

Memoria de proyectos de innovación y buenas prácticas docentes

A. Datos generales del proyecto de innovación y buenas prácticas docentes

Título	DESARROLLO DE HERRAMIENTAS WEB INTERACTIVAS PARA EL APRENDIZAJE DE ALGORITMOS EN METAHEURÍSTICAS,		
Código	22-15	Fecha de Realización:	29 DE MAYO DE 2023
Coordinación	.Apellidos	MOLINA CABRERA	
	.Nombre	DANIEL	
Tipología	Tipología de proyecto	PIDB BÁSICO	
	Rama del Conocimiento	Ciencias de la Computación e Inteligencia Artificial	
	.Línea de innovación	Dimensión 1. Diseño, organización, desarrollo y evaluación de la docencia.	

B. Objetivo Principal

El desarrollo herramientas pedagógicas para la visualización del comportamiento de la mayoría de los distintos algoritmos de la asignatura de Metaheurísticas pedidos en las entregas de prácticas. Estas herramientas, al favorecer una mejor comprensión de los conceptos tanto generales como particulares de cada tipo de metaheurística.

A diferencia de unas simples visualizaciones estáticas acompañando a las presentaciones magistrales, las herramientas generadas se han caracterizado en dos aspectos:

- Carácter web, que permite al alumnado acceder a dicho contenido en su propia casa, cuando estén repasando la asignatura, ya sea para el desarrollo de las prácticas, o preparándose para la evaluación. Este aspecto es importante, puesto que supone un paso más hacia el denominado aprendizaje ubicuo.
- Carácter interactivo, en el que el/la estudiante pueda cambiar distintas condiciones y ver cómo afecta eso al comportamiento del algoritmo. Cada vez más hay que realizar un mayor esfuerzo para captar la atención y mantener la motivación del alumnado por aprender. Al incorporar la participación activa del alumnado no solo permite aumentar la motivación del alumnado, sino también permite una mayor experimentación y un conocimiento más profundo.

Dada la falta de herramientas adecuadas para nuestros objetivos, hemos desarrollado de nuestras propias herramientas didácticas innovadoras y atractivas que contribuirán al objetivo común propuesto, mejorar el aprendizaje de la asignatura de una manera ágil y flexible que encaje con la distinta estrategia didáctica de cada miembro del equipo docente.

C. Descripción del proyecto de innovación y buenas prácticas docentes

Resumen del proyecto realizado: Objetivos, metodología, logros alcanzados, aplicación práctica a la docencia habitual, etc.

Objetivos:

El objetivo principal de este proyecto de innovación docente es elaborar nuevas herramientas docentes para la enseñanza de la asignatura de Metaheurísticas, así como facilitar su aprendizaje y la relación de los conocimientos adquiridos tanto en el Campus de Ceuta como del Campus de Granada. Para ello nos hemos planteado los siguientes objetivos específicos:

1. Permitir visualizar uno o varios problemas de optimización referencia, que sirven para mostrar el comportamiento de distintas técnicas de optimización sobre ellos.

2. Crear simulaciones webs interactivas que permiten mostrar cómo distintas metaheurísticas abordan la resolución de un problema.
3. Que el componente interactivo permitiría a quien lo use (alumnado o profesorado) configurar distintos parámetros, con lo que se permitiría ver cómo varía el comportamiento del algoritmo en función de dichos parámetros.
4. Que el uso de dichas herramientas sirvan de apoyo a las explicaciones del profesorado.
5. Que la utilización de dichas herramientas sirvan de apoyo a la comprensión de conceptos por parte del alumnado, fuera del aula.
6. Evaluar el material didáctico al final del semestre mediante cuestionarios para comprobar su evolución en el aprendizaje de los conocimientos adquiridos.

Por otro lado, también planteamos que los alumnos identifiquen los conocimientos y competencias adquiridos y la utilidad y aplicación de esas tareas y habilidades en su futuro profesional, gracias al carácter práctico.

Como **metodología** se ha planteado una metodología de desarrollo de software ágil el desarrollo de distintas herramientas interactivas, que se han ido mostrando como complemento a las explicaciones de cada práctica, en la semana posterior a su explicación inicial. También se han planteado encuestas para conocer la opinión tanto del estudiantado como de compañeros profesores conocedores del temario de la asignatura.

Sobre el estudiantado, hemos pasado la encuesta <https://encuestas.ugr.es/index.php/487614?lang=es>, y para conocer la opinión de compañeros conocedores de la asignatura (por haberla impartido otros años, o ser muy cercana a su investigación). Dicha encuesta es: <https://encuestas.ugr.es/index.php/754513?lang=es>.

Mostraremos los resultados en ambas encuestas en la parte de resultados.

Logros alcanzados:

El software planteado se ha creado de forma exitosa cumpliendo todos los objetivos previstos, está disponible para ser usado localmente pero también se ha ofrecido un servidor para ser usado sin instalar, aunque requería más recursos de los inicialmente previstos.

Aplicación práctica a la docencia habitual:

Por su enfoque práctico, la aplicación práctica es inmediata.

Summary of the Project (In English):

Objectives:

The main objective of this teaching innovation project is to develop new teaching tools for the teaching of the subject of Metaheuristics, as well as to facilitate its learning and the relationship of the knowledge acquired both in the Campus of Ceuta and the Campus of Granada. For this we have set the following specific objectives:

1. to allow visualizing one or several benchmark optimization problems, which serve to show the behavior of different optimization techniques on them.
2. To create interactive web simulations that allow to show how different metaheuristics approach the resolution of a problem.
3. That the interactive component would allow the user (students or teachers) to configure different parameters, which would allow to see how the behavior of the algorithm varies depending on these parameters.

4. That the use of these tools would serve as a support for the teacher's explanations.
5. The use of these tools to support the understanding of concepts by the students outside the classroom.
6. Evaluate the didactic material at the end of the semester by means of questionnaires to check their evolution in the learning of the acquired knowledge.

