES 12 Tecnología with Xim H.		MARTES 12 Inglés with Miquel B.
MARTES 13 Lengua Catalana y literatura with Esther C.	MARTES 13 Matemáticas with Eva M.	MARTES 13 Lengua Castellana y literatura with Carlos C.	MARTES 13 Matemáticas B with Llorenç S.		MARTES 13 Lengua Castellana y literatura with Carlos C.
MARTES 14 Educación física with Gloria F.	MARTES 14 Ciencias sociales, geografía e historia with Alexandre S.	MARTES 14 Matemáticas with Eva M.	MARTES 14 Lengua Catalana y literatura with Esther C.	MARTES 14 Inglés with Miquel B.	MARTES 14 Historia de la filosofía with Carlos S.
MIÉRCOLES 8 Educación plástica y visual with Pere V.	MIÉRCOLES 8 Lengua Castellana y literatura with Carlos C.	MIÉRCOLES 8 Matemáticas with Eva M.		MIÉRCOLES 8 Tecnología de la información y la comunicación with Carmen R.	MIÉRCOLES 8 Inglés with Miquel B.
MIÉRCOLES 9 Ciencia de la naturaleza with Andreu F.		MIÉRCOLES 9 Tecnología with Xim H.	MIÉRCOLES 9 Educación física with Gloria F.	MIÉRCOLES 9 Matemáticas 1 with Cristina O.	MIÉRCOLES 9 Lengua Catalana y literatura with Esther C.
	MIÉRCOLES 10 Lengua Catalana y literatura with Esther C.	MIÉRCOLES 10 Matemáticas with Eva M.	MIÉRCOLES 10 Física y química with Ángel F.	MIÉRCOLES 10 Tecnología de la información y la comunicación with Carmen R.	
MIÉRCOLES 11 Matemáticas with Eva M.	MIÉRCOLES 11 Educación física with Gloria F.	MIÉRCOLES 11 Ciencias sociales, geografía e historia with Alexandre S.	MIÉRCOLES 11 Física y química with Ángel F.	MIÉRCOLES 11 Lengua Castellana y literatura with Carlos C.	MIÉRCOLES 11 Lengua Catalana y literatura with Esther C.
MIÉRCOLES 12 Inglés with Miquel B.	MIÉRCOLES 12 Matemáticas with Eva M.	MIÉRCOLES 12 Ciencias sociales, geografía e historia with Alexandre S.	MIÉRCOLES 12 Tecnología with Xim H.	MIÉRCOLES 12 Educación física with Gloria F.	MIÉRCOLES 12 Tecnología industrial with Yolanda M.
MIÉRCOLES 13 Educación física with Gloria F.	MIÉRCOLES 13 Inglés with Miquel B.	MIÉRCOLES 13 Música with María P.	MIÉRCOLES 13 Lengua Catalana y literatura with Esther C.	MIÉRCOLES 13 Ciéncias para el mundo contemporáneo with Cristina B.	MIÉRCOLES 13 Historia de España with Alexandre S.
MIÉRCOLES 14 Lengua Catalana y literatura with Esther C.	MIÉRCOLES 14 Tecnología with Xim H.	MIÉRCOLES 14 Lengua Castellana y literatura with Carlos C.	MIÉRCOLES 14 Inglés with Miquel B.	MIÉRCOLES 14 Tecnología industrial with Yolanda M.	MIÉRCOLES 14 Lengua Castellana y literatura with Carlos C.
JUEVES 8 Lengua Castellana y literatura with Carlos C.	JUEVES 8 Educación para la ciudadanía y los derechos humanos with Leonor A.	JUEVES 8 Educación física with Gloria F.	JUEVES 8 Lengua Catalana y literatura with Esther C.	JUEVES 8 Inglés with Miquel B.	JUEVES 8 Tecnología industrial with Yolanda M.
JUEVES 9 Música with María P.	JUEVES 9 Matemáticas with Eva M.	JUEVES 9 Inglés with Miquel B.	JUEVES 9 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 9 Matemáticas 1 with Cristina O.	JUEVES 9 Tecnología industrial with Yolanda M.
JUEVES 10 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 10 Ciencia de la naturaleza with Andreu F.	JUEVES 10 Inglés with Miquel B.	JUEVES 10 Religión with Silvia S.	JUEVES 10 Lengua Catalana y literatura with Esther C.	JUEVES 10 Matemáticas 2 with Eva M.
JUEVES 11 Lengua Catalana y literatura with Esther C.	JUEVES 11 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 11 Ciencia de la naturaleza with Andreu F.		JUEVES 11 Lengua Castellana y literatura with Carlos C.	
JUEVES 12 Música with María P.	JUEVES 12 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 12 Matemáticas with Eva M.	JUEVES 12 Lengua Castellana y literatura with Carlos C.		JUEVES 12 Inglés with Miquel B.
	JUEVES 13 Lengua Castellana y literatura with Carlos C.	JUEVES 13 Tecnología with Xim H.	JUEVES 13 Inglés with Miquel B.	JUEVES 13 Tecnología industrial with Yolanda M.	JUEVES 13 Matemáticas 2 with Eva M.
JUEVES 14 Matemáticas with Eva M.	JUEVES 14 Francés with Agathe N.	JUEVES 14 Lengua Catalana y literatura with Esther C.	JUEVES 14 Lengua Castellana y literatura with Carlos C.	JUEVES 14 Física y química with Ángel F.	
VIERNES 8 Ciencias sociales, geografía e historia with Alexandre S.	VIERNES 8 Inglés with Miquel B.	VIERNES 8 Lengua Catalana y literatura with Esther C.	VIERNES 8 Lengua Castellana y literatura with Carlos C.	VIERNES 8 Matemáticas 1 with Cristina O.	VIERNES 8 Electrotécnia with Ruth B.
VIERNES 9 Educación plástica y visual with Pere V.	VIERNES 9 Lengua Catalana y literatura with Esther C.	VIERNES 9 Lengua Castellana y literatura with Carlos C.	VIERNES 9 Educación eticocívica with Pedro P.		VIERNES 9 Física with Ángel F.
VIERNES 10 Inglés with Miquel B.	VIERNES 10 Tecnología with Xim H.			VIERNES 10 Física y química with Ángel F.	VIERNES 10 Electrotécnia with Ruth B.
VIERNES 11 Lengua Castellana y literatura with Carlos C.	VIERNES 11 Lengua Catalana y literatura with Esther C.		VIERNES 11 Ciencias sociales, geografía e historia with Alexandre S.	VIERNES 11 Tecnología de la información y la comunicación with Carmen R.	VIERNES 11 Matemáticas 2 with Eva M.
VIERNES 12 Lengua Castellana y literatura with Carlos C.	VIERNES 12 Ciencia de la naturaleza with Andreu F.		VIERNES 12 Lengua Catalana y literatura with Esther C.	VIERNES 12 Ciéncias para el mundo contemporáneo with Cristina B.	VIERNES 12 Historia de la filosofía with Carlos S.
VIERNES 13 Francés with Agathe N.			VIERNES 13 Ciencias sociales, geografía e historia with Alexandre S.	VIERNES 13 Física y química with Ángel F.	
VIERNES 14 Inglés with Miquel B.			VIERNES 14 Matemáticas B with Llorenç S.	VIERNES 14 Lengua Catalana y literatura with Esther C.	VIERNES 14 Historia de España with Alexandre S.

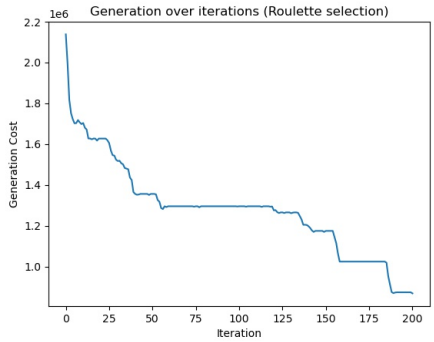
Figura 4: **Datos:** $teacher_hours_hard$ **Caso: Roulette, 500 gen. prob. mut. 0.15**. Horario resultante de la ejecución.

1ESO	2ESO	3ESO	4ESO	1BATX	2BATX
LUNES 8 Lengua Catalana y literatura with Esther C.	LUNES 8 Inglés with Miquel B.	LUNES 8 Tecnología with Xim H.	LUNES 8 Lengua Castellana y literatura with Carlos C.	LUNES 8 Matemáticas 1 with Cristina O.	LUNES 8 Matemáticas 2 with Eva M.
LUNES 9 Francés with Agathe N.	LUNES 9 Lengua Castellana y literatura with Carlos C.	LUNES 9 Ciencia de la naturaleza with Andreu F.	LUNES 9 Matemáticas B with Llorenç S.		LUNES 9 Lengua Catalana y literatura with Esther C.
LUNES 10 Ciencia de la naturaleza with Andreu F.	LUNES 10 Lengua Castellana y literatura with Carlos C.	LUNES 10 Lengua Catalana y literatura with Esther C.	LUNES 10 Educación eticocívica with Pedro P.	LUNES 10 Inglés with Miquel B.	LUNES 10 Tecnología industrial with Yolanda M.
LUNES 11 Lengua Castellana y literatura with Carlos C.		LUNES 11 Música with María P.	LUNES 11 Lengua Catalana y literatura with Esther C.	LUNES 11 Física y química with Ángel F.	LUNES 11 Matemáticas 2 with Eva M.
LUNES 12 Lengua Catalana y literatura with Esther C.	LUNES 12 Lengua Castellana y literatura with Carlos C.	LUNES 12 Matemáticas with Eva M.	LUNES 12 Educación física with Gloria F.		LUNES 12 Historia de la filosofía with Carlos S.
LUNES 13 Inglés with Miquel B.	LUNES 13 Francés with Agathe N.	LUNES 13 Educación plástica y visual with Pere V.		LUNES 13 Lengua Castellana y literatura with Carlos C.	
LUNES 14 Religión with Silvia S.	LUNES 14 Ciencia de la naturaleza with Andreu F.	LUNES 14 Lengua Castellana y literatura with Carlos C.	LUNES 14 Física y química with Ángel F.	LUNES 14 Tecnología industrial with Yolanda M.	LUNES 14 Lengua Catalana y literatura with Esther C.
	MARTES 8 Matemáticas with Eva M.	MARTES 8 Taller de ciéncias experimentales with Yolanda M.	MARTES 8 Tecnología with Xim H.	MARTES 8 Lengua Catalana y literatura with Esther C.	MARTES 8 Historia de España with Alexandre S.
MARTES 9 Ciencias sociales, geografía e historia with Alexandre S.	MARTES 9 Ciencia de la naturaleza with Andreu F.		MARTES 9 Informática with Llorenç S.	MARTES 9 Tecnología industrial with Yolanda M.	MARTES 9 Inglés with Miquel B.
MARTES 10 Educación física with Gloria F.		MARTES 10 Lengua Catalana y literatura with Esther C.	MARTES 10 Matemáticas B with Llorenç S.		MARTES 10 Lengua Castellana y literatura with Carlos C.
MARTES 11 Matemáticas with Eva M.	MARTES 11 Educación para la ciudadanía y los derechos humanos with Leonor A.	MARTES 11 Lengua Castellana y literatura with Carlos C.	MARTES 11 Lengua Catalana y literatura with Esther C.	MARTES 11 Educación física with Gloria F.	
MARTES 12 Educación plástica y visual with Pere V.	MARTES 12 Tecnología with Xim H.	MARTES 12 Lengua Catalana y literatura with Esther C.	MARTES 12 Ciencias sociales, geografía e historia with Alexandre S.	MARTES 12 Inglés with Miquel B.	MARTES 12 Lengua Castellana y literatura with Carlos C.
MARTES 13 Ciencia de la naturaleza with Andreu F.	MARTES 13 Ciencias sociales, geografía e historia with Alexandre S.	MARTES 13 Lengua Castellana y literatura with Carlos C.	MARTES 13 Lengua Catalana y literatura with Esther C.	MARTES 13 Física y química with Ángel F.	
MARTES 14 Lengua Catalana y literatura with Esther C.		MARTES 14 Educación física with Gloria F.	MARTES 14 Inglés with Miquel B.	MARTES 14 Filosofía y ciutadania with Maria Teresa L.	
MIÉRCOLES 8 Música with María P.	MIÉRCOLES 8 Lengua Catalana y literatura with Esther C.	MIÉRCOLES 8 Ciencias sociales, geografía e historia with Alexandre S.	MIÉRCOLES 8 Lengua Castellana y literatura with Carlos C.	MIÉRCOLES 8 Educación física with Gloria F.	MIÉRCOLES 8 Historia de la filosofía with Carlos S.
MIÉRCOLES 9 Inglés with Miquel B.	MIÉRCOLES 9 Matemáticas with Eva M.	MIÉRCOLES 9 Lengua Castellana y literatura with Carlos C.		MIÉRCOLES 9 Tecnología industrial with Yolanda M.	MIÉRCOLES 9 Historia de España with Alexandre S.
	MIÉRCOLES 10 Francés with Agathe N.	MIÉRCOLES 10 Inglés with Miquel B.	MIÉRCOLES 10 Lengua Castellana y literatura with Carlos C.	MIÉRCOLES 10 Tecnología de la información y la comunicación with Carmen R.	MIÉRCOLES 10 Tecnología industrial with Yolanda M.
MIÉRCOLES 11 Educación plástica y visual with Pere V.	MIÉRCOLES 11 Educación física with Gloria F.	MIÉRCOLES 11 Matemáticas with Eva M.		MIÉRCOLES 11 Lengua Catalana y literatura with Esther C.	MIÉRCOLES 11 Inglés with Miquel B.
MIÉRCOLES 12 Ciencias sociales, geografía e historia with Alexandre S.	MIÉRCOLES 12 Educación física with Gloria F.	MIÉRCOLES 12 Tecnología with Xim H.	MIÉRCOLES 12 Inglés with Miquel B.	MIÉRCOLES 12 Filosofía y ciutadania with Maria Teresa L.	MIÉRCOLES 12 Lengua Catalana y literatura with Esther C.
MIÉRCOLES 13 Matemáticas with Eva M.	MIÉRCOLES 13 Inglés with Miquel B.	MIÉRCOLES 13 Lengua Catalana y literatura with Esther C.	MIÉRCOLES 13 Física y química with Ángel F.	MIÉRCOLES 13 Ciéncias para el mundo contemporáneo with Cristina B.	MIÉRCOLES 13 Lengua Castellana y literatura with Carlos C.
	MIÉRCOLES 14 Lengua Catalana y literatura with Esther C.			MIÉRCOLES 14 Lengua Castellana y literatura with Carlos C.	MIÉRCOLES 14 Física with Ángel F.
JUEVES 8 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 8 Tecnología with Xim H.	JUEVES 8 Matemáticas with Eva M.	JUEVES 8 Inglés with Miquel B.	JUEVES 8 Matemáticas 1 with Cristina O.	JUEVES 8 Lengua Castellana y literatura with Carlos C.
JUEVES 9 Matemáticas with Eva M.		JUEVES 9 Ciencia de la naturaleza with Andreu F.	JUEVES 9 Tecnología with Xim H.	JUEVES 9 Inglés with Miquel B.	JUEVES 9 Historia de España with Alexandre S.
JUEVES 10 Lengua Castellana y literatura with Carlos C.	JUEVES 10 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 10 Inglés with Miquel B.	JUEVES 10 Religión with Silvia S.	JUEVES 10 Física y química with Ángel F.	JUEVES 10 Matemáticas 2 with Eva M.
JUEVES 11 Lengua Castellana y literatura with Carlos C.	JUEVES 11 Matemáticas with Eva M.	JUEVES 11 Educación física with Gloria F.	JUEVES 11 Ciencias sociales, geografía e historia with Alexandre S.	JUEVES 11 Matemáticas 1 with Cristina O.	JUEVES 11 Inglés with Miquel B.
JUEVES 12 Francés with Agathe N.	JUEVES 12 Educación para la ciudadanía y los derechos humanos with Leonor A.	JUEVES 12 Música with María P.	JUEVES 12 Matemáticas B with Llorenç S.	JUEVES 12 Tecnología de la información y la comunicación with Carmen R.	JUEVES 12 Electrotécnia with Ruth B.
JUEVES 13 Ciencia de la naturaleza with Andreu F.	JUEVES 13 Matemáticas with Eva M.	JUEVES 13 Inglés with Miquel B.	JUEVES 13 Educación física with Gloria F.		JUEVES 13 Tecnología industrial with Yolanda M.
JUEVES 14 Educación física with Gloria F.				JUEVES 14 Lengua Castellana y literatura with Carlos C.	JUEVES 14 Historia de la filosofía with Carlos S.
VIERNES 8 Música with María P.	VIERNES 8 Lengua Castellana y literatura with Carlos C.	VIERNES 8 Religión with Silvia S.	VIERNES 8 Ciencias sociales, geografía e historia with Alexandre S.	VIERNES 8 Lengua Catalana y literatura with Esther C.	VIERNES 8 Tecnología industrial with Yolanda M.
VIERNES 9 Matemáticas with Eva M.	VIERNES 9 Ciencias sociales, geografía e historia with Alexandre S.	VIERNES 9 Educación plástica y visual with Pere V.	VIERNES 9 Matemáticas B with Llorenç S.	VIERNES 9 Ciéncias para el mundo contemporáneo with Cristina B.	VIERNES 9 Electrotécnia with Ruth B.
VIERNES 10 Inglés with Miquel B.	VIERNES 10 Religión with Silvia S.	VIERNES 10 Ciencias sociales, geografía e historia with Alexandre S.	VIERNES 10 Lengua Castellana y literatura with Carlos C.		
	VIERNES 11 Lengua Catalana y literatura with Esther C.	VIERNES 11 Tecnología with Xim H.	VIERNES 11 Informática with Llorenç S.	VIERNES 11 Tecnología de la información y la comunicación with Carmen R.	VIERNES 11 Física with Ángel F.
VIERNES 12 Lengua Castellana y literatura with Carlos C.	VIERNES 12 Inglés with Miquel B.		VIERNES 12 Tecnología with Xim H.	VIERNES 12 Matemáticas 1 with Cristina O.	VIERNES 12 Matemáticas 2 with Eva M.
	VIERNES 13 Ciencia de la naturaleza with Andreu F.	VIERNES 13 Matemáticas with Eva M.	VIERNES 13 Lengua Catalana y literatura with Esther C.	VIERNES 13 Lengua Castellana y literatura with Carlos C.	VIERNES 13 Inglés with Miquel B.
VIERNES 14 Lengua Catalana y literatura with Esther C.	VIERNES 14 Lengua Catalana y literatura with Esther C.		VIERNES 14 Educación eticocívica with Pedro P.	VIERNES 14 Lengua Catalana y literatura with Esther C.	VIERNES 14 Lengua Catalana y literatura with Esther C.

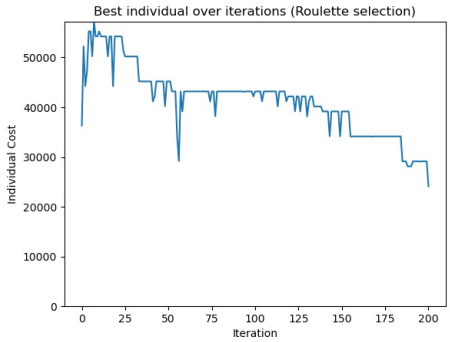
Figura 5: **Datos:** $teacher_hours_hard$ **Caso: Tournament, 500 gen. prob. mut. 0.15**. Horario resultante de la ejecución.

V. AnexoResultados: Gráficas de evolución de las diferentes ejecuciones

V-A. *Resultados de teacher_hours_easy*

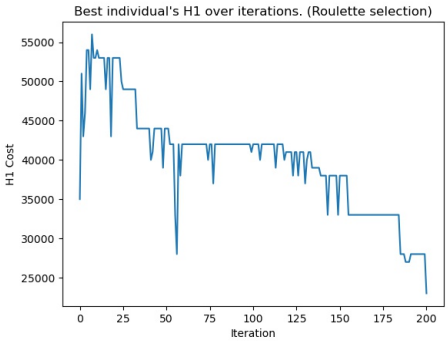


(a) Coste de las sucesivas generaciones.

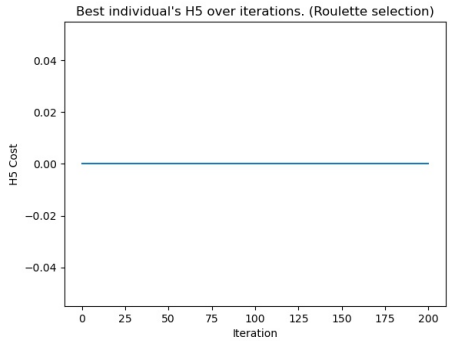


(b) Coste del mejor individuo.

Figura 6: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Evolución de los costes de las generaciones y del mejor individuo de cada generación.

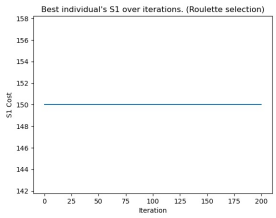


(a) Evolución `H1`.

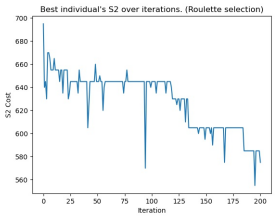


(b) Evolución `H5`.

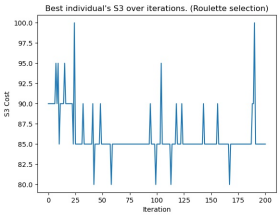
Figura 7: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Costes de las restricciones *hard*.



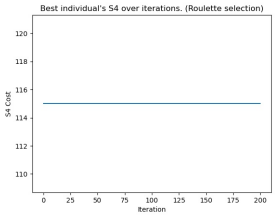
(a) Evolución S1.



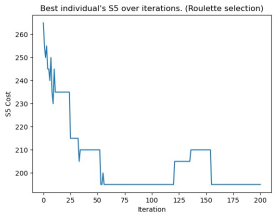
(b) Evolución S2.



(c) Evolución S3.

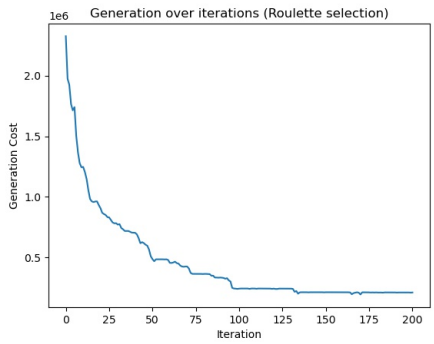


(d) Evolución S4.

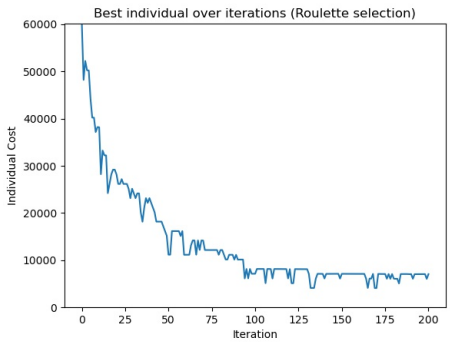


(e) Evolución S5.

Figura 8: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Evolución de las restricciones *soft* a lo largo de las generaciones.

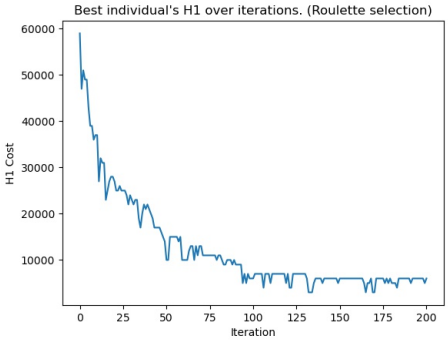


(a) Coste de las sucesivas generaciones.

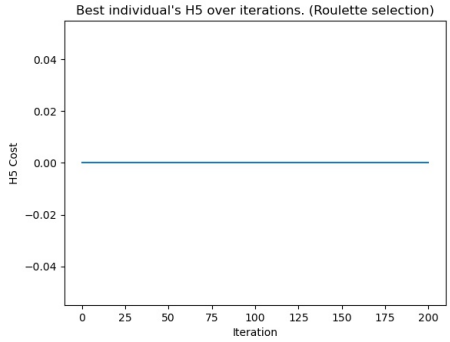


(b) Coste del mejor individuo.

Figura 9: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

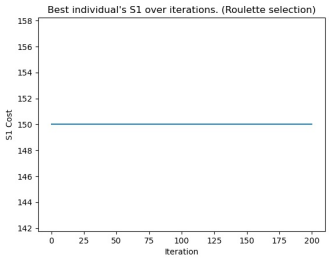


(a) Evolución `H1`.

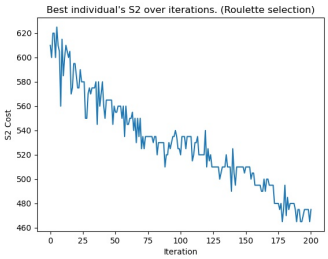


(b) Evolución `H5`.

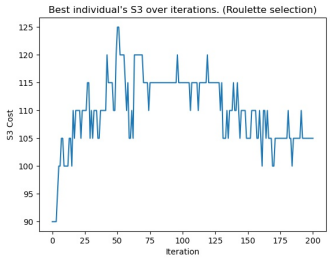
Figura 10: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



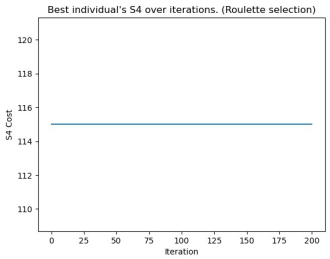
(a) Evolución S1.



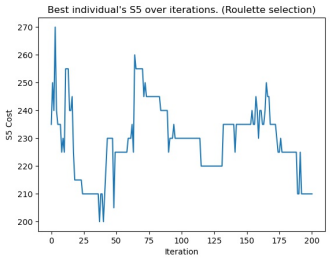
(b) Evolución S2.



(c) Evolución S3.

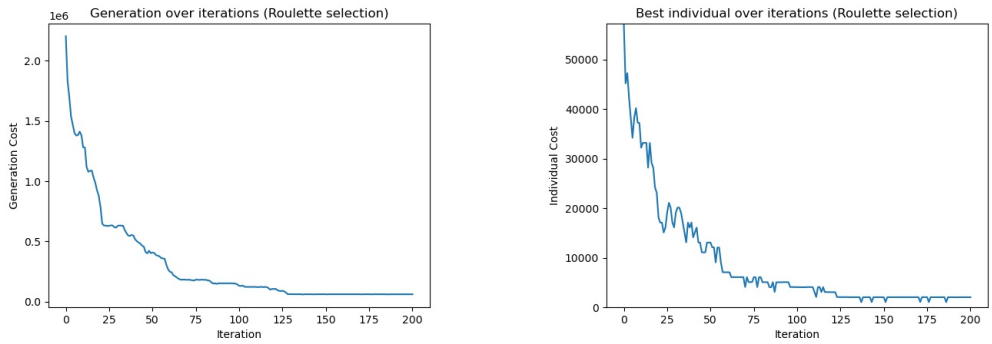


(d) Evolución S4.



(e) Evolución S5.

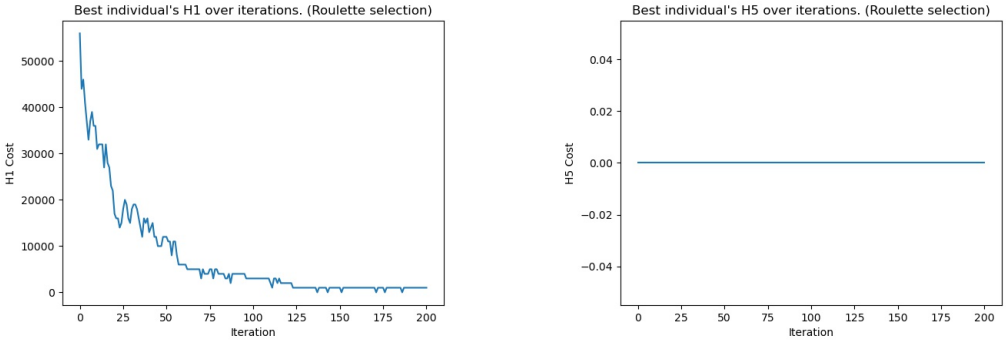
Figura 11: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.



(a) Coste de las sucesivas generaciones

(b) Coste del mejor individuo.

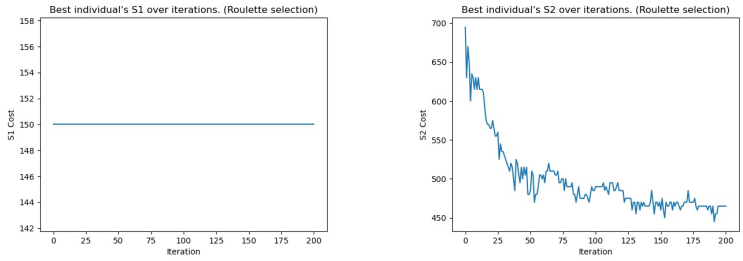
Figura 12: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.



(a) Evolución `H1`.

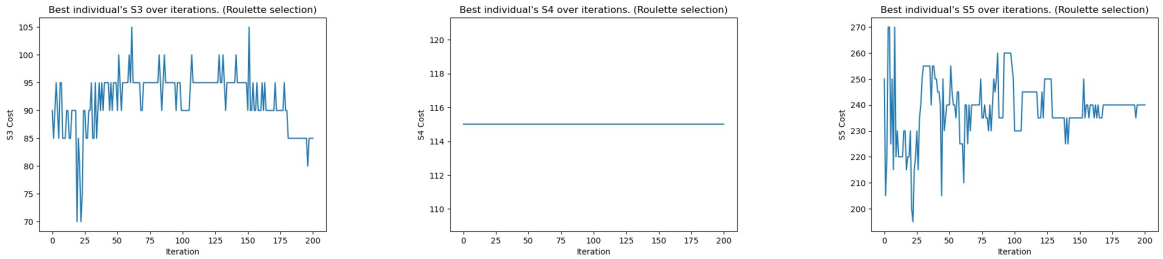
(b) Evolución `H5`.

Figura 13: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



(a) Evolución S1.

(b) Evolución S2.

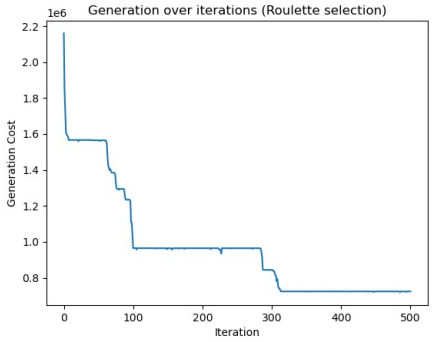


(c) Evolución S3.

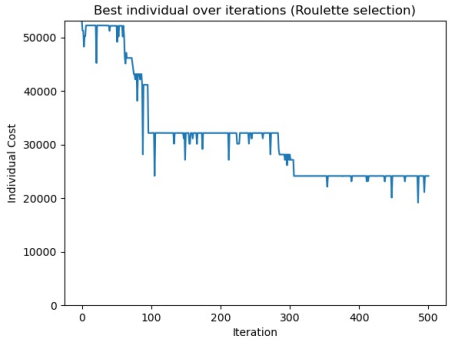
(d) Evolución S4.

(e) Evolución S5.

Figura 14: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

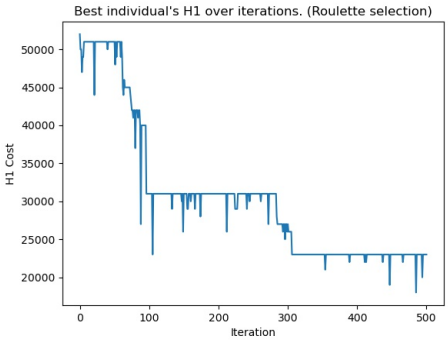


(a) Coste de las sucesivas generaciones

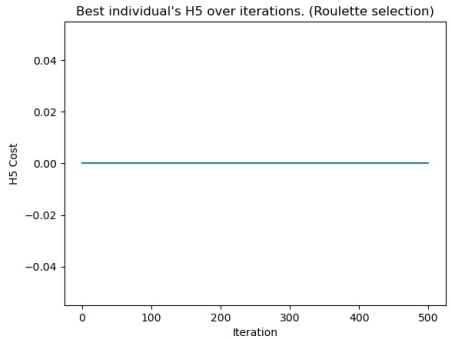


(b) Coste del mejor individuo

Figura 15: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

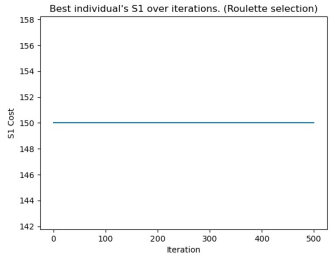


(a) Evolución `H1`.

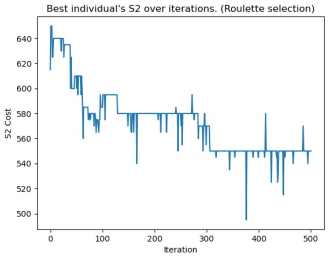


(b) Evolución `H5`.

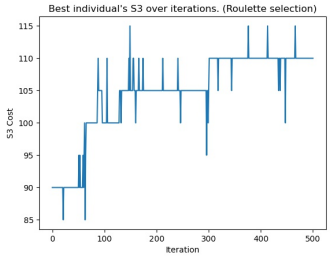
Figura 16: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



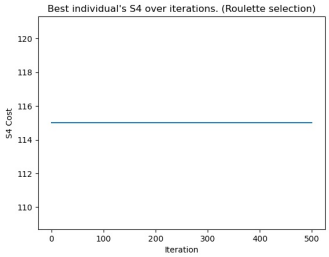
(a) Evolución S1.



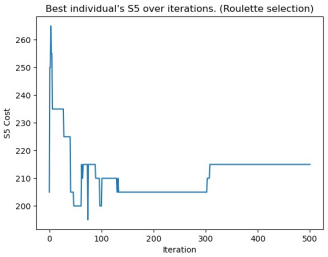
(b) Evolución S2.



(c) Evolución S3.

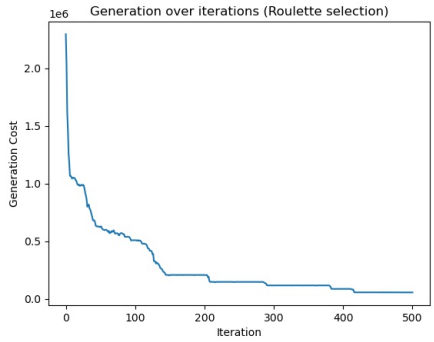


(d) Evolución S4.

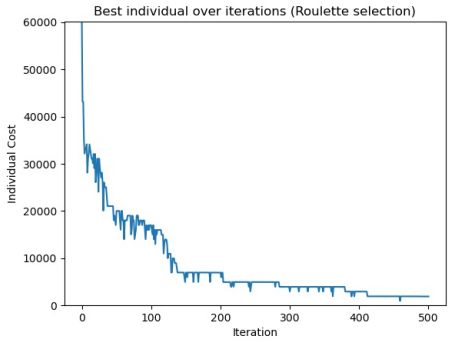


(e) Evolución S5.

Figura 17: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

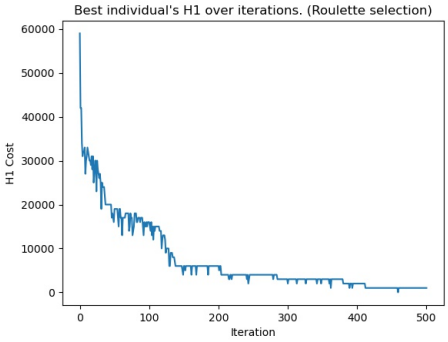


(a) Coste de las sucesivas generaciones.

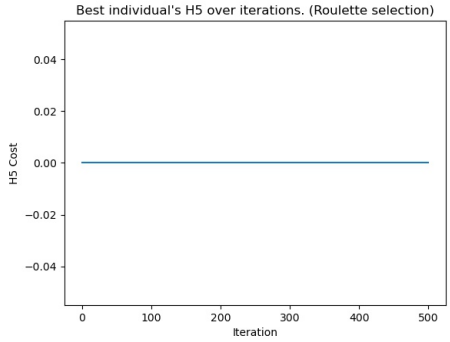


(b) Coste del mejor individuo.

Figura 18: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

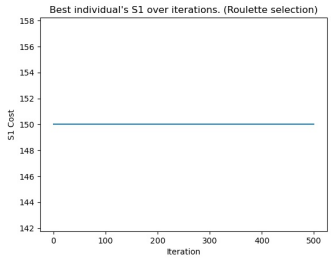


(a) Evolución `H1`.

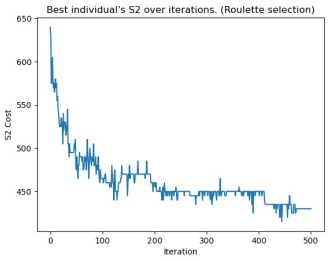


(b) Evolución `H5`.

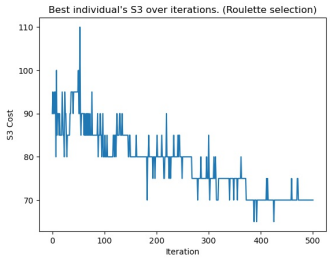
Figura 19: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



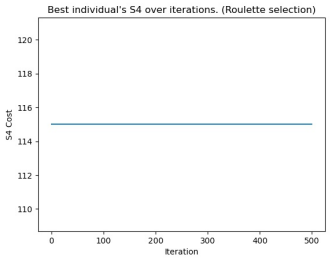
(a) Evolución S1.



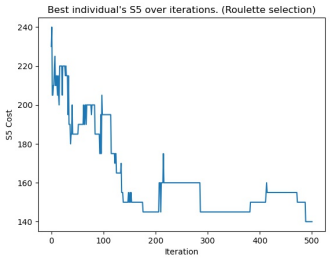
(b) Evolución S2.



(c) Evolución S3.

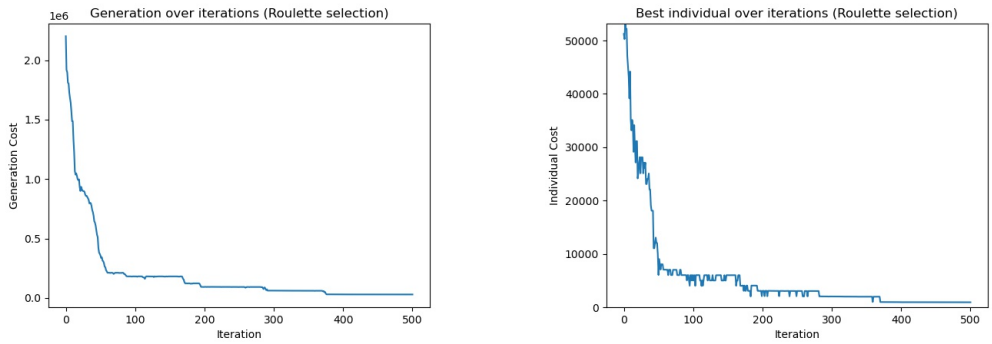


(d) Evolución S4.



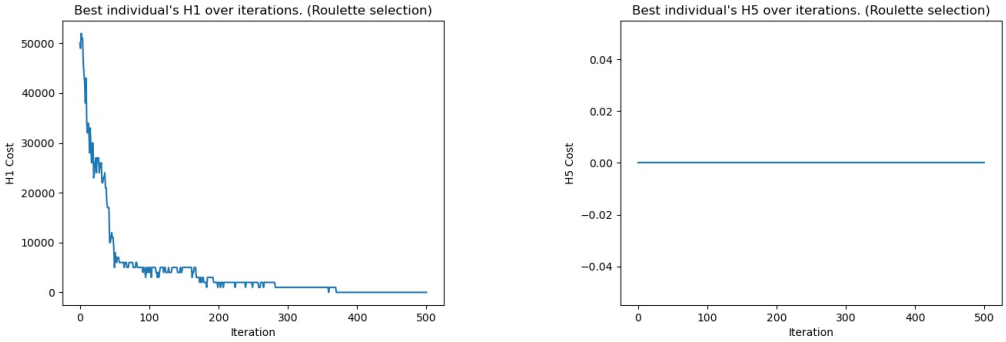
(e) Evolución S5.

Figura 20: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.



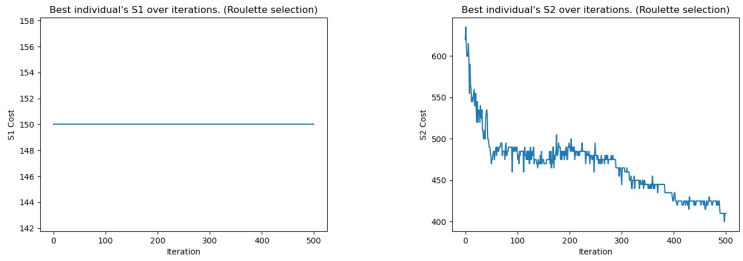
(a) Coste de las sucesivas generaciones (b) Coste del mejor individuo

Figura 21: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

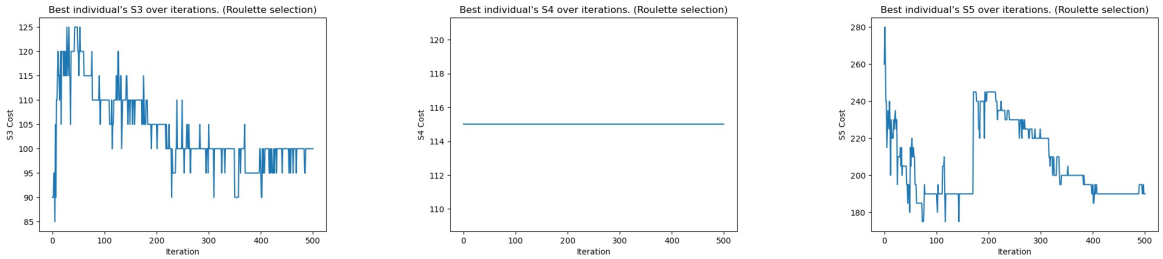


(a) Evolución `H1`. (b) Evolución `H5`.