On the other hand, we also propose that students identify the knowledge and skills acquired and the usefulness and application of these tasks and skills in their professional future, thanks to the practical nature.

As a **methodology**, an agile software development methodology has been proposed, the development of different interactive tools, which have been shown as a complement to the explanations of each practice, in the week after the initial explanation. Surveys have also been raised to know the opinion of both the student body and fellow teachers knowledgeable about the subject's agenda.

About the students, we have passed the survey <https://encuestas.ugr.es/index.php/487614?lang=es>, and to know the opinion of colleagues who know the subject (for having taught it other years, or being very close to their research). The survey is: <https://encuestas.ugr.es/index.php/754513?lang=es>.

We will show the results of both surveys in the results section.

Achievements:

The proposed software has been created successfully fulfilling all the planned objectives, it is available to be used locally but a server has also been offered to be used without installing, although it required more resources than initially planned.

Practical application to regular teaching:

Because of its practical approach, the practical application is immediate.

D. Resultados obtenidos

El resultado software está públicamente disponible en <https://codeberg.org/dmolina/innovaMH> con licencia de software libre. En particular, está disponible con la licencia GNU Affero General Public License, que posibilita que otros/as docentes puedan utilizarlo como punto de partida, siempre que publiquen sus cambios con la misma licencia.

Se divide en 3 partes:

Evaluación de fitness:

En esta parte, se permite a los propios estudiantes comprobar si están evaluando bien las soluciones:

Por ejemplo, para uno de los problemas (los alumnos eligen el problema a optimizar) permite validar tanto la solución como evaluar la solución, así pueden comprobar que su evaluación es correcta.

Por ejemplo, para uno de los problemas:

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema **APC**

Introduzca un dataset: **diabetes**

Introduzca la solución de tamaño 8 con números entre 0.0 y 1.0 separados mediante comas:

solution to test: **1, 0.3, 1, 1, 0.1, 0.5, 1**

Submit Query

Error in solution

"Error, tamaño no correcto, faltan 1"

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema **APC**

Introduzca un dataset: **diabetes**

Introduzca la solución de tamaño 8 con números entre 0.0 y 1.0 separados mediante comas:

solution to test: **1, 0.3, 1, 1, 0.1, 0.5, 1, 0.5**

Submit Query

Partición	Train [en %]	Test [en %]	Reducción [en %]	Fitness
1	68.24	62.99	12.5	52.89
2	67.43	66.88	12.5	56.01
3	69.71	63.64	12.5	53.41
4	62.87	68.18	12.5	57.05
5	69.16	69.08	12.5	57.76

Para el otro problema:

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema **QAP**

Introduzca un fichero de entrada: **chr22a.dat**

Introduzca la solución mediante una permutación de 1 a 22 separados mediante comas:

solution to test: **21 22 10 1 6 8 18 14 15 3 20 2 17 13 7 11 5 19 8 16 4 12**

Submit Query

Error in solution

"Valor 8 repetido"

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema **QAP**

Introduzca un fichero de entrada: **chr22a.dat**

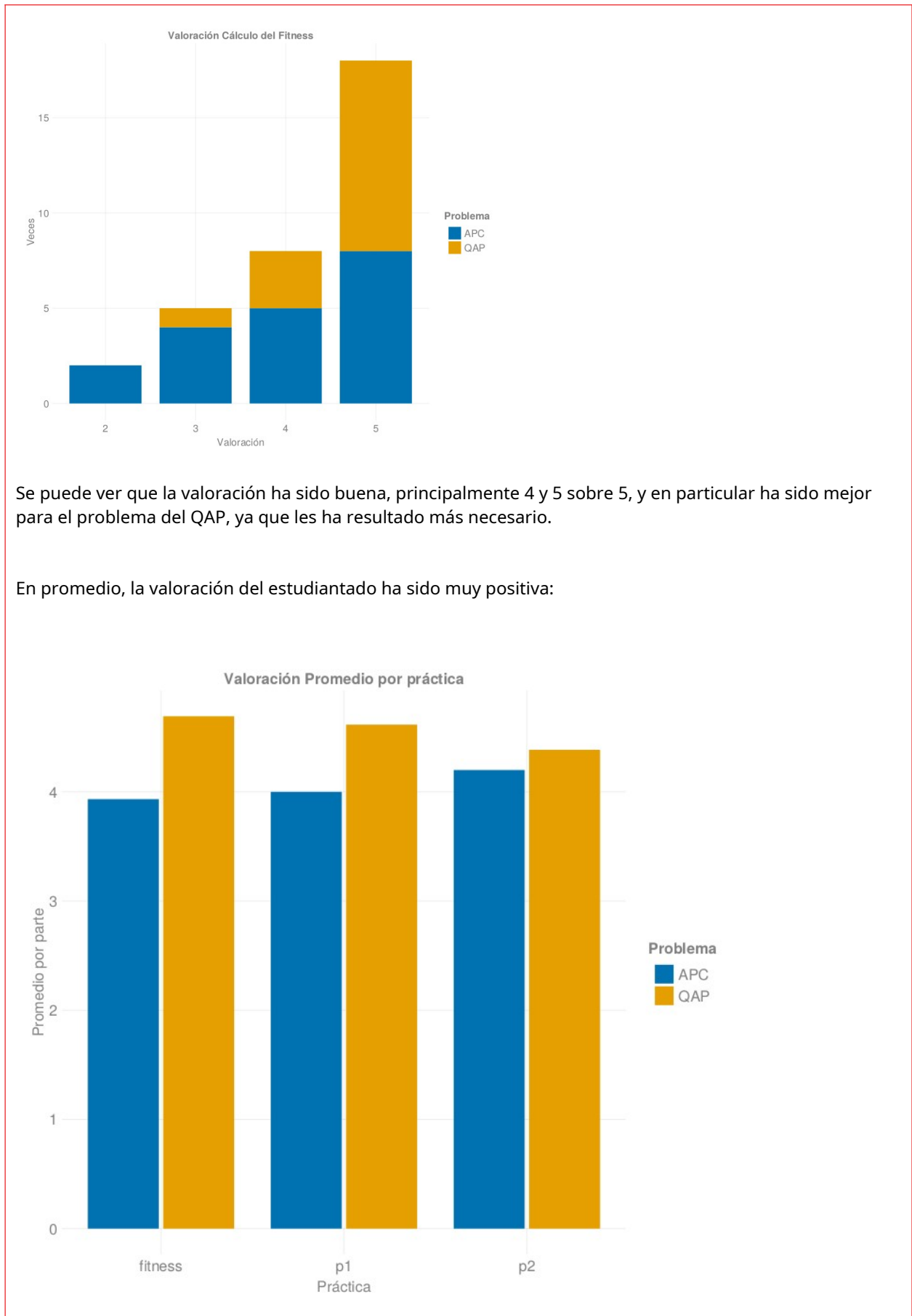
Introduzca la solución mediante una permutación de 1 a 22 separados mediante comas:

solution to test: **21 22 10 1 6 8 18 14 15 3 20 2 17 13 7 11 5 19 9 16 4 12**

Submit Query

Coste: 12220
Error: 6064
Fitness (Ratio): 0.9850552306692657

La recepción de dicho componente ha sido muy buena, indico los resultados de la encuesta sobre dicha parte:



y no solo es bien valorado, si no que mayoritariamente recomendarías su uso a otros compañeros:

Parte 1:

En esta parte se mostrará cómo se pueden usar los algoritmos vistos para resolver un problema clásico visto en clase, El Viajante de Comercio.

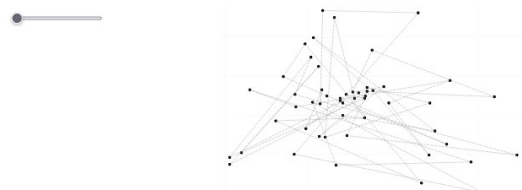
El problema del Viajante de Comercio

El problema del Viajante de Comercio (*Traveling Salesman Problem*, TSP) es un problema conocido de la literatura, en el que existe un viajante que debe de recorrer una serie de ciudades y desea encontrar la mejor ruta para ello. Por tanto, el problema consiste en:

- Encontrar la ruta más corta entre todas las ciudades.
- Pasar por cada ciudad únicamente una vez.
- Volver a la ciudad origen.

El objetivo es encontrar la asignación S que minimiza:

$$\sum_{i=0}^{N-1} Distancia(S_i, S_{i+1}) + Distancia(S_N, S_0)$$



Para abordar el problema usaremos un conjunto de ciudades de TSP sacadas del [TSPLib](#)

Escoge un problema: **52 locations in Berlin (Groetschel)**

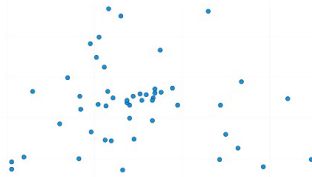
read_TSP (generic function with 1 method)

Vamos a leer sus datos (suponiendo que existe la función `read_TSP` que devuelve el tamaño, las posiciones (para pintar visualmente) y las distancias entre ellas (calculadas como las distancias euclídeas a partir de sus posiciones).

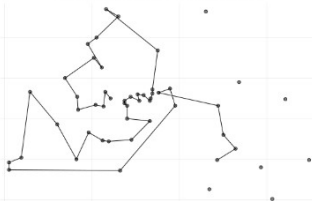
`N, Position_cities, Distances_cities = read_TSP(fname):`

Vamos a visualizar el problema:

52 locations in Berlin (Groetschel)



52 locations in Berlin (Groetschel)



Saved animation to /home/daniel/mh222/greedy.gif

Algoritmo de Búsqueda Local

En este caso vamos a aplicar el modelo de Búsqueda Local visto en clase, en particular el enfoque primero mejor.

Primero vamos a crear una solución aleatoria, y luego aplicamos un operador que cambie la solución un poco, la comparemos y nos quedamos la mejor de ambas, y luego se vuelve a aplicar lo mismo, hasta alcanzar un número de soluciones.

Definimos la función que muta:

mutate (generic function with 1 method)

```
function mutate(solution)
    result = copy(solution)
    N = length(solution)
    pos1 = rand(1:N)
    pos2 = rand(1:N)

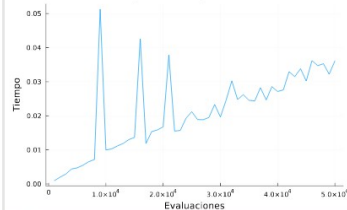
    # Compruebo que sea diferente de la primera
    while pos2 == pos1
        pos2 = rand(1:N)
    end

    # Intercambia posiciones
    result[pos1] = solution[pos2]
    result[pos2] = solution[pos1]
    return result
end
```

Midiendo el rendimiento

Ahora vamos a medir cómo de rápido es el algoritmo. Para ello vamos a probar los tiempos con distinto número máximo de evaluaciones (maxevals) y ver cómo funciona.

Tiempo con Búsqueda Aleatoria



```
begin
    # Recorro desde 1000 hasta 50000
    evals_time = collect{1:1000:1000:50_000}
    local time_alg = Float64[]

    for evals in evals_time
        time = @elapsed optim_random(Distances_cities, evals)
        push!(time_alg, time)
    end

    plot(evals_time, time_alg, legend=false, title="Tiempo con Búsqueda Aleatoria", xlabel="Evaluaciones", ylabel="Tiempo")
end
```

Tabla comparativa

Vamos a medir los algoritmos con distintas evaluaciones y medir los tiempos.