Figura 22: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.

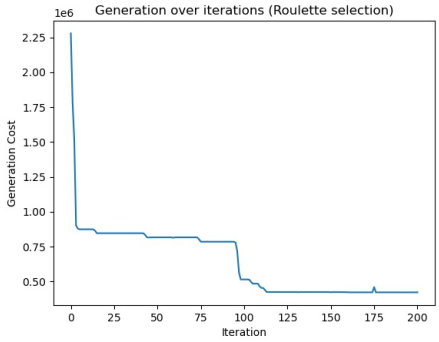


(a) Evolución S1. (b) Evolución S2.

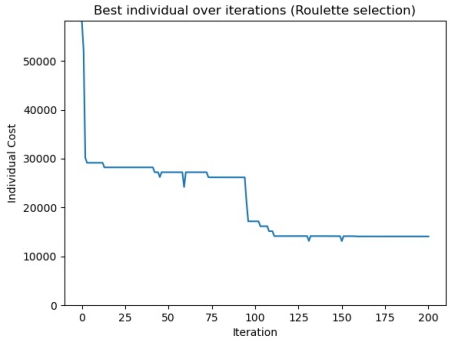


(c) Evolución S3. (d) Evolución S4. (e) Evolución S5.

Figura 23: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

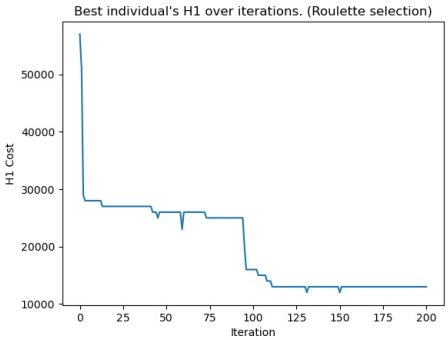


(a) Coste de las sucesivas generaciones

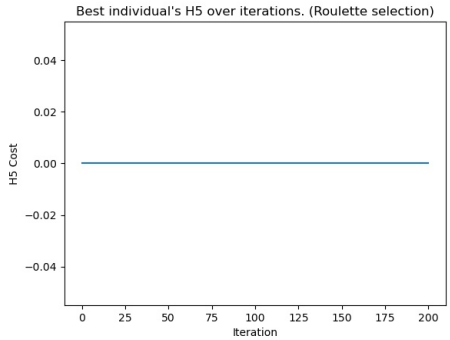


(b) Coste del mejor individuo

Figura 24: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

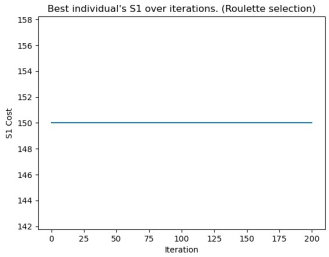


(a) Evolución `H1`.

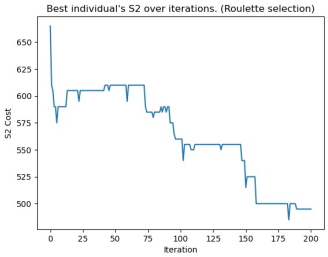


(b) Evolución `H5`.

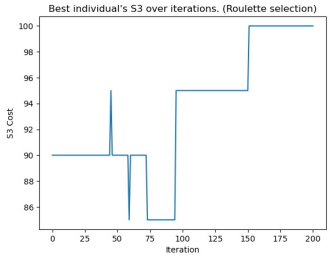
Figura 25: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



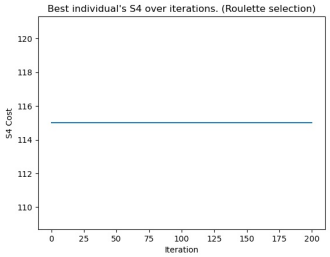
(a) Evolución S1.



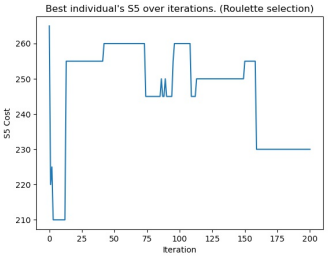
(b) Evolución S2.



(c) Evolución S3.

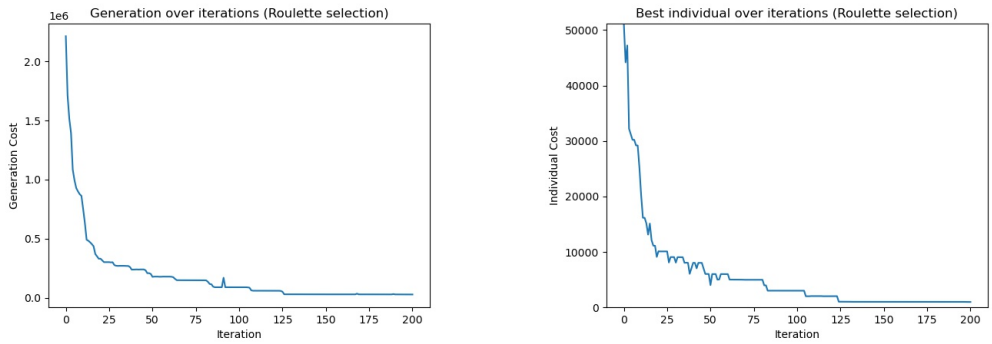


(d) Evolución S4.



(e) Evolución S5.

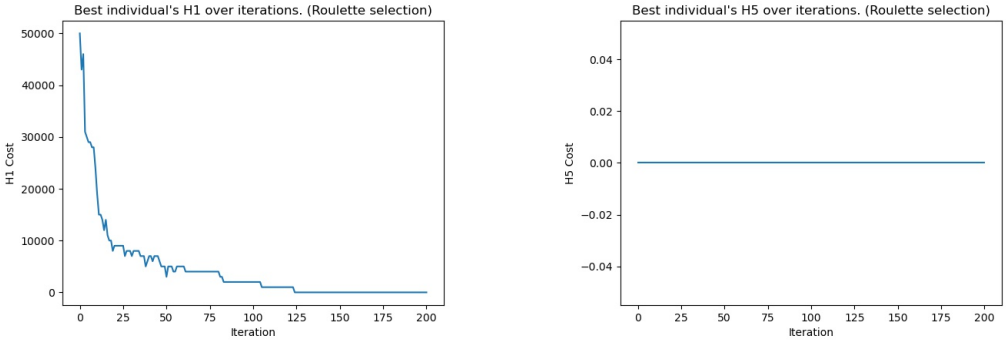
Figura 26: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.



(a) Coste de las sucesivas generaciones

(b) Coste del mejor individuo

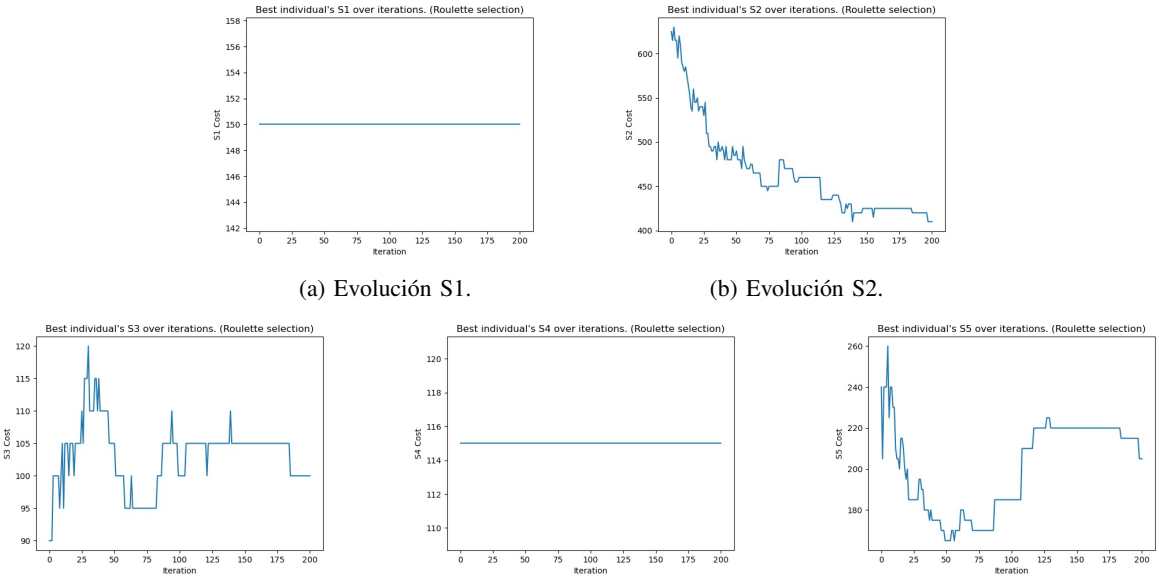
Figura 27: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.



(a) Evolución `H1`.

(b) Evolución `H5`.

Figura 28: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



(a) Evolución S1.

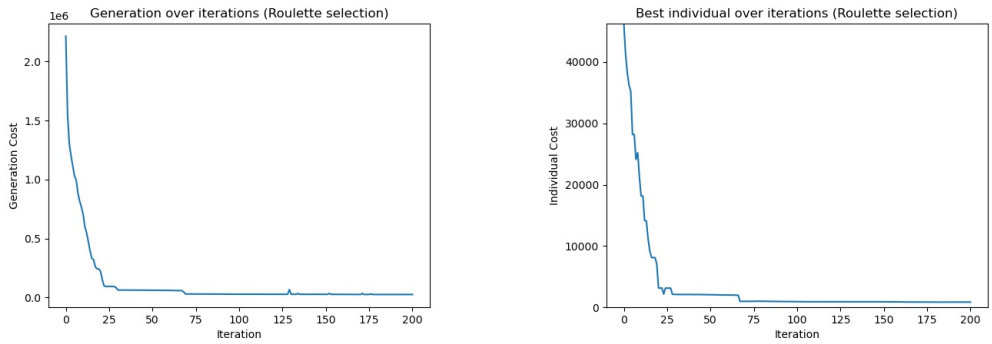
(b) Evolución S2.

(c) Evolución S3.

(d) Evolución S4.

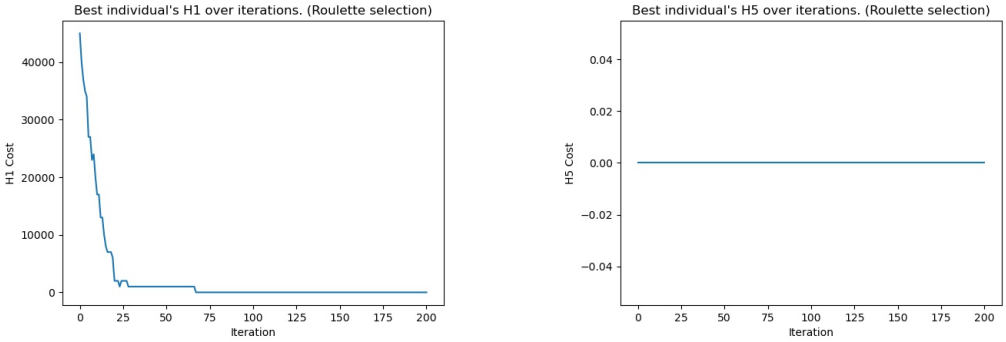
(e) Evolución S5.

Figura 29: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.



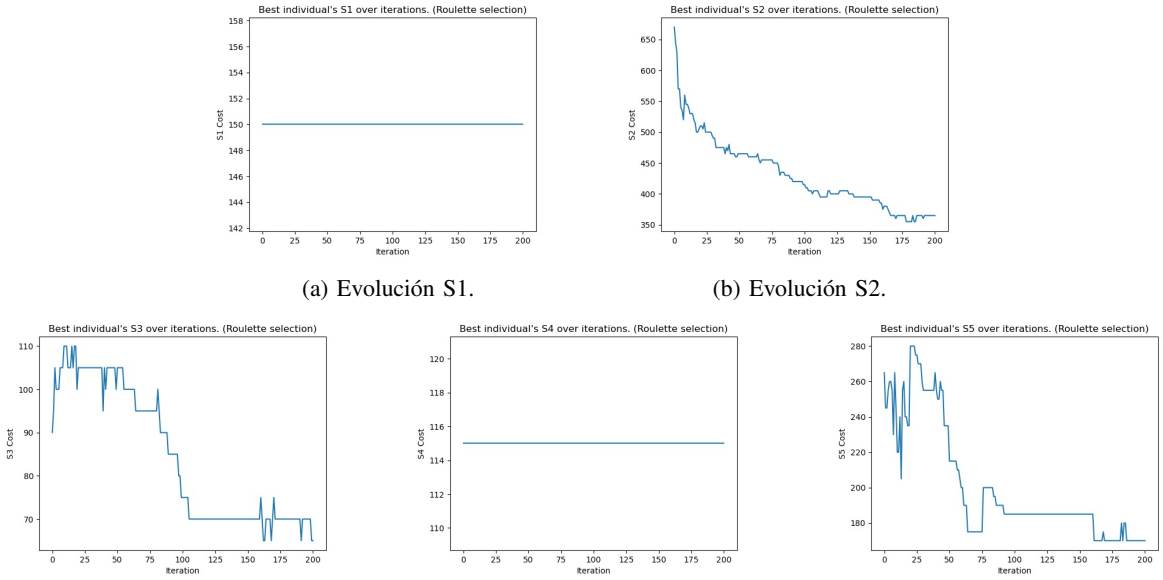
(a) Coste de las sucesivas generaciones (b) Coste del mejor individuo

Figura 30: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.



(a) Evolución `H1`. (b) Evolución `H5`.

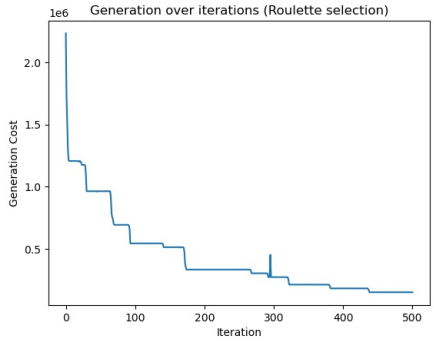
Figura 31: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



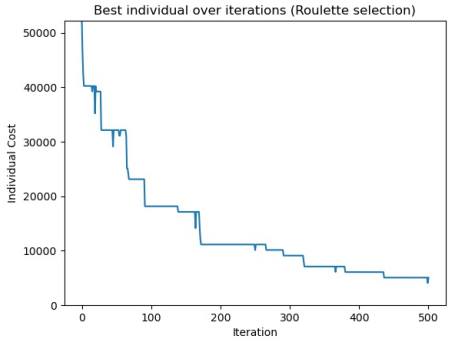
(a) Evolución S1. (b) Evolución S2.

(c) Evolución S3. (d) Evolución S4. (e) Evolución S5.

Figura 32: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

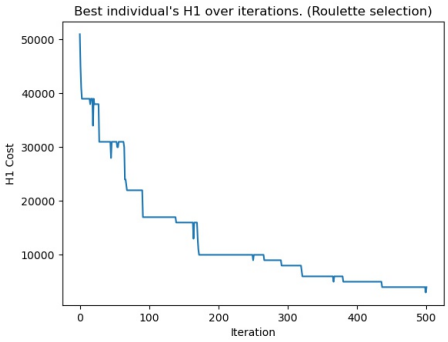


(a) Coste de las sucesivas generaciones.

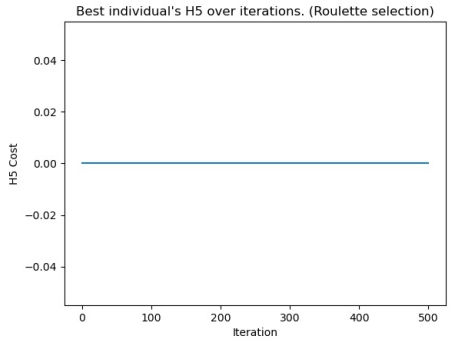


(b) Coste del mejor individuo.

Figura 33: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

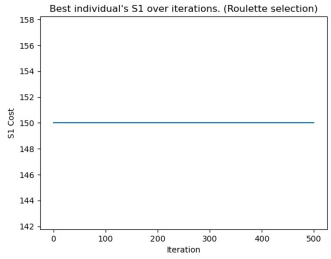


(a) Evolución `H1`.

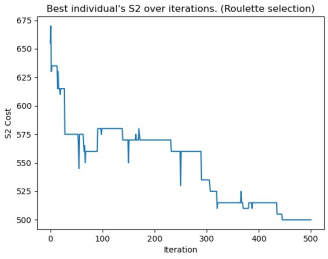


(b) Evolución `H5`.

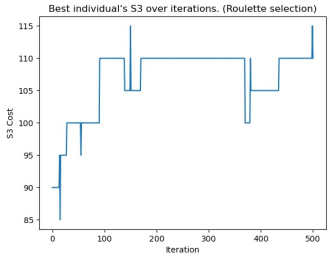
Figura 34: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



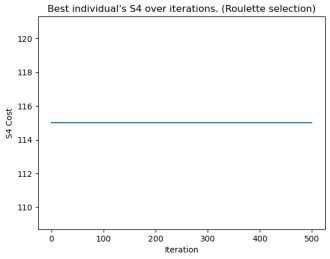
(a) Evolución S1.



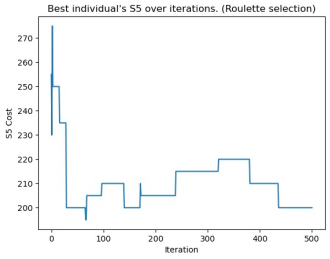
(b) Evolución S2.



(c) Evolución S3.

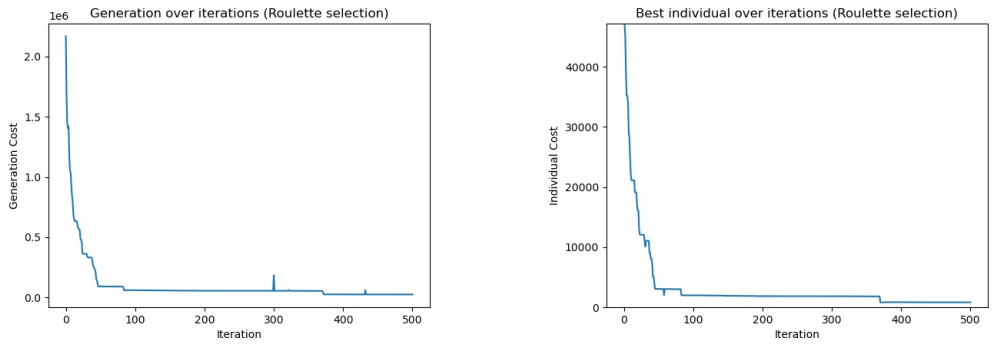


(d) Evolución S4.



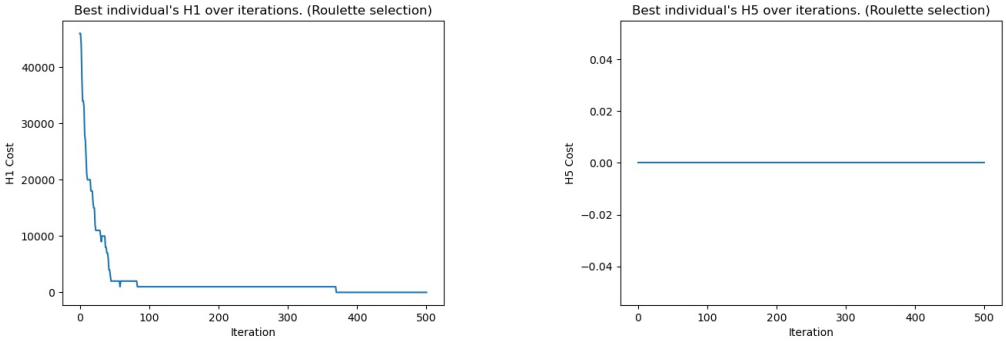
(e) Evolución S5.

Figura 35: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.



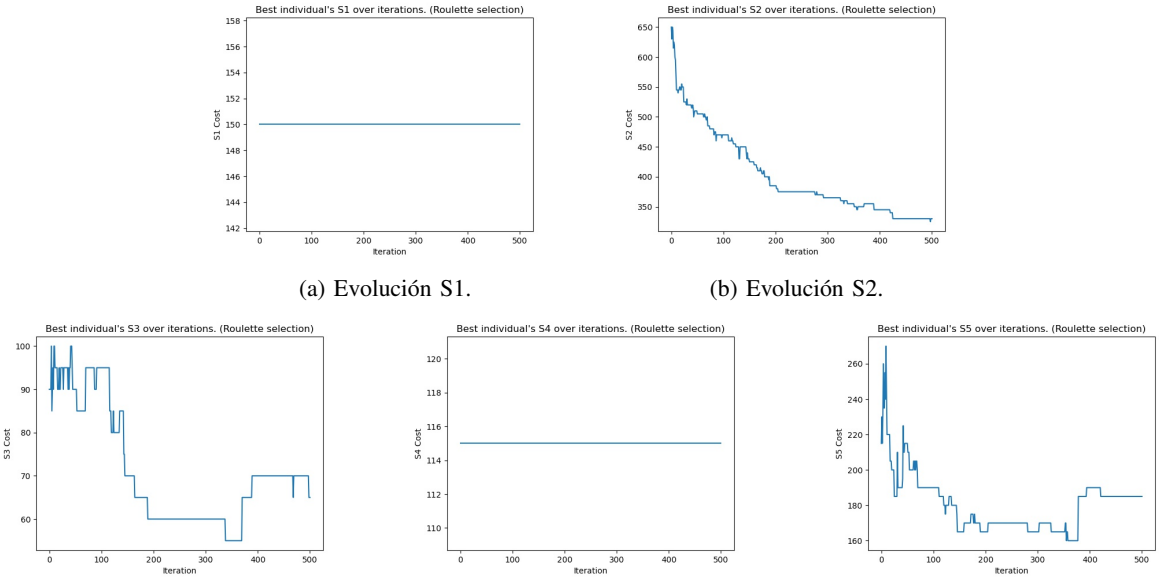
(a) Coste de las sucesivas generaciones (b) Coste del mejor individuo

Figura 36: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.



(a) Evolución `H1`. (b) Evolución `H5`.

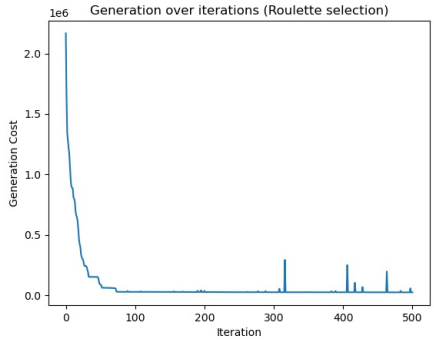
Figura 37: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



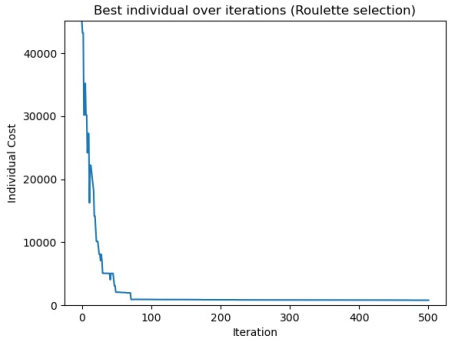
(a) Evolución S1. (b) Evolución S2.

(c) Evolución S3. (d) Evolución S4. (e) Evolución S5.

Figura 38: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

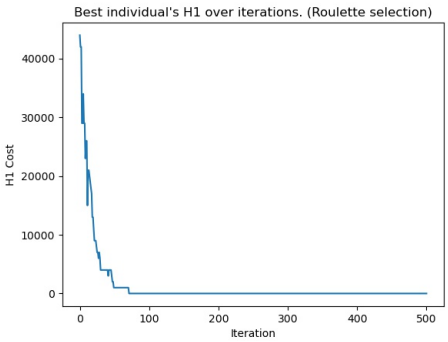


(a) Coste de las sucesivas generaciones

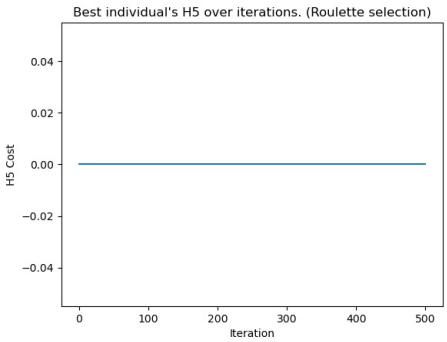


(b) Coste del mejor individuo

Figura 39: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

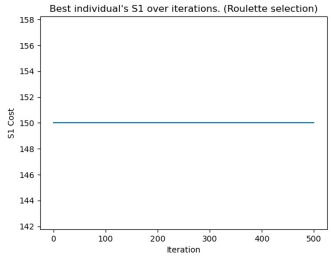


(a) Evolución `H1`.

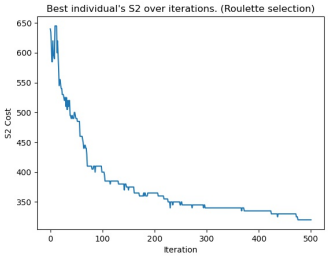


(b) Evolución `H5`.

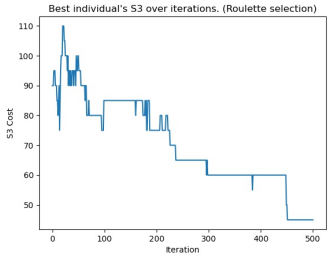
Figura 40: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



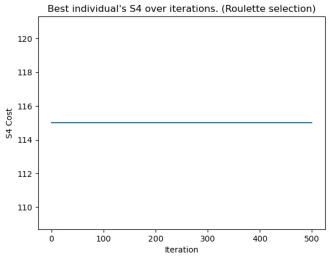
(a) Evolución S1.



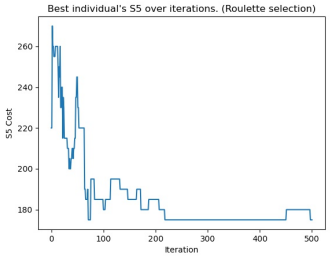
(b) Evolución S2.



(c) Evolución S3.



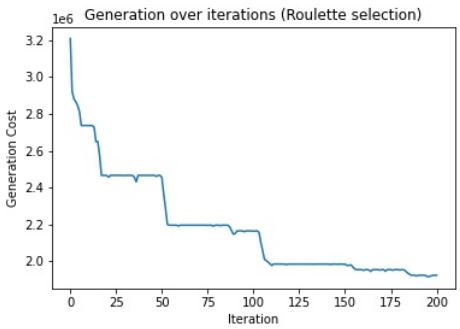
(d) Evolución S4.



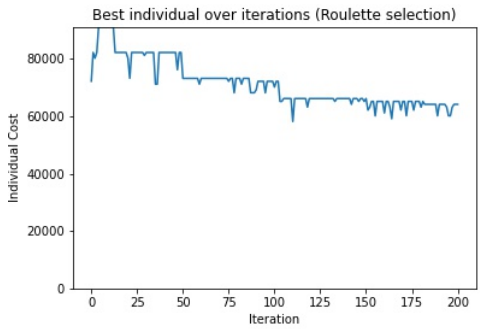
(e) Evolución S5.

Figura 41: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

V-B. Resultados de teacher_hours

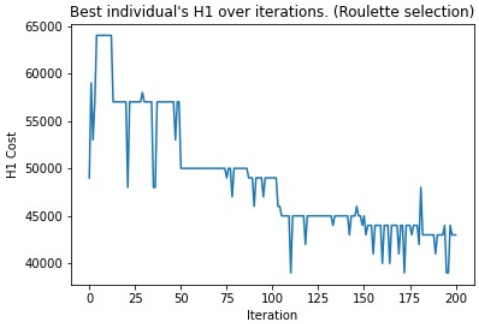


(a) Coste de las sucesivas generaciones.

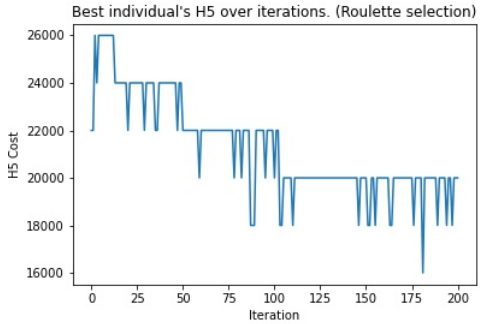


(b) Coste del mejor individuo.

Figura 42: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Evolución de los costes de las generaciones y del mejor individuo de cada generación.

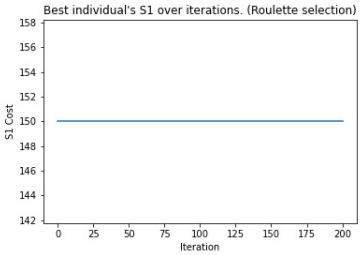


(a) Evolución `H1`.

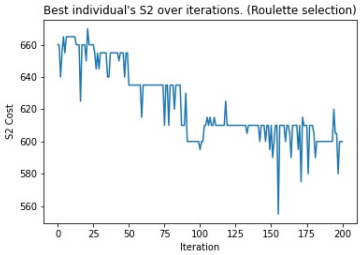


(b) Evolución `H5`.

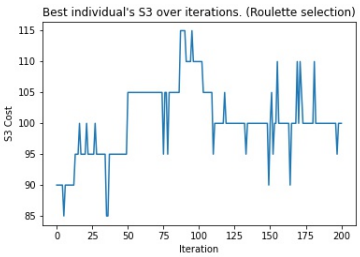
Figura 43: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Costes de las restricciones *hard*.



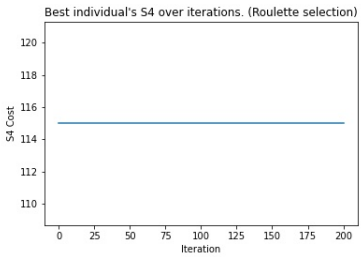
(a) Evolución S1.



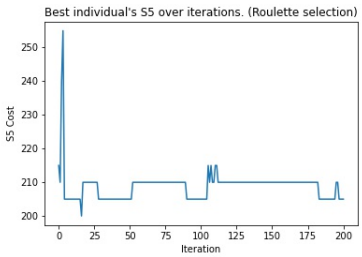
(b) Evolución S2.



(c) Evolución S3.

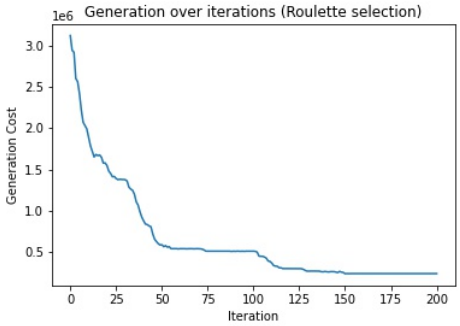


(d) Evolución S4.

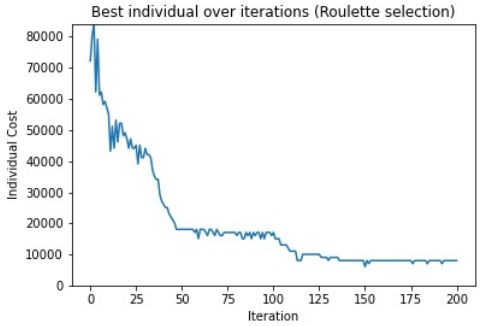


(e) Evolución S5.

Figura 44: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Evolución de las restricciones *soft* a lo largo de las generaciones.

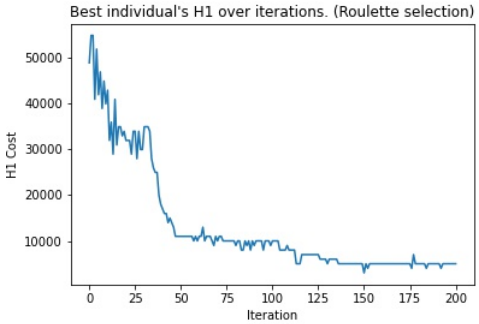


(a) Coste de las sucesivas generaciones.

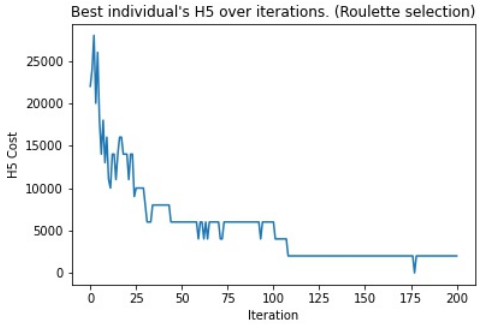


(b) Coste del mejor individuo.

Figura 45: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

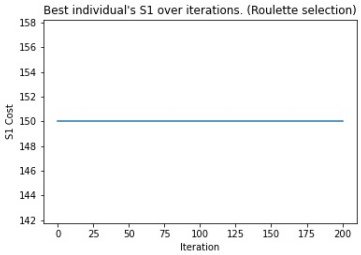


(a) Evolución `H1`.

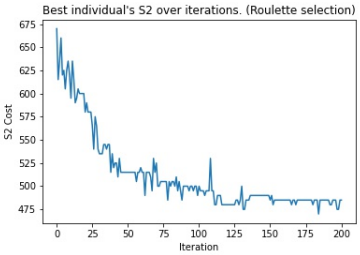


(b) Evolución `H5`.

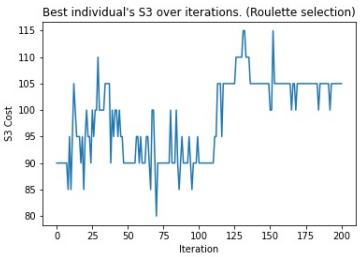
Figura 46: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



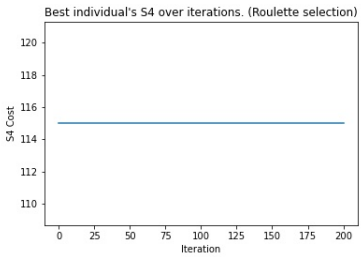
(a) Evolución S1.



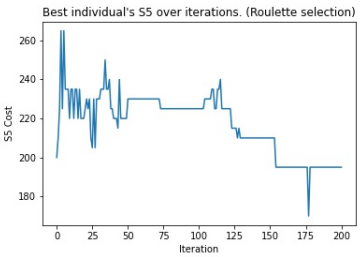
(b) Evolución S2.



(c) Evolución S3.



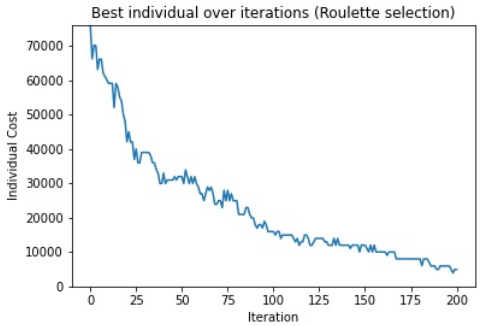
(d) Evolución S4.



(e) Evolución S5.

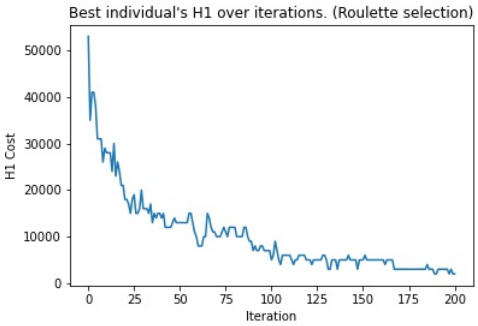
Figura 47: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

(a) Coste de las sucesivas generaciones

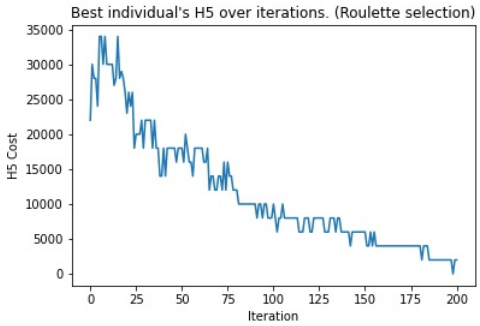


(b) Coste del mejor individuo.

Figura 48: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

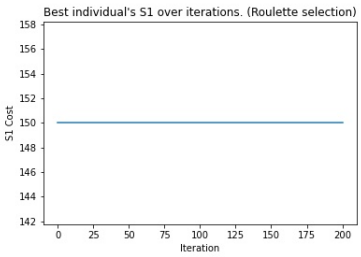


(a) Evolución `H1`.

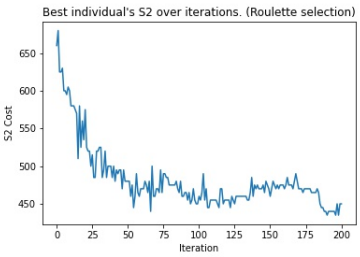


(b) Evolución `H5`.

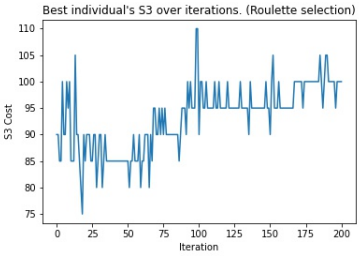
Figura 49: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



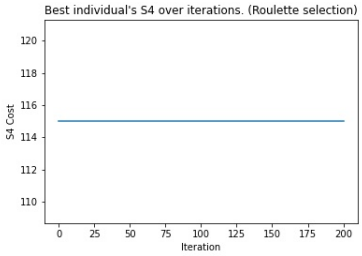
(a) Evolución S1.



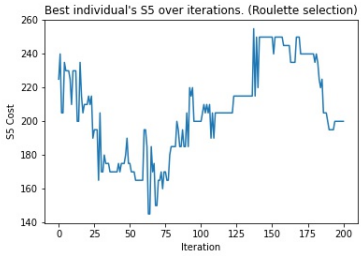
(b) Evolución S2.



(c) Evolución S3.

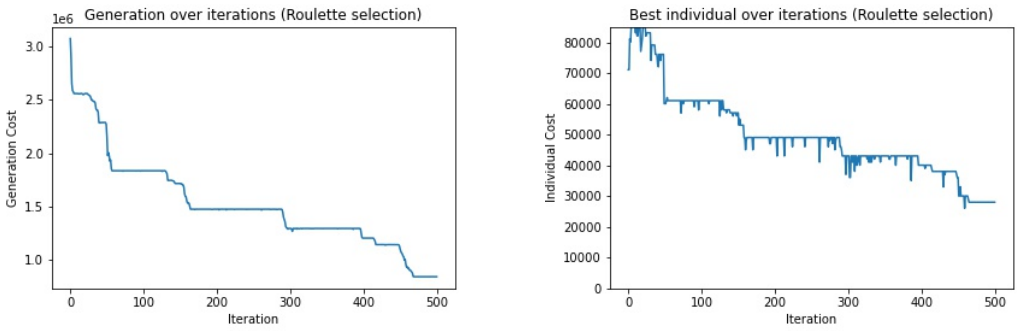


(d) Evolución S4.



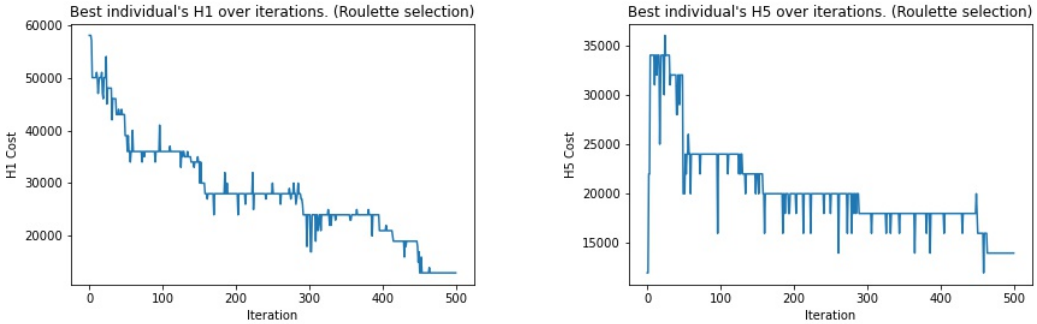
(e) Evolución S5.

Figura 50: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.



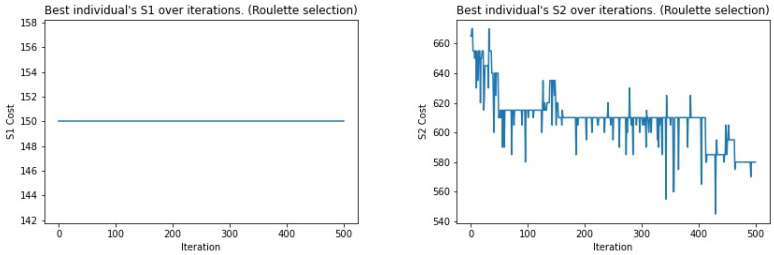
(a) Coste de las sucesivas generaciones (b) Coste del mejor individuo

Figura 51: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

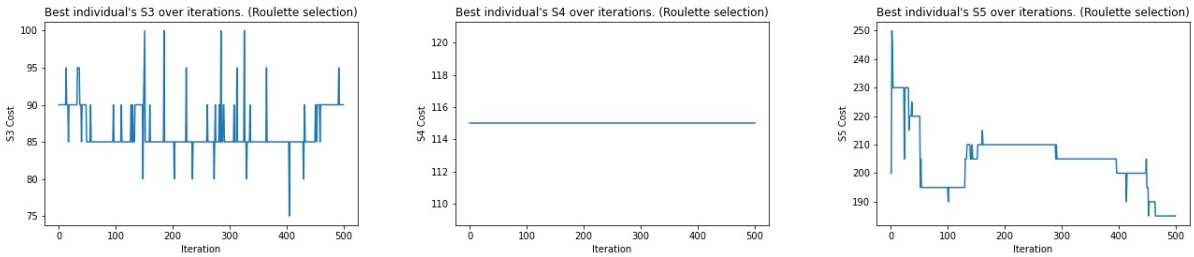


(a) Evolución `H1`. (b) Evolución `H5`.