Resultado con 34335 evaluaciones

Algoritmo	Tiempo	Mejor Fit.
Greedy	2.584e-04	10200.890
Random	2.990e-02	25990.361
Búsqueda Local	2.178e-02	10524.587

```
begin
    function times(distances)
        df = DataFrame{name::String[], tiempo::Float32[], best::Float32[]}
        seed!(100)

        time_greedy = @elapsed _, fit_greedy = greedy(distances)
        push!(df, (name="Greedy", tiempo=time_greedy, best=fit_greedy))
        seed!(100)

        time_random = @elapsed _, fit_random = optim_random(distances, evals_comp)
        push!(df, (name="Random", tiempo=time_random, best=fit_random))
        seed!(100)

        time_BL = @elapsed _, fit_BL = busquedaLocal(distances, evals_comp)
        push!(df, (name="Búsqueda Local", tiempo=time_BL, best=fit_BL))
    end

    PrettyTables.pretty_table{HTML, df, headers=["Algoritmo", "Tiempo", "Mejor Fit."], title="Resultado con $(evals_comp) evaluaciones", formatters = (PrettyTables.ft_printf("%2.3e", [2]), PrettyTables.ft_printf("%2.3f", [3]))}
end
times(Distances_cities)
```

Preguntas

▼ Preguntas

Question:

¿Qué te parecen los resultados? Indica lo que te parece correcto.

- La búsqueda local mejora a la aleatoria
- La búsqueda local no mejora a la aleatoria

Keep working on it!

The answer is not quite right.

Se puede ver que no solo se visualizan resultados (dinámicos, la distribución de ciudades puede cambiar) sino también se muestra código en un lenguaje de alto nivel (Julia, similar a Python) e incluso preguntas a los alumnos que el sistema comprueba si son correctas o no.

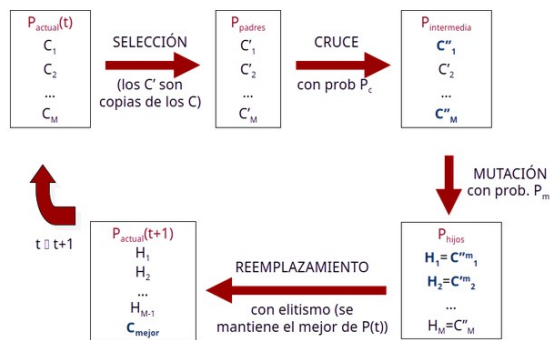
Parte 3:

En esta tercera parte se muestran dos algoritmos nuevos, que se piden en la práctica 2:

Algoritmo Genético Generacional

El esquema se visualiza a continuación:

Mientras no se cumpla el criterio de parada, es decir, mientras no se haya generado un número máximo de evaluaciones, se aplican sucesivamente las siguientes fases:



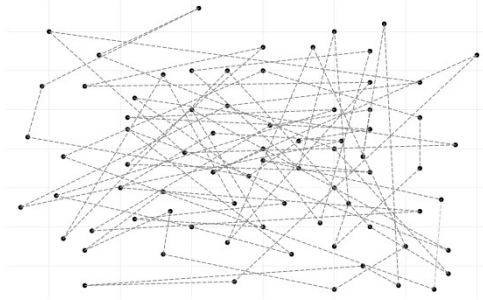
- Selección: Se seleccionan soluciones de la población original para constituir una población de igual tamaño. Se seleccionan de forma aleatoriamente, pero normalmente con mayor tendencia a escoger las que presenten mayor fitness.
- Cruce: Se cruzan pares de soluciones para crear nuevas soluciones, reemplazando a las originales. Dado que tras la selección las soluciones están desordenadas, se pueden combinar por pares (la 1 con la 2ª, la 3ª con la 4ª, ...). No todas son cruzadas, solo el P_c % de pares.
- Mutación: Con una cierta probabilidad P_mut muy baja, se mutan las nuevas soluciones, mediante un cambio muy pequeño.
- Reemplazo: Las nuevas soluciones, la nueva población, reemplaza a la población anterior. Para evitar perder la mejor solución anterior, se copia en la nueva si no está.

```

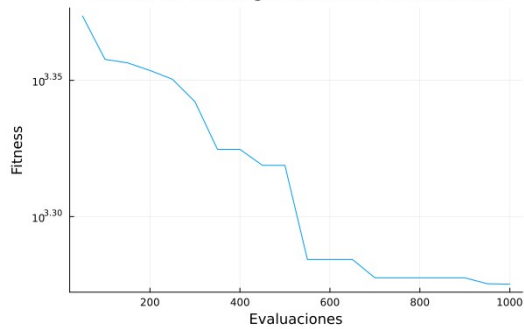
AGG (generic function with 1 method)
function AGG(dimension, popsize, pcross, pmut; maxevals=50_000,
historico=Tuple{Int,AbstractVector{<:Integer},Float64}[], copia_historico=false)
    pop = zeros{Int, popsize, dimension}
    # Crea la nueva población
    newpop = copy(pop)
    # Inicializo el vector
    for i in 1:popsize
        pop[i,:] = new_solution(dimension)
    end
    # Vector de fitness
    fitness_pop = [fitness(sol) for sol in eachsol(pop)]
    mejor = argmin(fitness_pop)
    # Inicio el número de evaluaciones
    evals = popsize
    if copia_historico
        push!(historico, (evals, pop[argmin(fitness_pop), :],
            obten_mejor_fitness(fitness_pop)))
    end
    while evals < maxevals
        # Aplico selección por torneo
        for i in indices(newpop)
            # Copio el ganador del torneo
            posi = torneo_binario(fitness_pop)
            newpop[i, :] = pop[posi, :]
        end
        # Empleo a cruzar de forma consecutiva
        total_cruzar = (popsize*pcross)
        i = 1
        while i < total_cruzar
            sol1 = newpop[i, :]
            sol2 = newpop[i+1, :]
            newsol1, newsol2 = crossover_0X(sol1, sol2)
            newpop[i, :] = newsol1
            newpop[i+1, :] = newsol2
            i += 2
        end
        # Mutación
        total_mutar = popsize + 10
        for i in enumerate(total_mutar)
            posi = posicion_aleatoria(newpop)
            newpop[posi, :] = mutation(newpop[posi, :])
        end
    end
  