Figura 52: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.

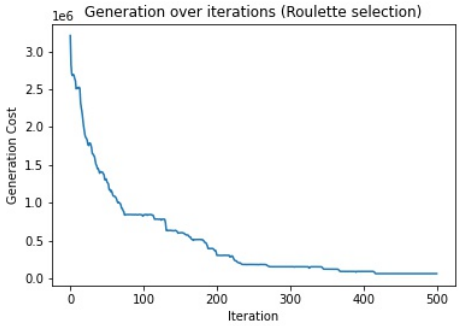


(a) Evolución S1. (b) Evolución S2.

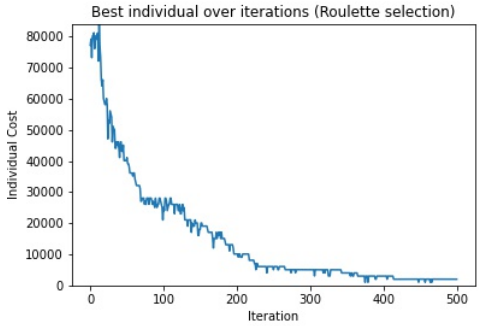


(c) Evolución S3. (d) Evolución S4. (e) Evolución S5.

Figura 53: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

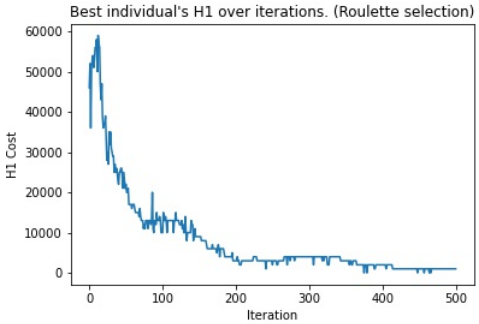


(a) Coste de las sucesivas generaciones.

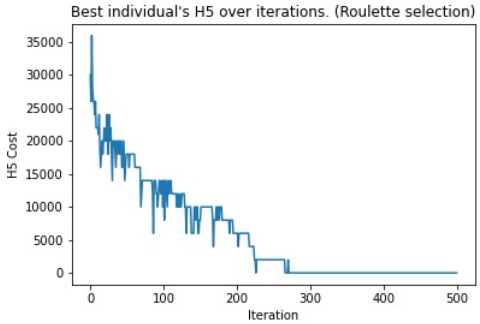


(b) Coste del mejor individuo.

Figura 54: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

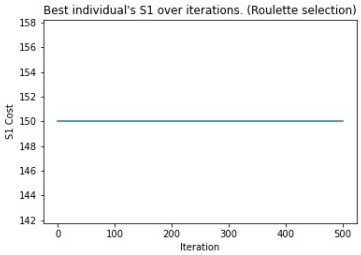


(a) Evolución `H1`.

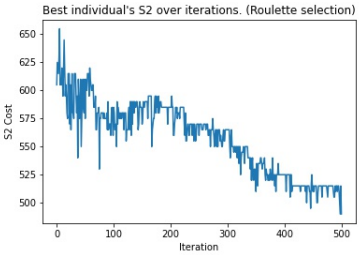


(b) Evolución `H5`.

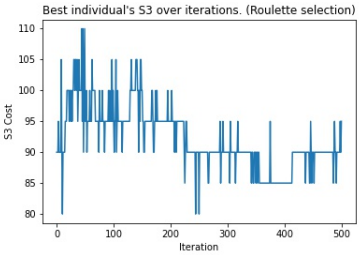
Figura 55: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



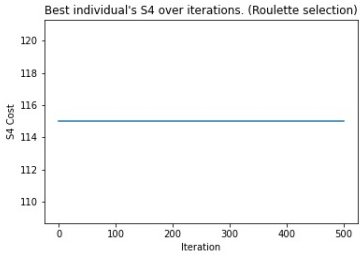
(a) Evolución S1.



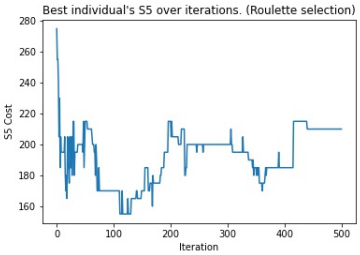
(b) Evolución S2.



(c) Evolución S3.

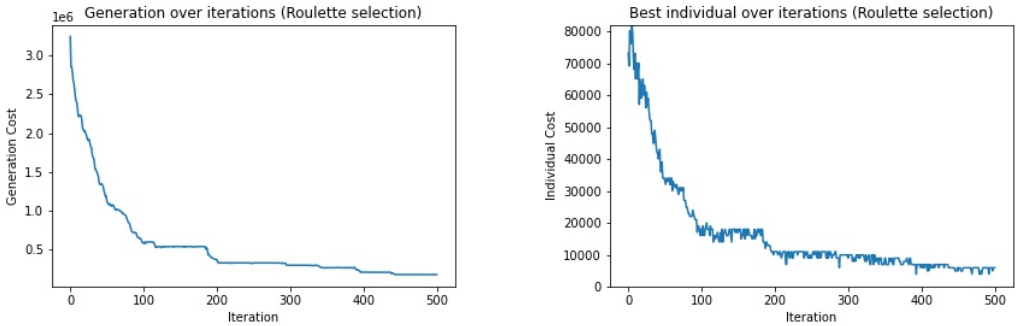


(d) Evolución S4.



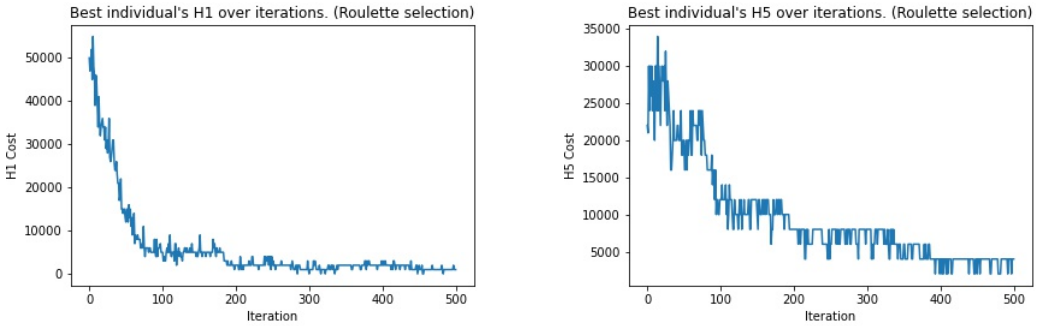
(e) Evolución S5.

Figura 56: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.



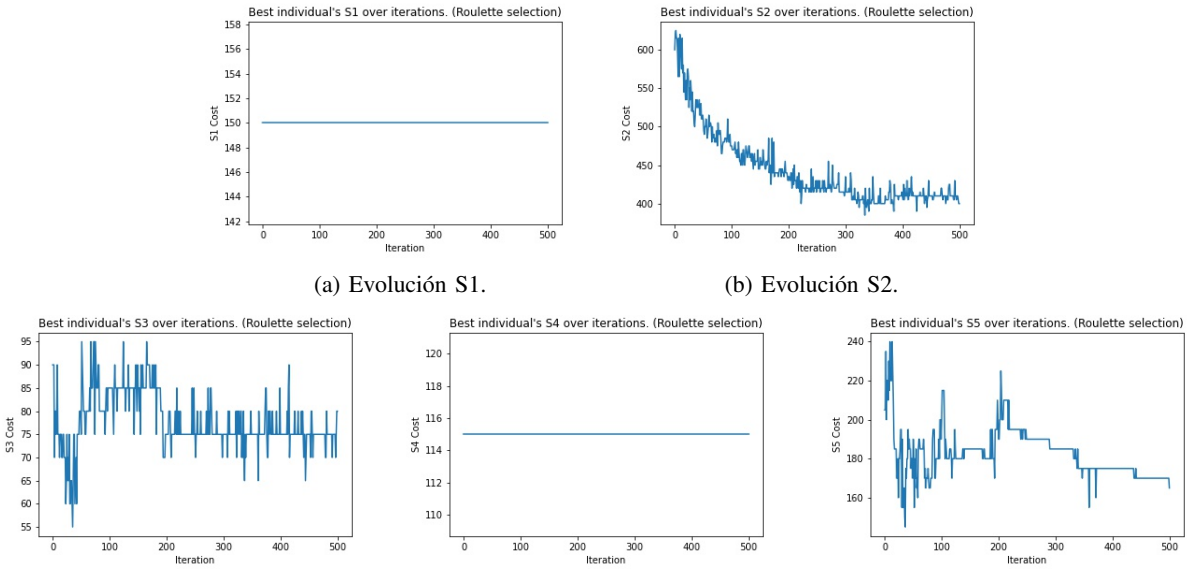
(a) Coste de las sucesivas generaciones (b) Coste del mejor individuo

Figura 57: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.



(a) Evolución `H1`. (b) Evolución `H5`.

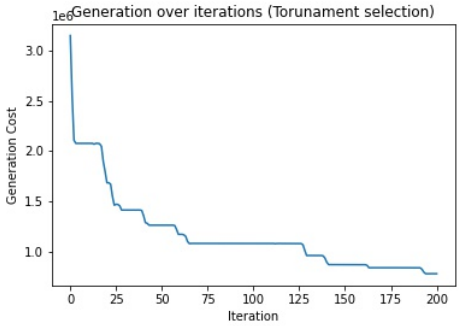
Figura 58: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



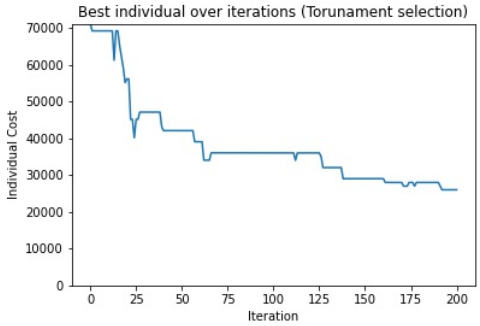
(a) Evolución S1. (b) Evolución S2.

(c) Evolución S3. (d) Evolución S4. (e) Evolución S5.

Figura 59: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

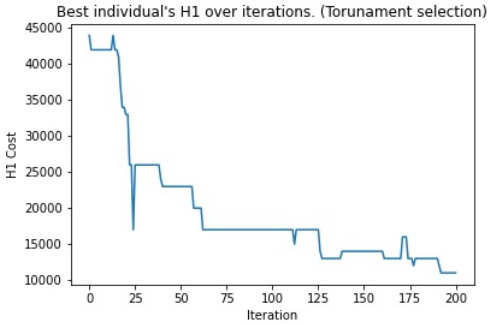


(a) Coste de las sucesivas generaciones

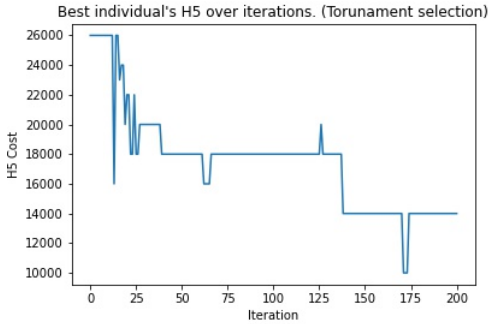


(b) Coste del mejor individuo

Figura 60: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

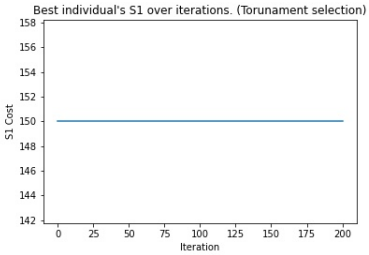


(a) Evolución `H1`.

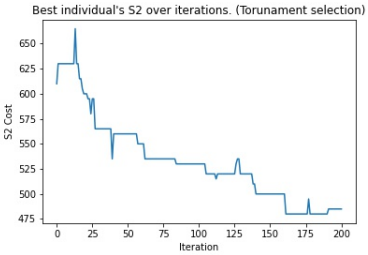


(b) Evolución `H5`.

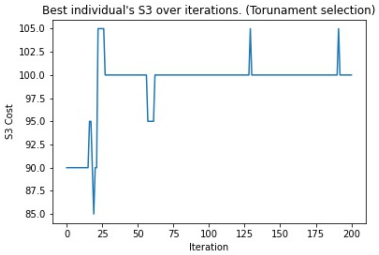
Figura 61: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



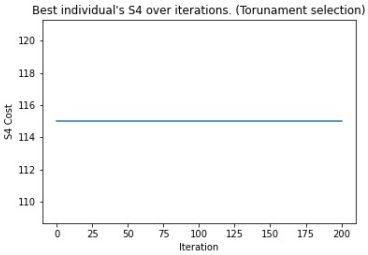
(a) Evolución S1.



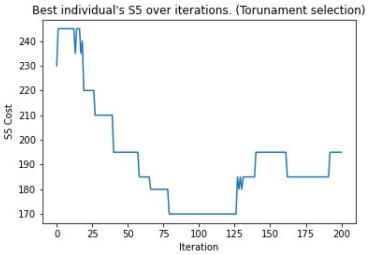
(b) Evolución S2.



(c) Evolución S3.

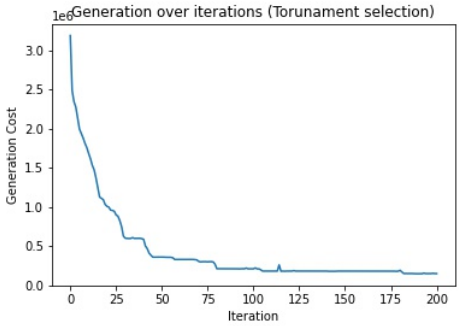


(d) Evolución S4.

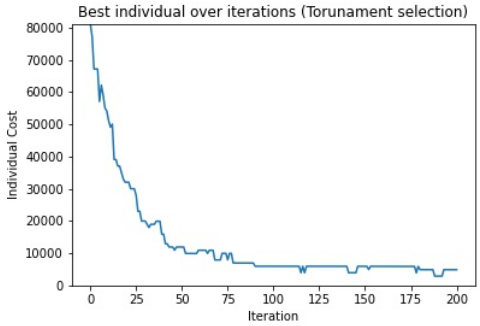


(e) Evolución S5.

Figura 62: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

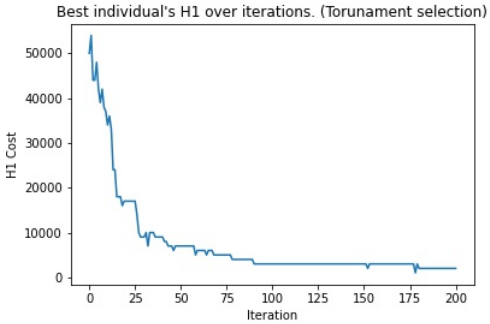


(a) Coste de las sucesivas generaciones

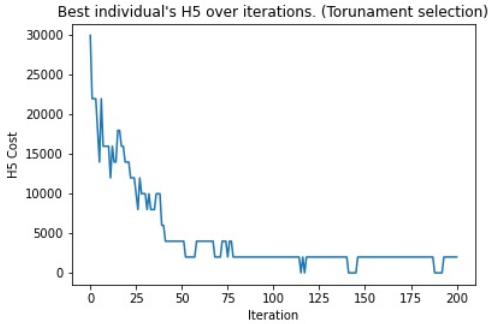


(b) Coste del mejor individuo

Figura 63: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

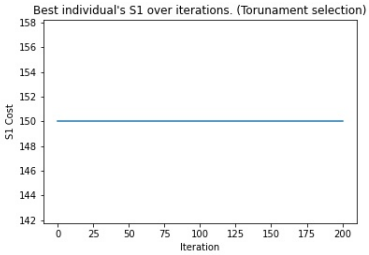


(a) Evolución `H1`.

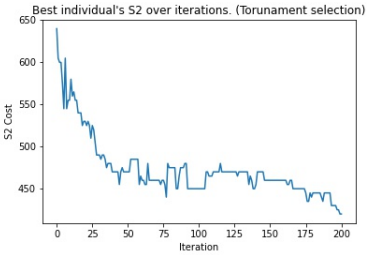


(b) Evolución `H5`.

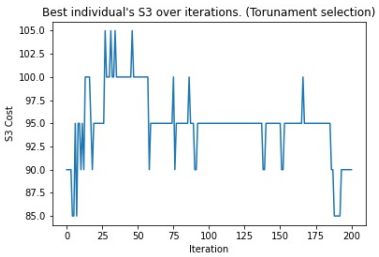
Figura 64: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



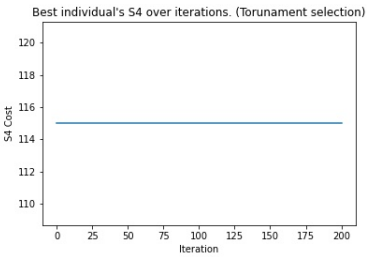
(a) Evolución S1.



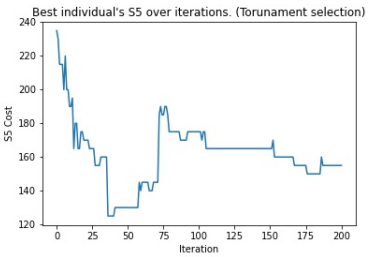
(b) Evolución S2.



(c) Evolución S3.

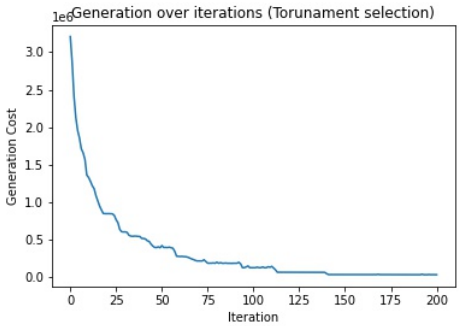


(d) Evolución S4.

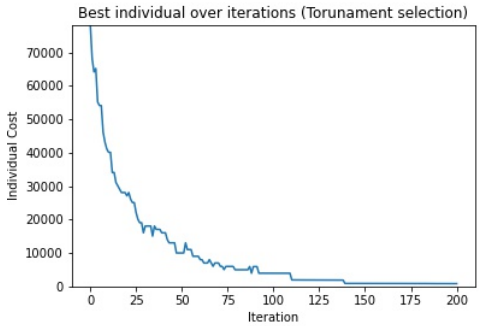


(e) Evolución S5.

Figura 65: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

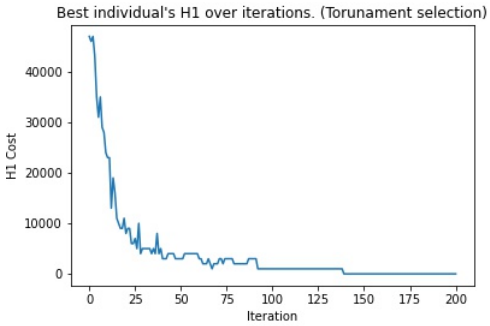


(a) Coste de las sucesivas generaciones

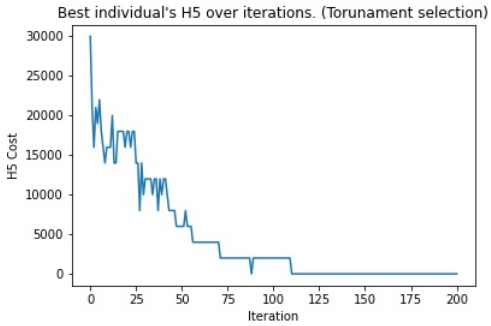


(b) Coste del mejor individuo

Figura 66: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

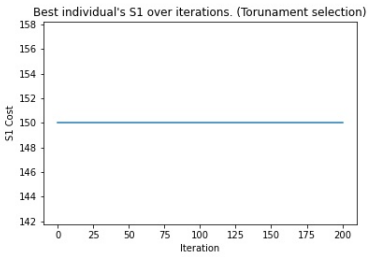


(a) Evolución `H1`.

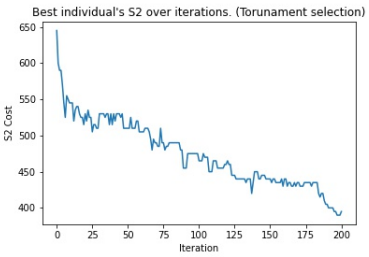


(b) Evolución `H5`.

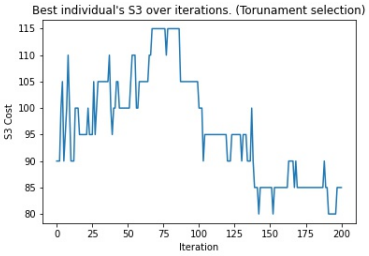
Figura 67: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



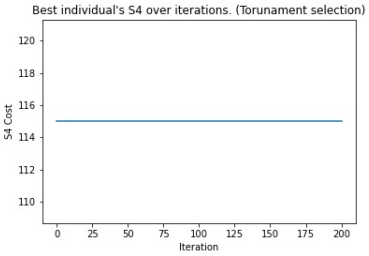
(a) Evolución S1.



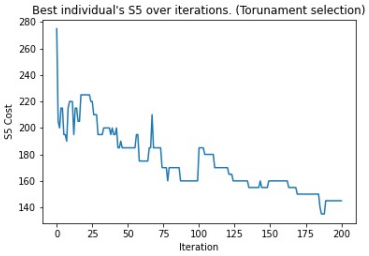
(b) Evolución S2.



(c) Evolución S3.

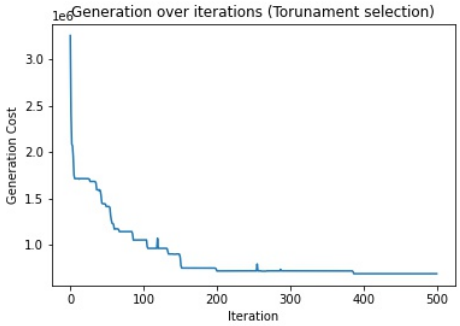


(d) Evolución S4.

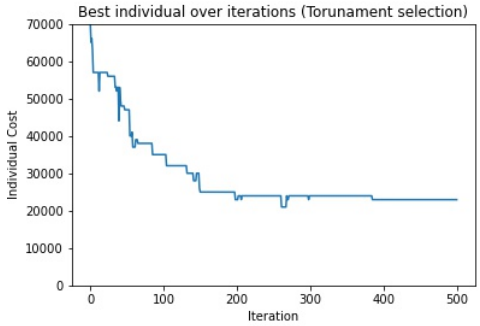


(e) Evolución S5.

Figura 68: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

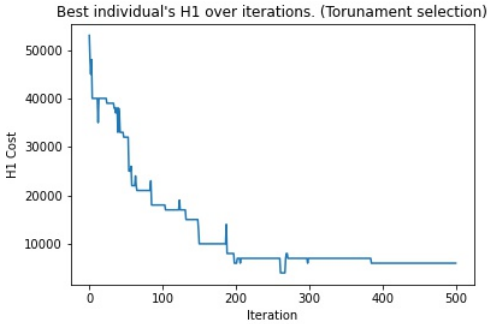


(a) Coste de las sucesivas generaciones.

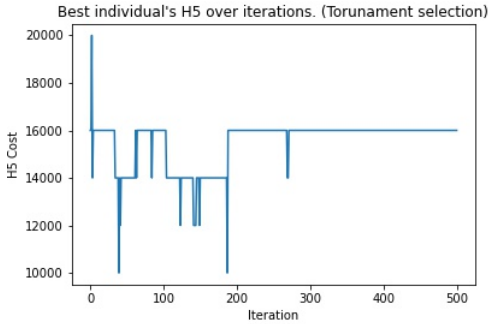


(b) Coste del mejor individuo.

Figura 69: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

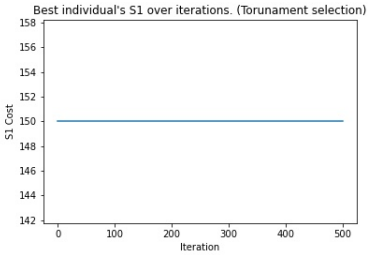


(a) Evolución `H1`.

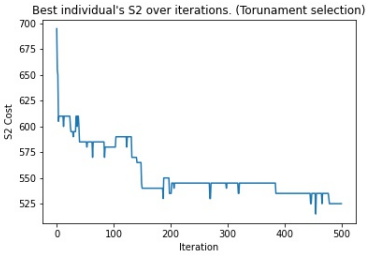


(b) Evolución `H5`.

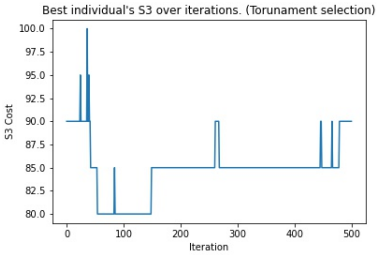
Figura 70: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



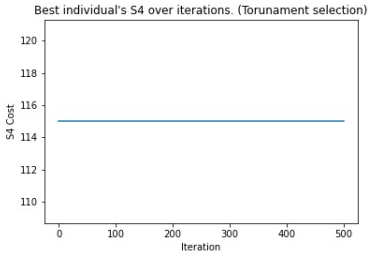
(a) Evolución S1.



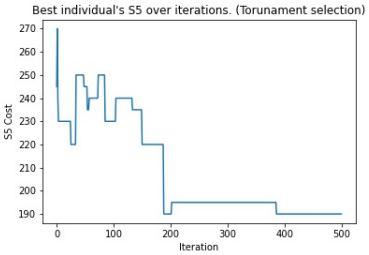
(b) Evolución S2.



(c) Evolución S3.

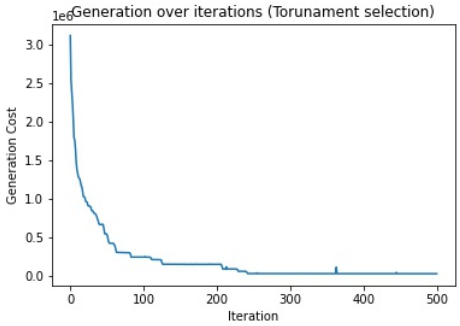


(d) Evolución S4.

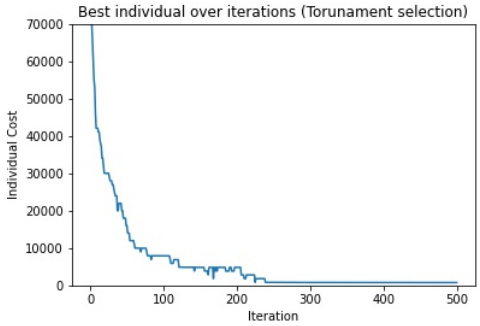


(e) Evolución S5.

Figura 71: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

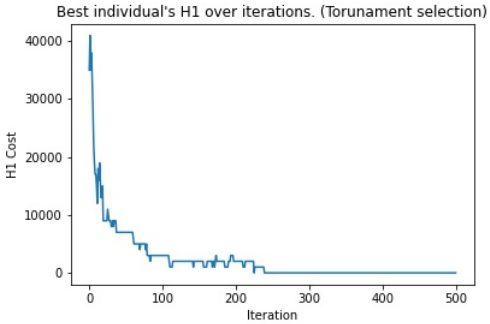


(a) Coste de las sucesivas generaciones

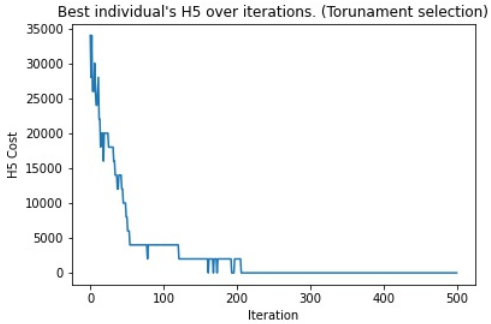


(b) Coste del mejor individuo

Figura 72: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

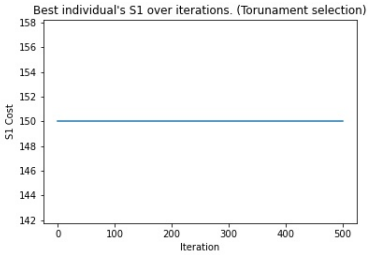


(a) Evolución `H1`.

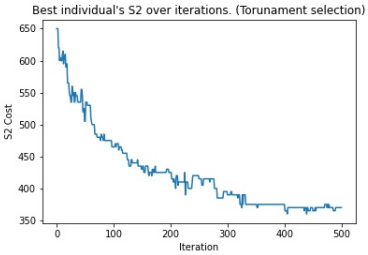


(b) Evolución `H5`.

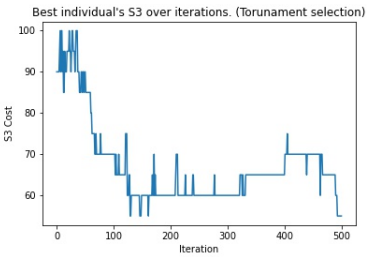
Figura 73: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



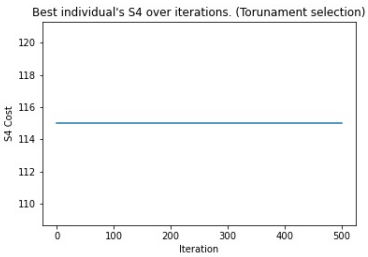
(a) Evolución S1.



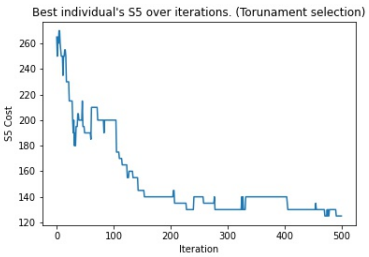
(b) Evolución S2.



(c) Evolución S3.

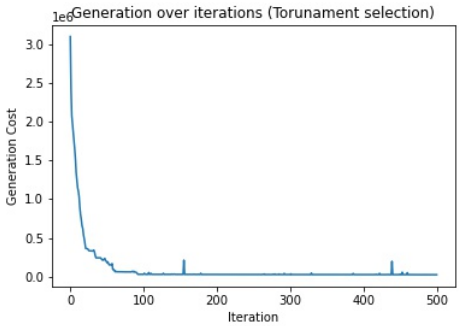


(d) Evolución S4.

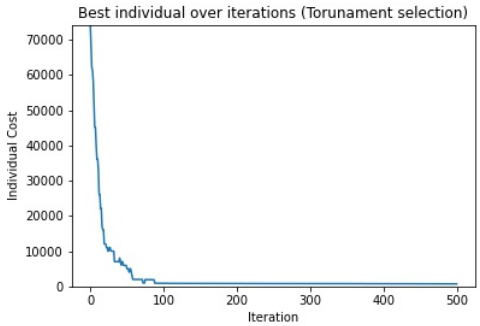


(e) Evolución S5.

Figura 74: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

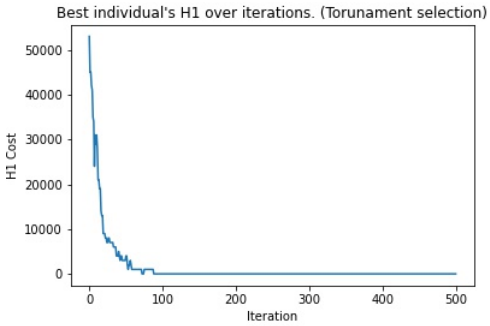


(a) Coste de las sucesivas generaciones

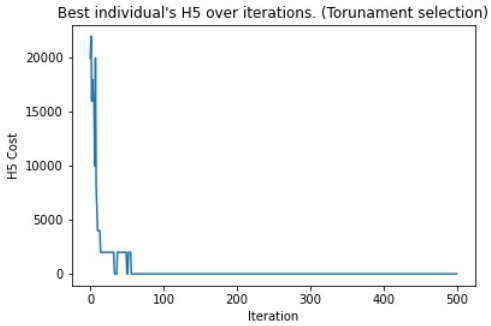


(b) Coste del mejor individuo

Figura 75: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

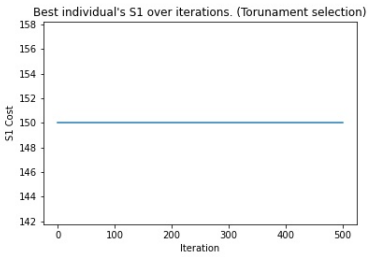


(a) Evolución `H1`.

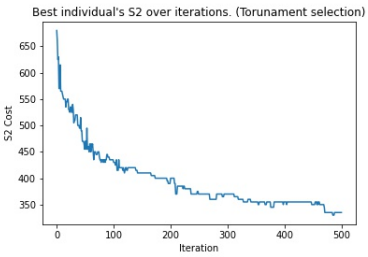


(b) Evolución `H5`.

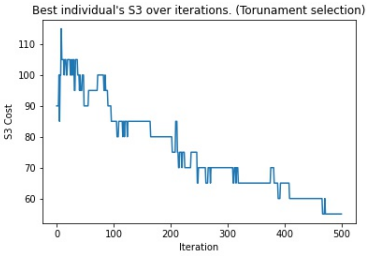
Figura 76: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



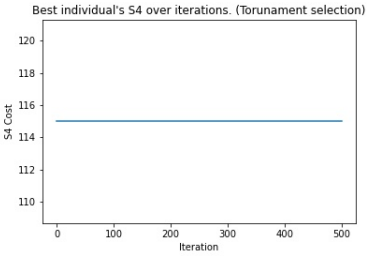
(a) Evolución S1.



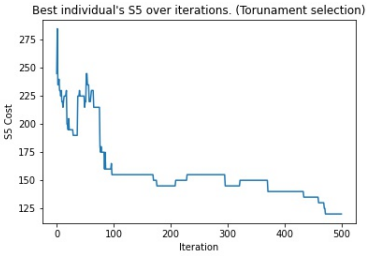
(b) Evolución S2.



(c) Evolución S3.



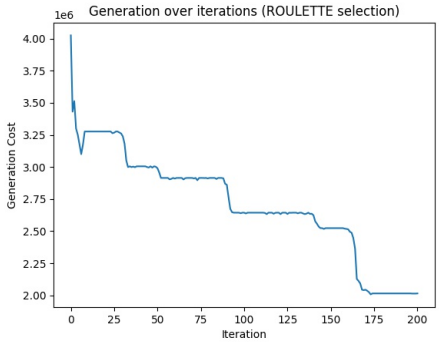
(d) Evolución S4.



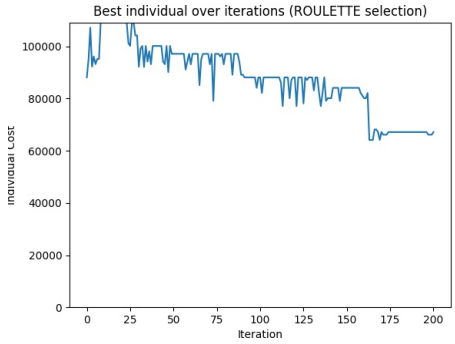
(e) Evolución S5.

Figura 77: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

V-C. *Resultados de teacher_hours_hard*

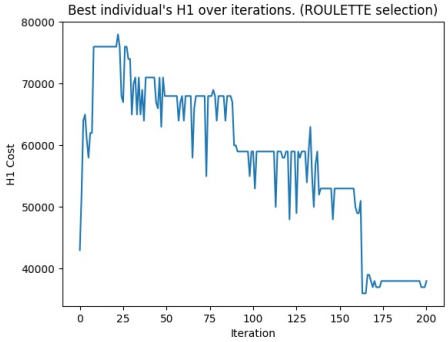


(a) Coste de las sucesivas generaciones.

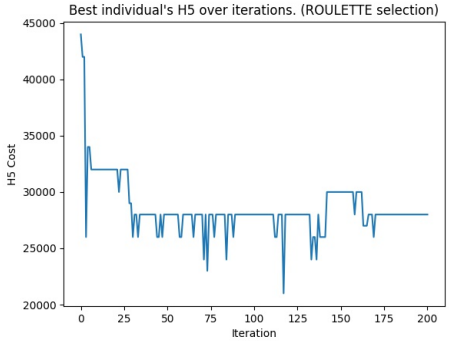


(b) Coste del mejor individuo.

Figura 78: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Evolución de los costes de las generaciones y del mejor individuo de cada generación.

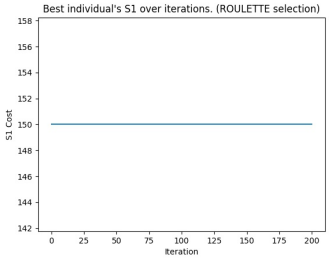


(a) Evolución `H1`.

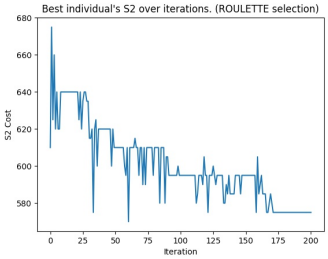


(b) Evolución `H5`.

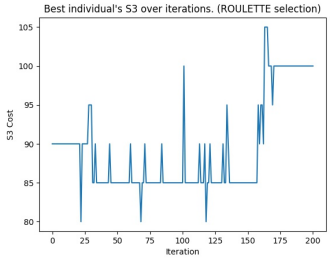
Figura 79: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Costes de las restricciones *hard*.



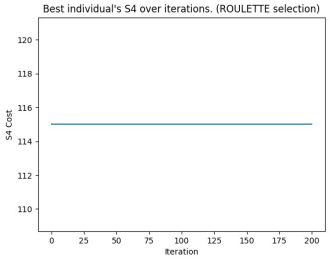
(a) Evolución S1.



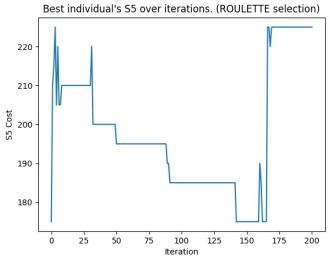
(b) Evolución S2.



(c) Evolución S3.

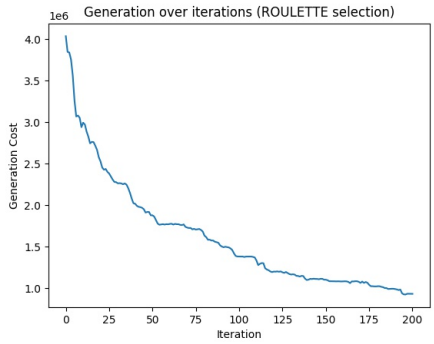


(d) Evolución S4.

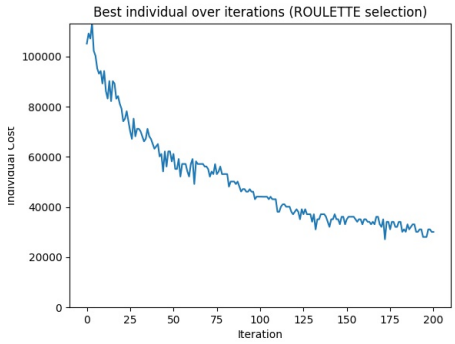


(e) Evolución S5.

Figura 80: **Caso: Roulette, 200 gen., prob. mut.: 0.01.** Evolución de las restricciones *soft* a lo largo de las generaciones.

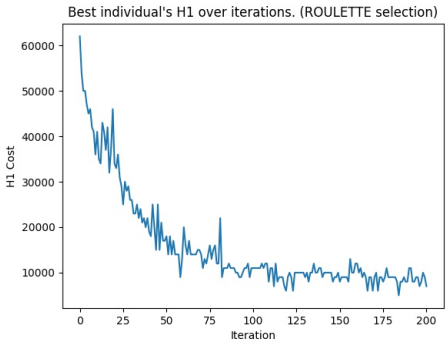


(a) Coste de las sucesivas generaciones.

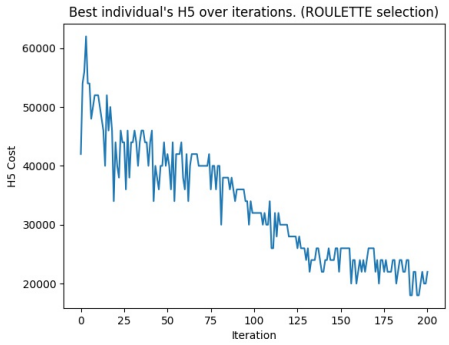


(b) Coste del mejor individuo.

Figura 81: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

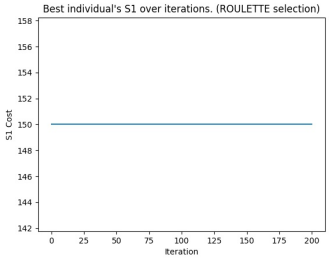


(a) Evolución `H1`.

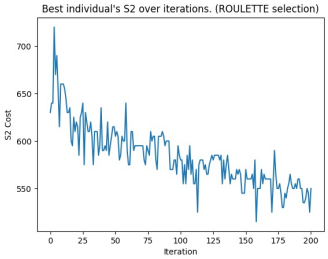


(b) Evolución `H5`.