```

Evaluaciones AGG:

Resultados AGG: 5.00E+01 evals, distancia 2363.8



Gráfica de Convergencia del AG Generacional



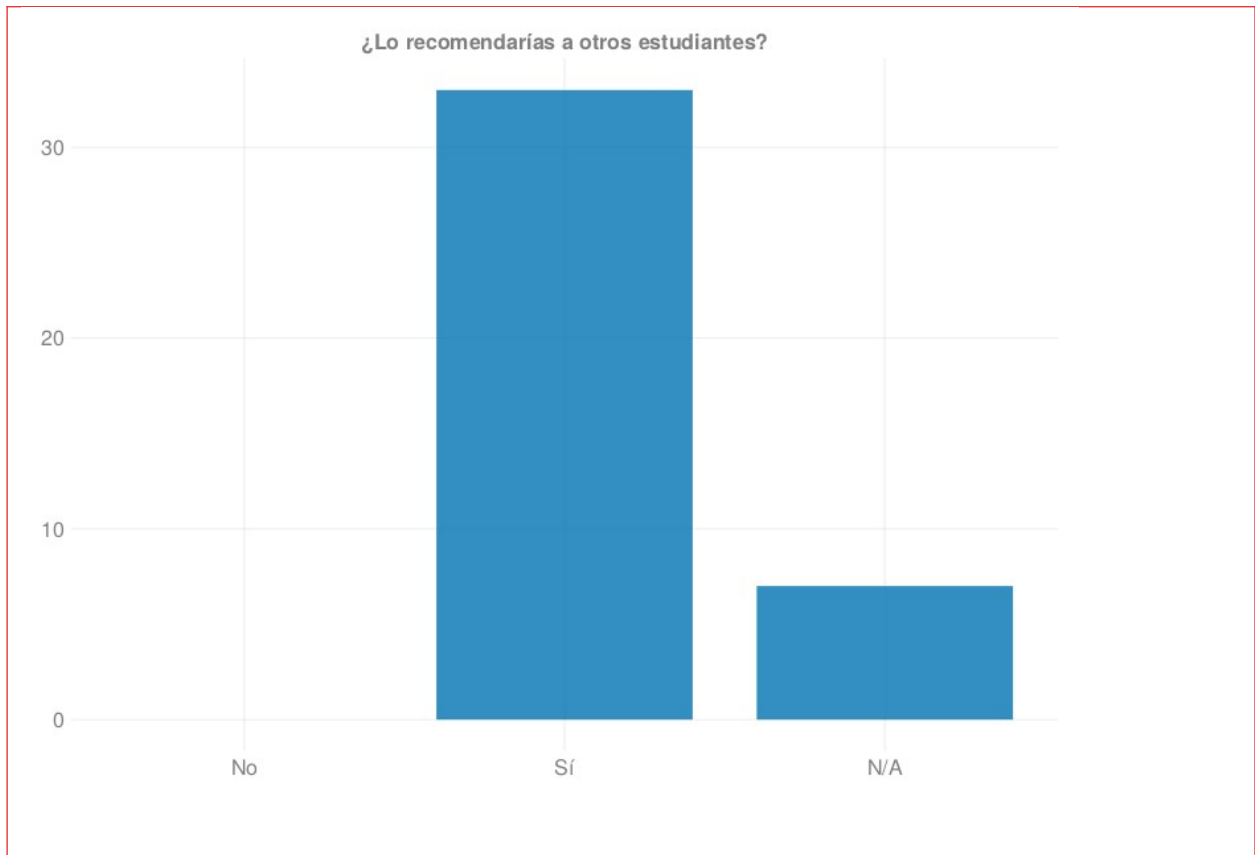
AG Generacional Vs AG Estacionario

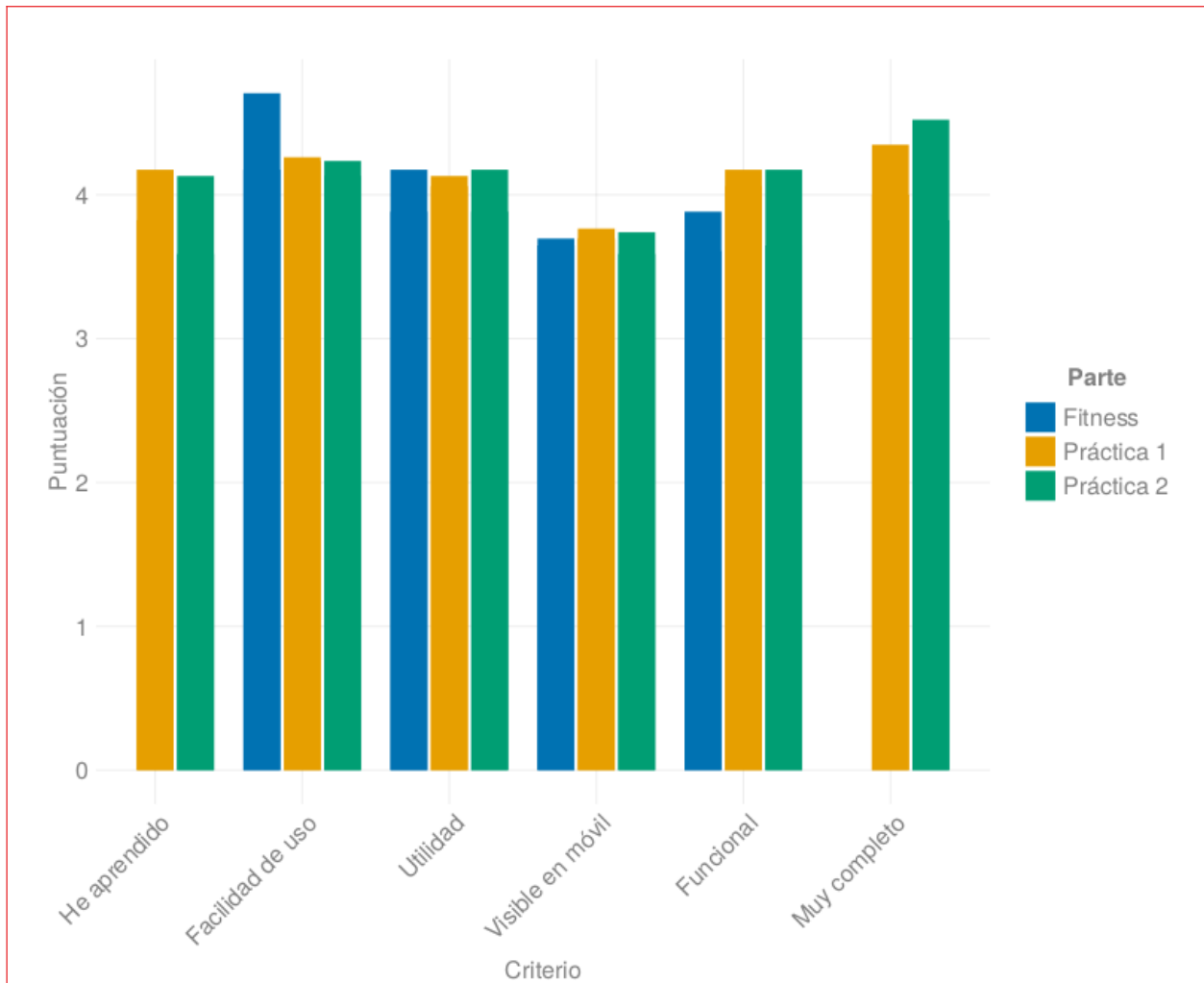
Vamos a comparar los resultados de ambos para el problema `eil76.tsp` con 1000 evaluaciones.