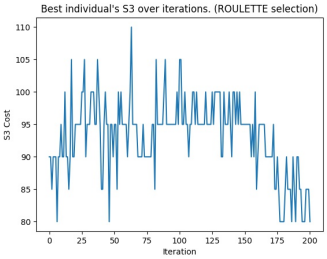
Figura 82: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



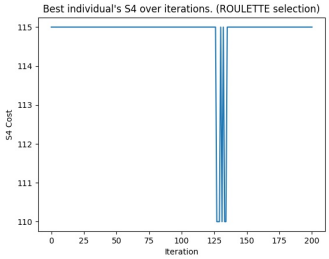
(a) Evolución S1.



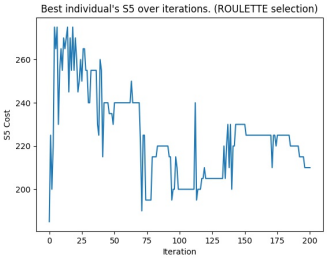
(b) Evolución S2.



(c) Evolución S3.

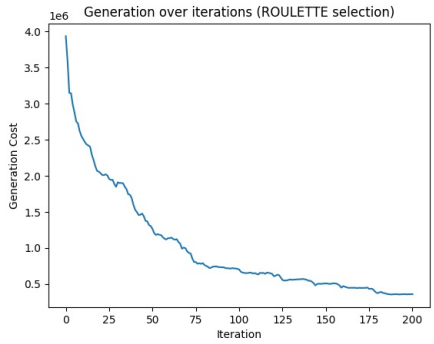


(d) Evolución S4.

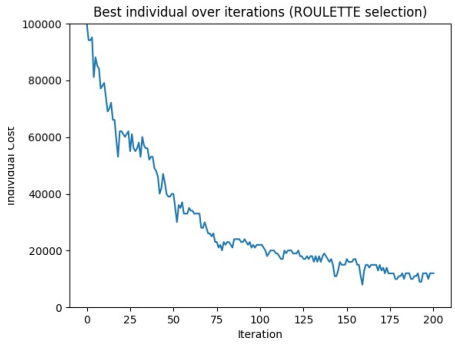


(e) Evolución S5.

Figura 83: **Caso: Roulette, 200 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

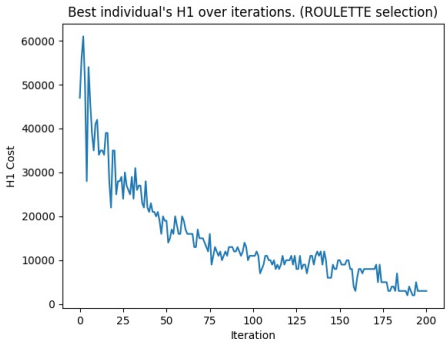


(a) Coste de las sucesivas generaciones

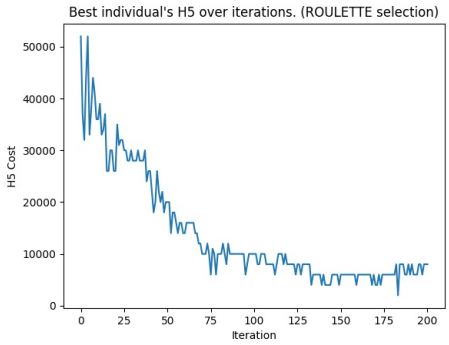


(b) Coste del mejor individuo.

Figura 84: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

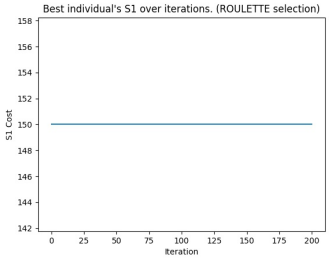


(a) Evolución `H1`.

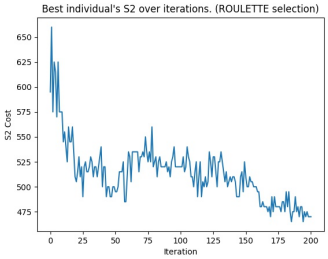


(b) Evolución `H5`.

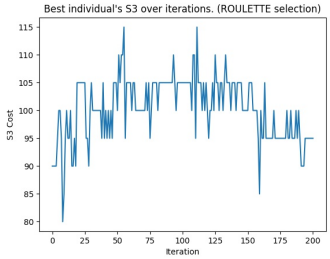
Figura 85: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



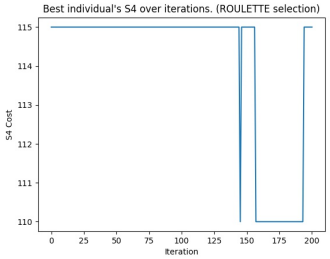
(a) Evolución S1.



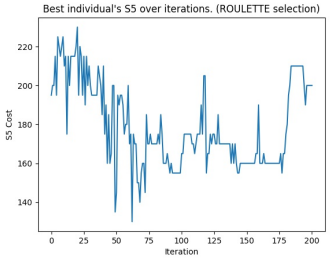
(b) Evolución S2.



(c) Evolución S3.

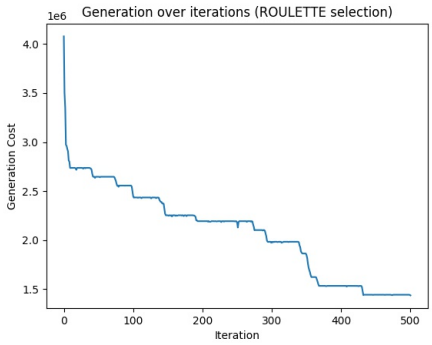


(d) Evolución S4.

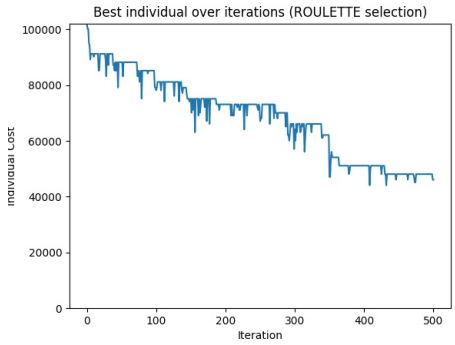


(e) Evolución S5.

Figura 86: **Caso: Roulette, 200 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

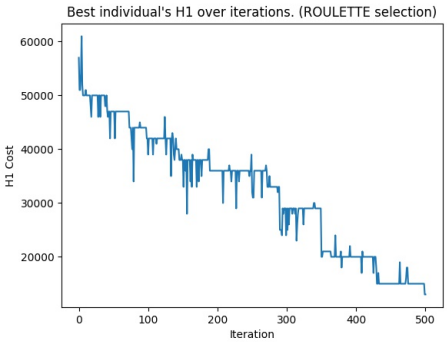


(a) Coste de las sucesivas generaciones

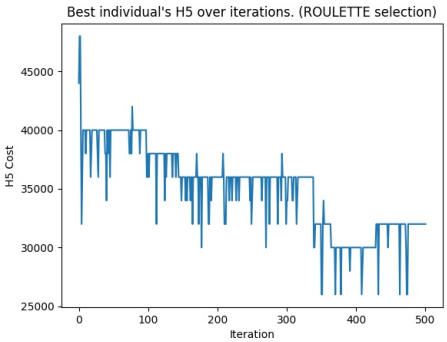


(b) Coste del mejor individuo

Figura 87: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

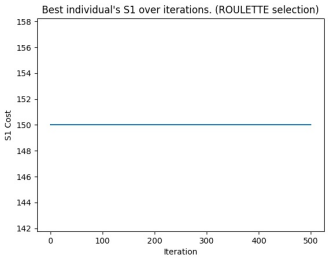


(a) Evolución `H1`.

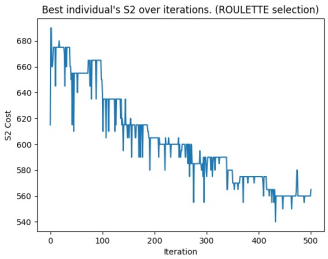


(b) Evolución `H5`.

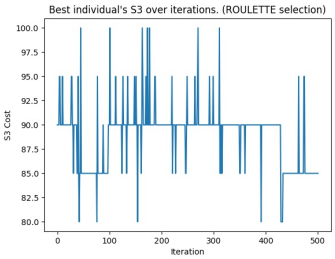
Figura 88: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



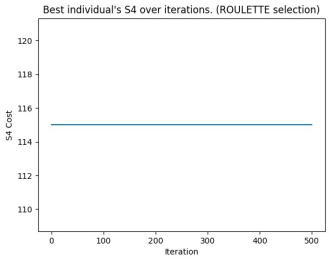
(a) Evolución S1.



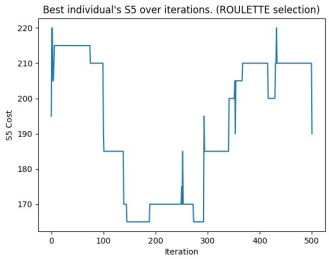
(b) Evolución S2.



(c) Evolución S3.

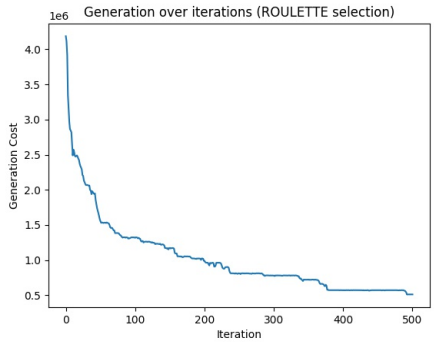


(d) Evolución S4.

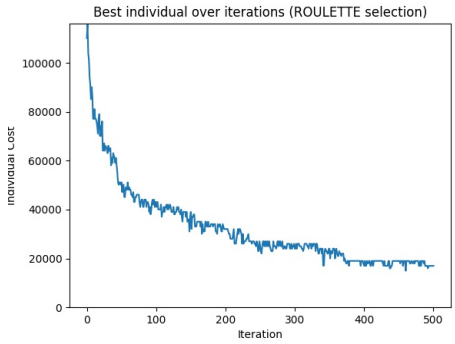


(e) Evolución S5.

Figura 89: **Caso: Roulette, 500 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

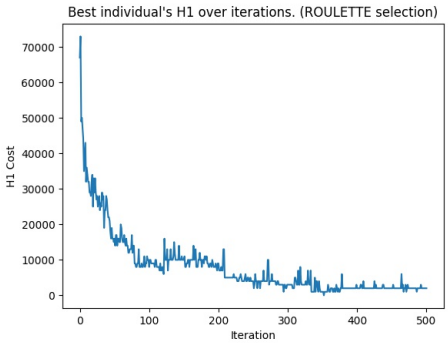


(a) Coste de las sucesivas generaciones.

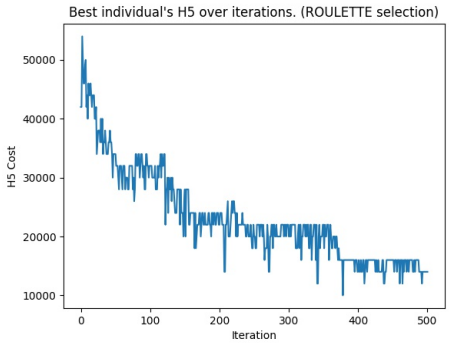


(b) Coste del mejor individuo.

Figura 90: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

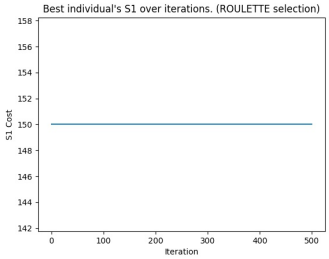


(a) Evolución `H1`.

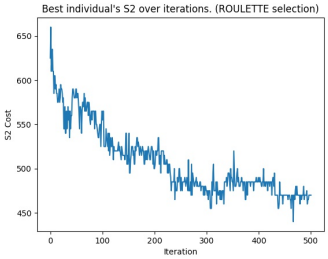


(b) Evolución `H5`.

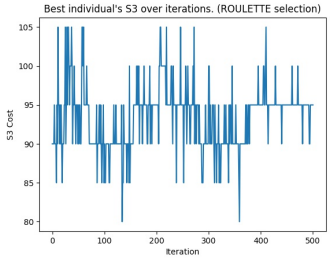
Figura 91: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



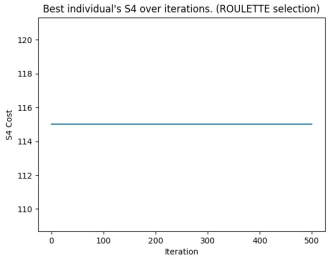
(a) Evolución S1.



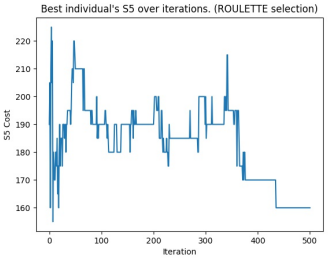
(b) Evolución S2.



(c) Evolución S3.

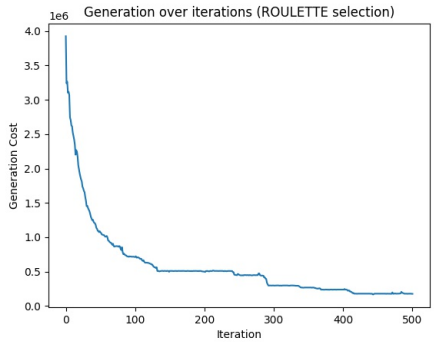


(d) Evolución S4.

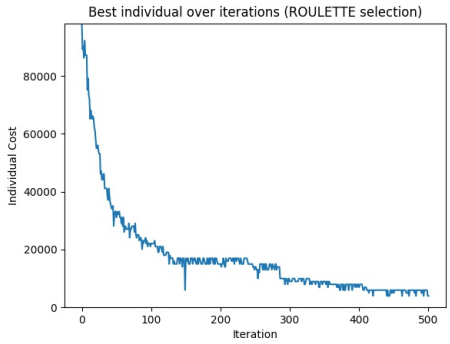


(e) Evolución S5.

Figura 92: **Caso: Roulette, 500 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

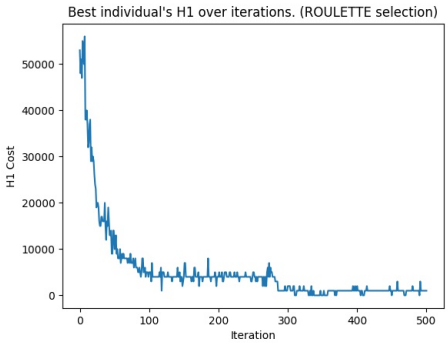


(a) Coste de las sucesivas generaciones

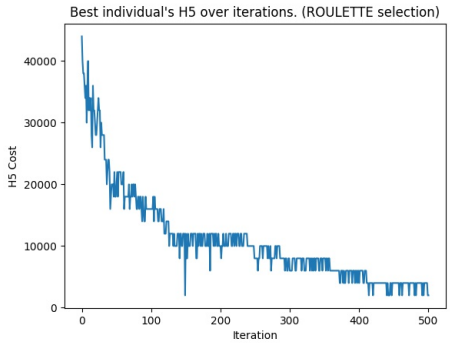


(b) Coste del mejor individuo

Figura 93: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

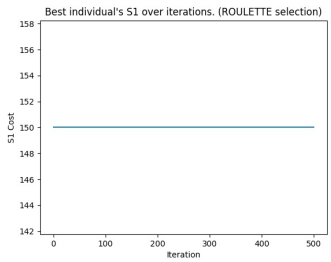


(a) Evolución `H1`.

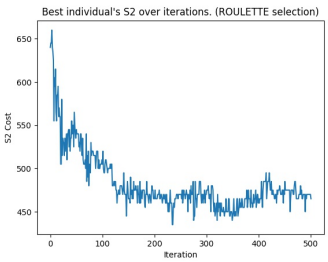


(b) Evolución `H5`.

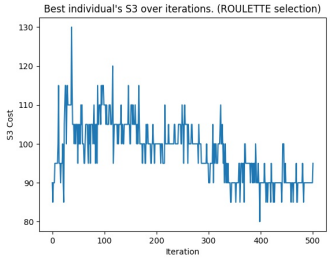
Figura 94: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



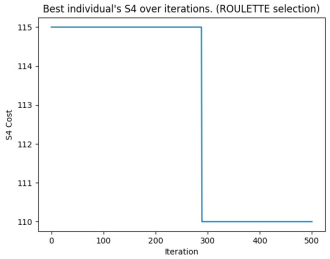
(a) Evolución S1.



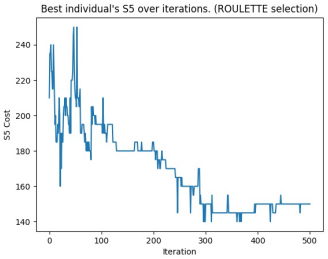
(b) Evolución S2.



(c) Evolución S3.

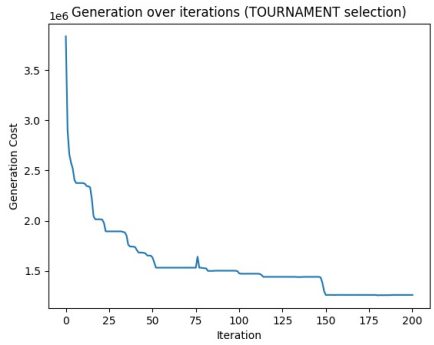


(d) Evolución S4.

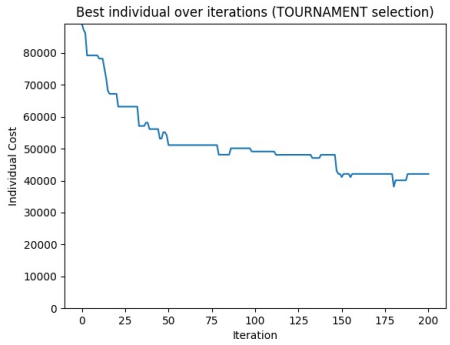


(e) Evolución S5.

Figura 95: **Caso: Roulette, 500 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

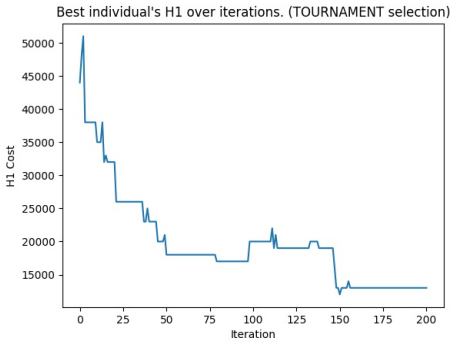


(a) Coste de las sucesivas generaciones

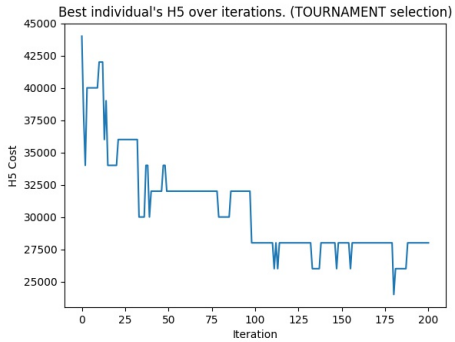


(b) Coste del mejor individuo

Figura 96: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

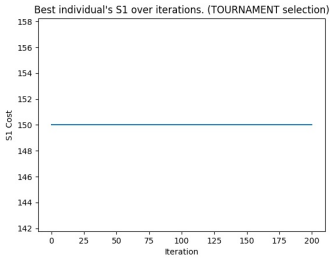


(a) Evolución `H1`.

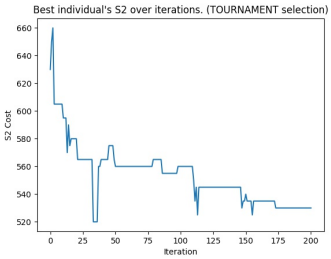


(b) Evolución `H5`.

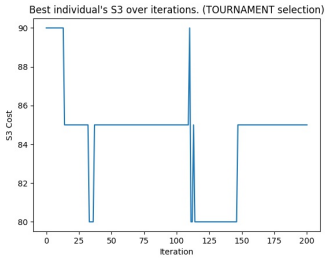
Figura 97: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



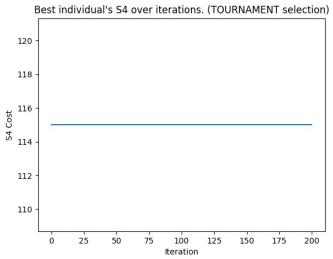
(a) Evolución S1.



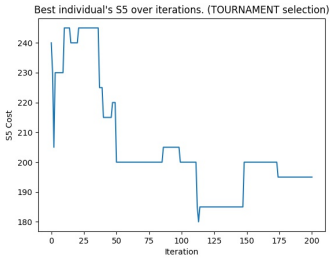
(b) Evolución S2.



(c) Evolución S3.