	Algoritmo	Mejor Fitness	Tiempo
1	"Generacional"	1884.57	0.711558
2	"Estacionario"	1620.25	0.320271

A continuación mostramos los resultados de las encuestas a los/las estudiantes:

En resumen, se les ha pedido a los alumnos que valoren distintos atributos de cada parte, y las valoraciones han sido muy buenas:



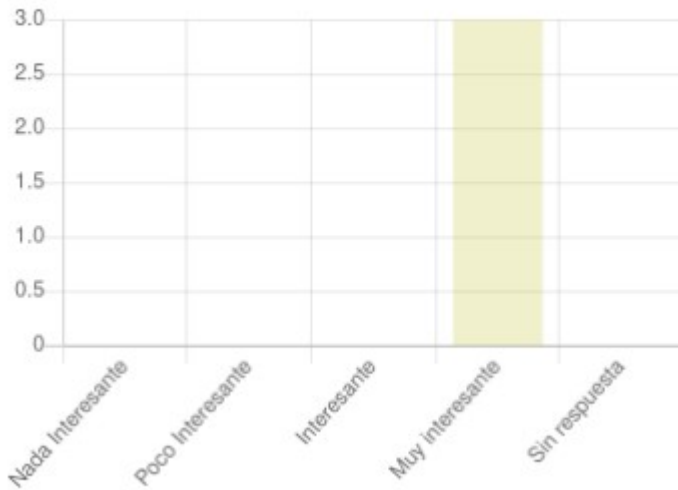


Como se puede ver, los resultados son muy semejantes, la primera se ha considerado más útil pero menos funcional que el resto.

Con respecto a las opiniones de los compañeros:

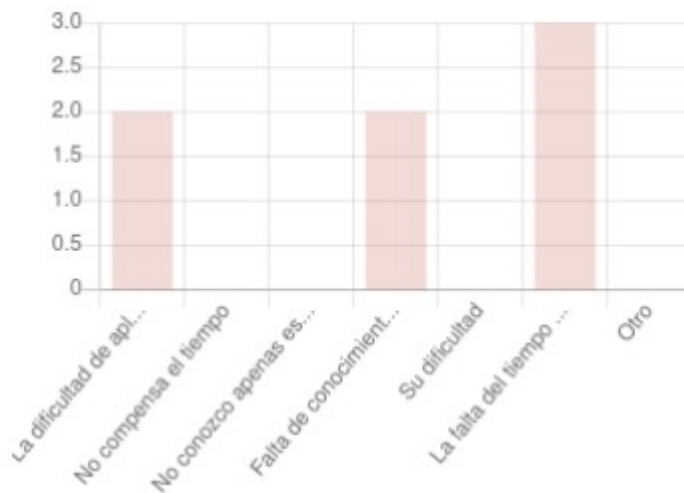
La mayoría lo ha considerado muy útil:

¿Cómo valoras la iniciativa de implicar al estudiantado con herramientas interactivas para la docencia?



Están muy de acuerdo en este tipo de herramientas, pero aunque consideran que es muy interesante para ser aplicado existen distintos motivos por lo que no lo hacen:

¿Cuáles son los principales problemas para el uso de este tipo de herramientas?



Es decir, se considera que el principal motivo para no usarlos es la falta del tiempo del docente, seguido por la dificultad de aplicación (que requiere mucho tiempo) y falta de conocimiento por parte del docente para usarlo.

Results obtained (In English)

The resulting software is publicly available at <https://codeberg.org/dmolina/innovaMH> under a free software license. In particular, it is available under the GNU Affero General Public License, which allows other teachers to use it as a starting point, as long as they publish their changes under the same license.

the same license.

It is divided into 3 parts:

Fitness evaluation:

In this part, the students themselves are allowed to check if they are evaluating the solutions well:

For example, for one of the problems (students choose the problem to optimize) it allows them to validate both the solution and evaluate the solution, so they can check that their evaluation is correct.