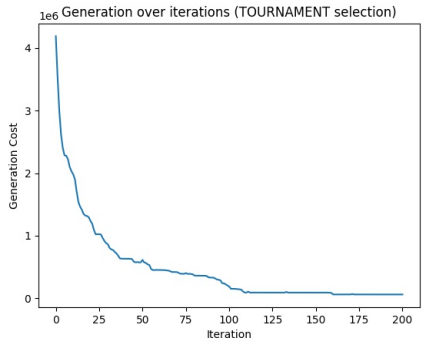


(d) Evolución S4.

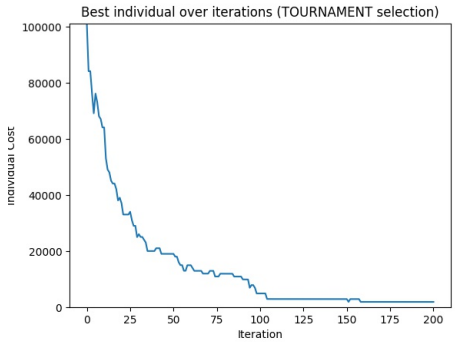


(e) Evolución S5.

Figura 98: **Caso: Tournament, 200 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

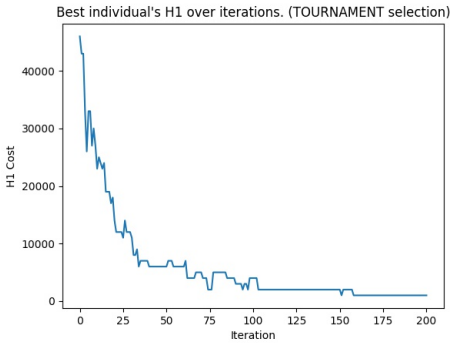


(a) Coste de las sucesivas generaciones

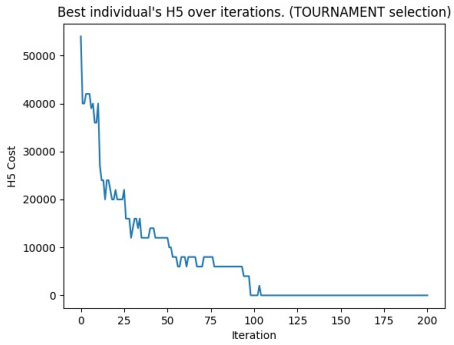


(b) Coste del mejor individuo

Figura 99: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

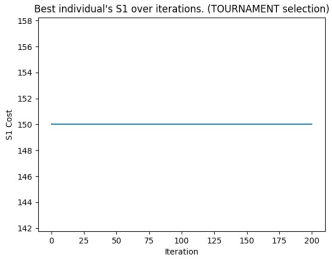


(a) Evolución `H1`.

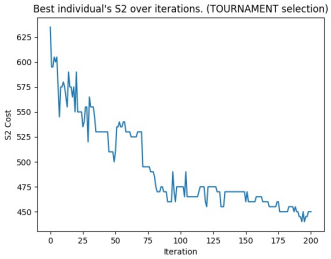


(b) Evolución `H5`.

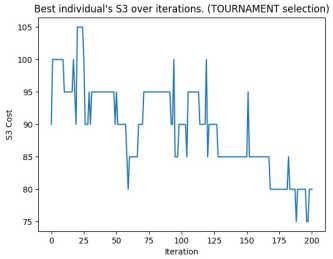
Figura 100: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



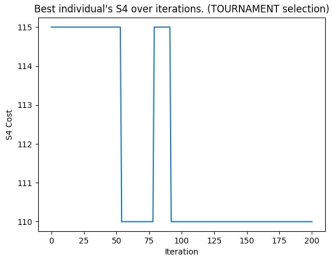
(a) Evolución S1.



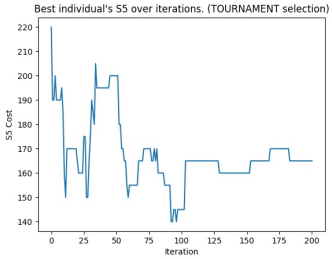
(b) Evolución S2.



(c) Evolución S3.

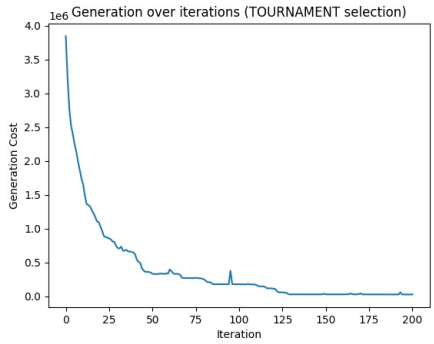


(d) Evolución S4.

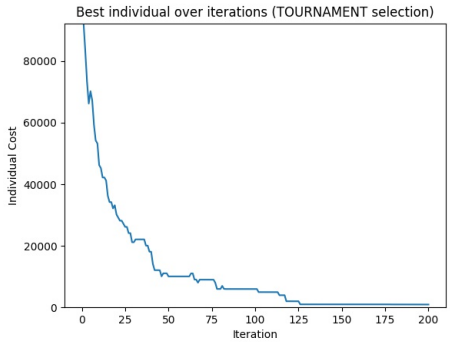


(e) Evolución S5.

Figura 101: **Caso: Tournament, 200 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

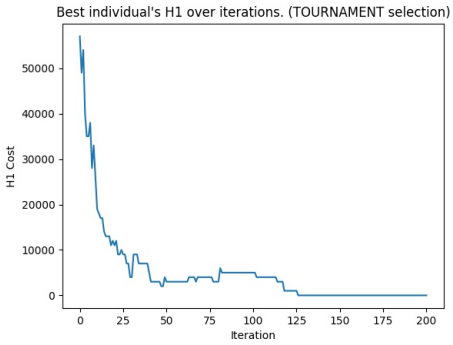


(a) Coste de las sucesivas generaciones

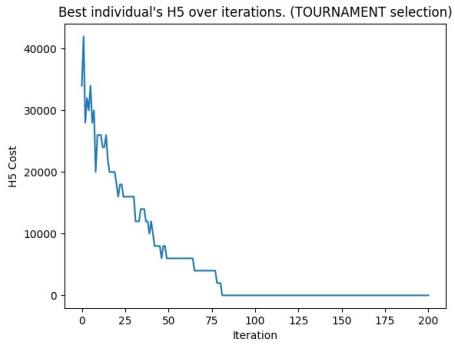


(b) Coste del mejor individuo

Figura 102: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

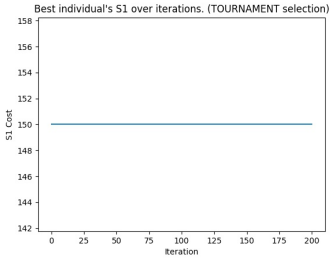


(a) Evolución `H1`.

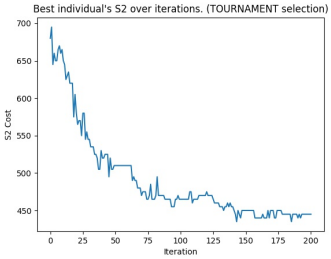


(b) Evolución `H5`.

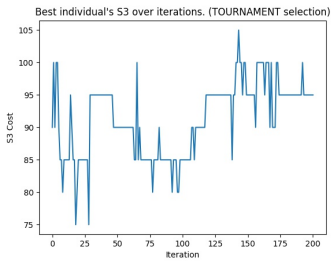
Figura 103: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



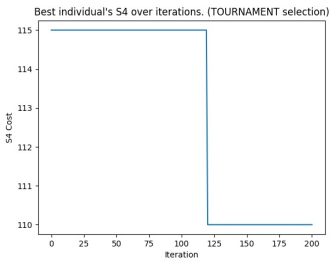
(a) Evolución S1.



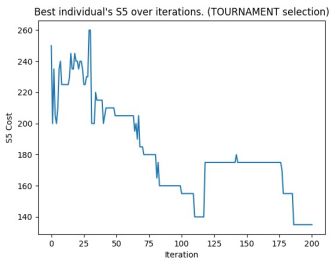
(b) Evolución S2.



(c) Evolución S3.

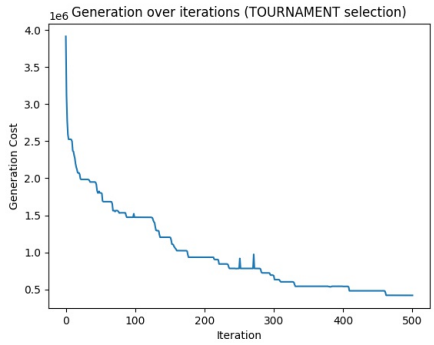


(d) Evolución S4.

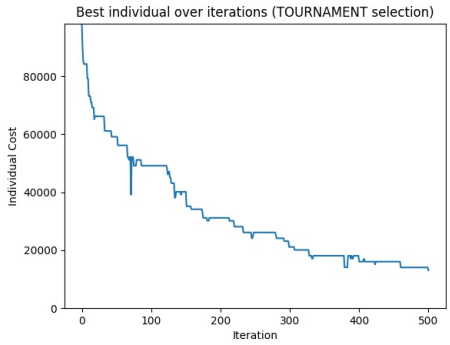


(e) Evolución S5.

Figura 104: **Caso: Tournament, 200 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

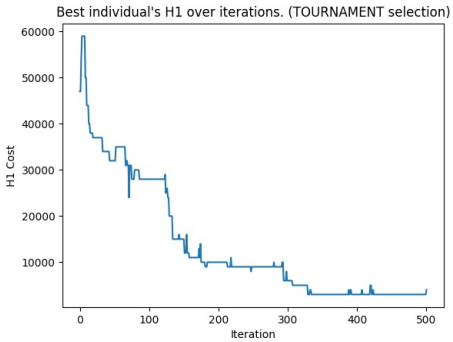


(a) Coste de las sucesivas generaciones.

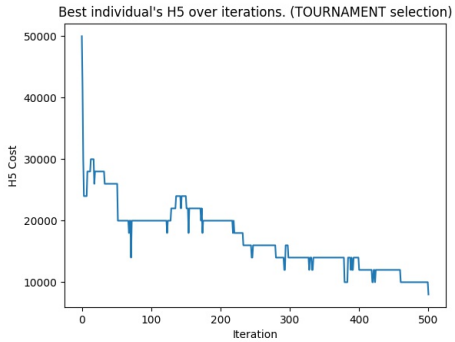


(b) Coste del mejor individuo.

Figura 105: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

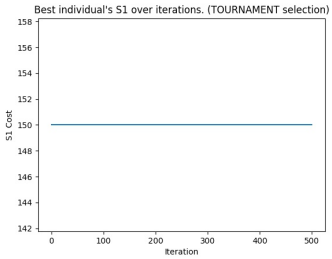


(a) Evolución `H1`.

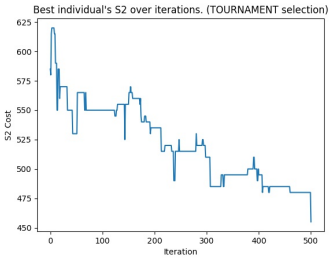


(b) Evolución `H5`.

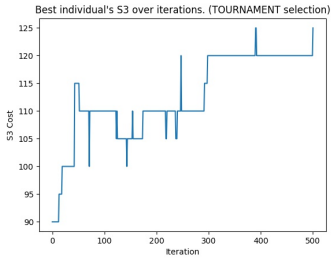
Figura 106: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Costes de las restricciones *hard*.



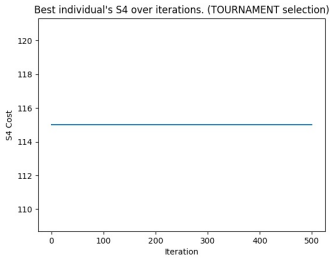
(a) Evolución S1.



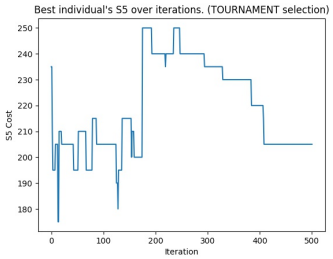
(b) Evolución S2.



(c) Evolución S3.

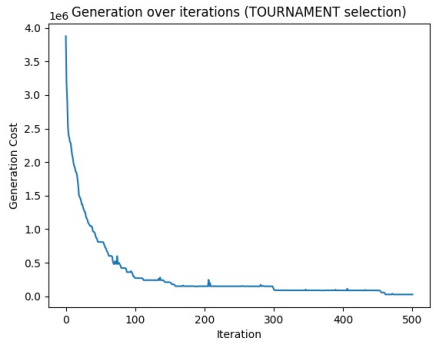


(d) Evolución S4.

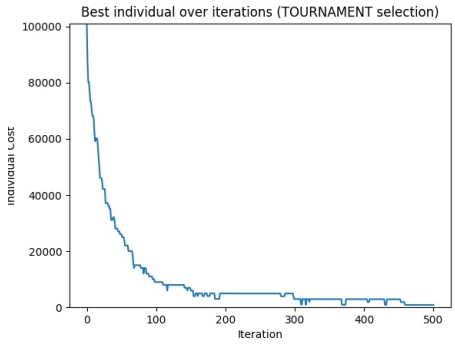


(e) Evolución S5.

Figura 107: **Caso: Tournament, 500 gen., prob. mut.: 0.01**. Evolución de las restricciones *soft* a lo largo de las generaciones.

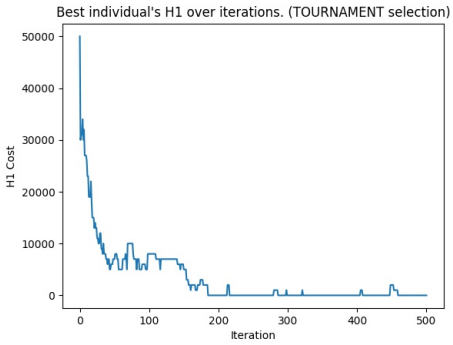


(a) Coste de las sucesivas generaciones

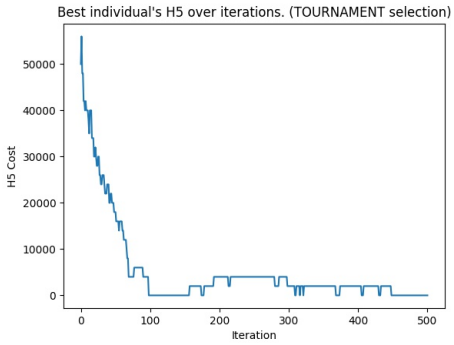


(b) Coste del mejor individuo

Figura 108: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

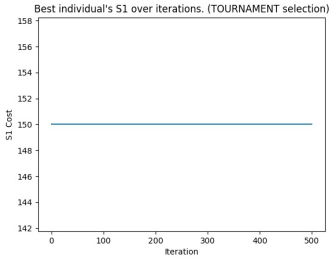


(a) Evolución `H1`.

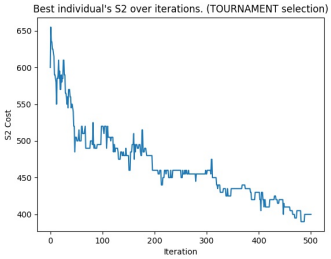


(b) Evolución `H5`.

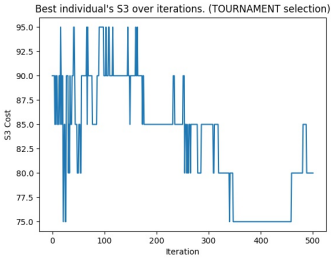
Figura 109: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Costes de las restricciones *hard*.



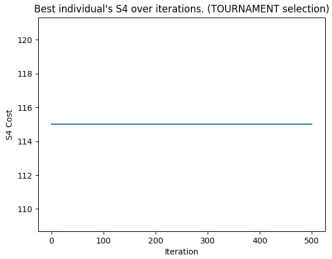
(a) Evolución S1.



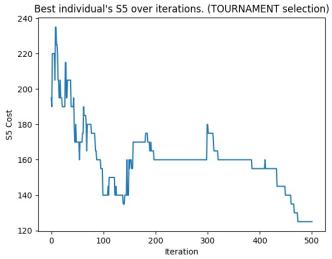
(b) Evolución S2.



(c) Evolución S3.

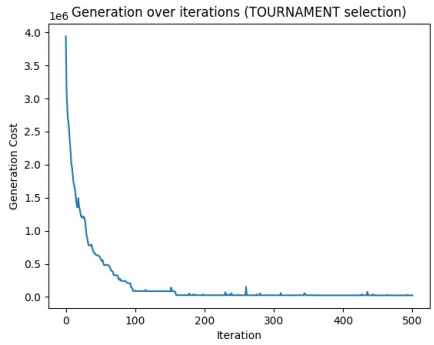


(d) Evolución S4.

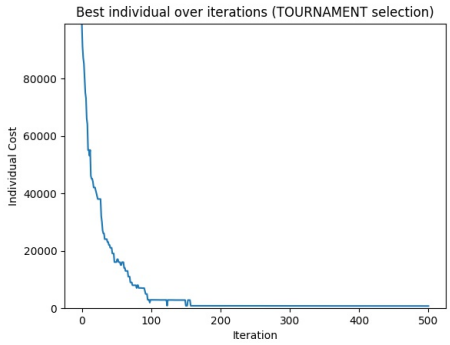


(e) Evolución S5.

Figura 110: **Caso: Tournament, 500 gen., prob. mut.: 0.15**. Evolución de las restricciones *soft* a lo largo de las generaciones.

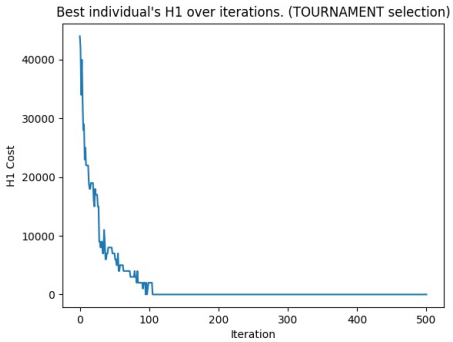


(a) Coste de las sucesivas generaciones

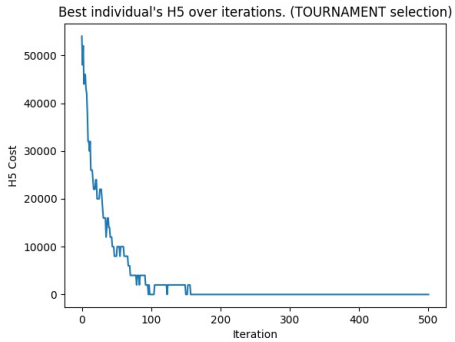


(b) Coste del mejor individuo

Figura 111: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Evolución de los costes de las generaciones y del mejor individuo de cada generación.

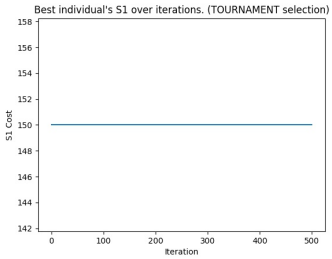


(a) Evolución `H1`.

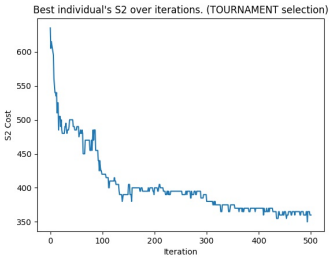


(b) Evolución `H5`.

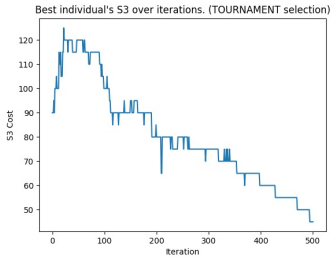
Figura 112: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Costes de las restricciones *hard*.



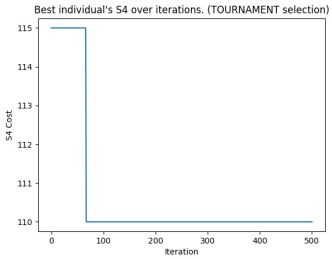
(a) Evolución S1.



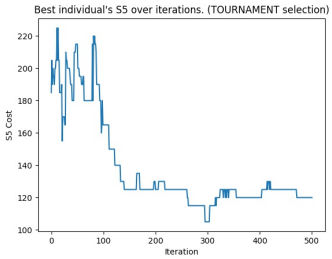
(b) Evolución S2.



(c) Evolución S3.



(d) Evolución S4.



(e) Evolución S5.

Figura 113: **Caso: Tournament, 500 gen., prob. mut.: 0.35**. Evolución de las restricciones *soft* a lo largo de las generaciones.

Referencias

[1] Alex M. Andrew. *Introduction to Evolutionary Computing*. 2004. DOI: 10.1108/03684920410699216.

[2] "Applying evolutionary computation to the school timetabling problem: The Greek case". En: *Computers and Operations Research* 35.4 (abr. de 2008), págs. 1265-1280. ISSN: 03050548.

[3] *Genetic Timetable source code*. https://github.com/magheata/Genetic_Timetable. Accessed: 2020-01-23.