For example, for one of the problems:

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema

Introduzca un dataset:

Introduzca la solución de tamaño 8 con números entre 0.0 y 1.0 separados mediante comas:

solution to test:

Error in solution

"Error, tamaño no correcto, faltan 1"

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema

Introduzca un dataset:

Introduzca la solución de tamaño 8 con números entre 0.0 y 1.0 separados mediante comas:

solution to test:

Partición	Train	Test	Reducción	Fitness
	[en %]	[en %]	[en %]	
1	68.24	62.99	12.5	52.89
2	67.43	66.88	12.5	56.01
3	69.71	63.64	12.5	53.41
4	62.87	68.18	12.5	57.05
5	69.16	69.08	12.5	57.76

For another:

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema

Introduzca un fichero de entrada:

Introduzca la solución mediante una permutación de 1 a 22 separados mediante comas:

solution to test:

Error in solution

"Valor 8 repetido"

Validando soluciones

Esta página permite validar soluciones para los problemas de QAP y APC pedidos en la asignatura de Metaheurísticas de la Universidad de Granada, curso 2022/23.

Elija un problema

Introduzca un fichero de entrada:

Introduzca la solución mediante una permutación de 1 a 22 separados mediante comas:

solution to test:

Coste: 12220

Error: 6064

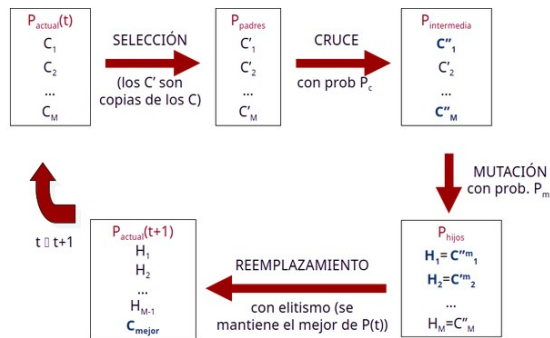
Fitness (Ratio): 0.9850552306692657

The last one:

Algoritmo Genético Generacional

El esquema se visualiza a continuación:

Mientras no se cumpla el criterio de parada, es decir, mientras no se haya generado un número máximo de evaluaciones, se aplican sucesivamente las siguientes fases:



- Selección: Se seleccionan soluciones de la población original para constituir una población de igual tamaño. Se seleccionan de forma aleatoriamente, pero normalmente con mayor tendencia a escoger las que presenten mayor fitness.
- Cruce: Se cruzan pares de soluciones para crear nuevas soluciones, reemplazando a las originales. Dado que tras la selección las soluciones están desordenadas, se pueden combinar por pares (la 1 con la 2ª, la 3ª con la 4ª, ...). No todas son cruzadas, solo el P_c % de pares.
- Mutación: Con una cierta probabilidad P_{mut} muy baja, se mutan las nuevas soluciones, mediante un cambio muy pequeño.
- Reemplazo: Las nuevas soluciones, la nueva población, reemplaza a la población anterior. Para evitar perder la mejor solución anterior, se copia en la nueva si no está.

```

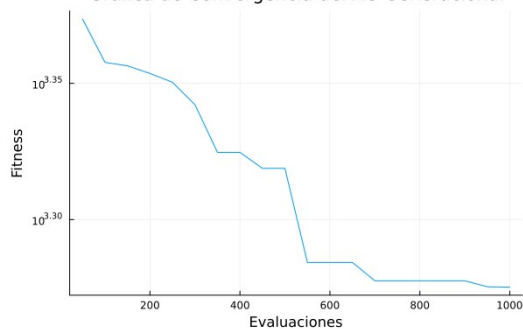
AGG (generic function with 1 method)
function AGG(dimension, popsize, pcross, pmut; maxevals=50_000,
historico=Tuple{Int,AbstractVector{<:Integer>,Float64}}[], copia_historico=false)
    pop = zeros{Int, popsize, dimension}
    # Crea la nueva población
    newpop = copy(pop)
    # Inicializo el vector
    for i in 1:popsize
        pop[i,:] = new_solution(dimension)
    end
    # Vector de fitness
    fitness_pop = [fitness(sol) for sol in eachsol(pop)]
    mejor = argmin(fitness_pop)
    # Inicio el número de evaluaciones
    evals = popsize
    if copia_historico
        push!(historico, (evals, pop[argmin(fitness_pop), :],
        obten_mejor_fitness(fitness_pop)))
    end
    while evals < maxevals
        # Aplico selección por torneo
        for i in indices(newpop)
            # Copio el ganador del torneo
            posi = torneo_binario(fitness_pop)
            newpop[i, :] = pop[posi, :]
        end
        # Empleo a cruzar de forma consecutiva
        total_cruzar = (popsize*pcross)
        i = 1
        while i < total_cruzar
            sol1 = newpop[i, :]
            sol2 = newpop[i+1, :]
            newsol1, newsol2 = crossover_0X(sol1, sol2)
            newpop[i, :] = newsol1
            newpop[i+1, :] = newsol2
            i += 2
        end
        # Mutación
        total_mutar = popsize + 10
        for i in enumerate(total_mutar)
            posi = posicion_aleatoria(newpop)
            newpop[posi, :] = mutation(newpop[posi, :])
        end
    end
  
```

Evaluaciones AGG:

Resultados AGG: 5.00E+01 evals, distancia 2363.8



Gráfica de Convergencia del AG Generacional

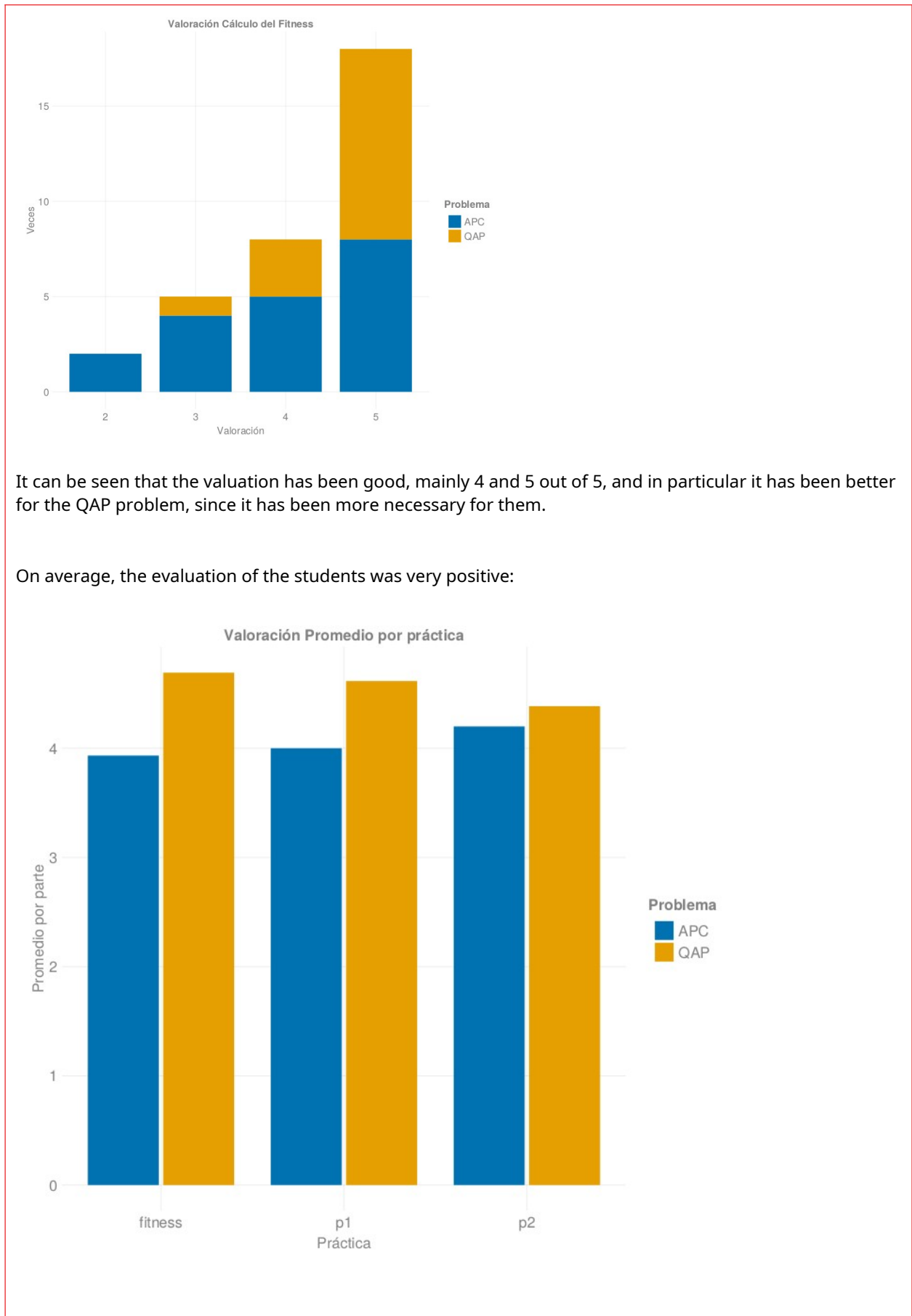


AG Generacional Vs AG Estacionario

Vamos a comparar los resultados de ambos para el problema **eil76.tsp** con **1000** evaluaciones.

	Algoritmo	Mejor Fitness	Tiempo
1	"Generacional"	1884.57	0.711568
2	"Estacionario"	1620.25	0.320271

The reception of this component has been very good, I indicate the results of the survey on this part:



Part 1:

This part will show how the algorithms seen can be used to solve a classic problem seen in class, The Traveling Salesman.

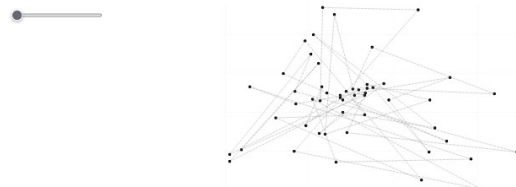
El problema del Viajante de Comercio

El problema del Viajante de Comercio (Traveling Salesman Problem, TSP) es un problema conocido de la literatura, en el que existe un viajante que debe de recorrer una serie de ciudades y desea encontrar la mejor ruta para ello. Por tanto, el problema consiste en:

- Encontrar la ruta más corta entre todas las ciudades.
- Pasar por cada ciudad únicamente una vez.
- Volver a la ciudad origen.

El objetivo es encontrar la asignación S que minimiza:

$$\sum_{i=0}^{N-1} Distancia(S_i, S_{i+1}) + Distancia(S_{N-1}, S_0)$$



Para abordar el problema usaremos un conjunto de ciudades de TSP sacadas del [TSPLib](#)

Escoge un problema:

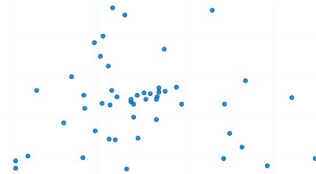
read_TSP (generic function with 1 method)

Vamos a leer sus datos (suponiendo que existe la función `read_TSP` que devuelve el tamaño, las posiciones (para pintar visualmente) y las distancias entre ellas (calculadas como las distancias euclídeas a partir de sus posiciones):

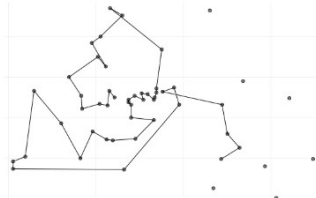
```
N, Position_cities, Distances_cities = read_TSP(name)
```

Vamos a visualizar el problema:

52 locations in Berlin (Groetschel)



52 locations in Berlin (Groetschel)



saved animation to /home/daniel/nh223/greedy.gif

Algoritmo de Búsqueda Local

En este caso vamos a aplicar el modelo de Búsqueda Local visto en clase, en particular el enfoque primero mejor.

Primero vamos a crear una solución aleatoria, y luego aplicamos un operador que cambie la solución un poco, la comparamos y nos quedamos la mejor de ambas, y luego se vuelve a aplicar lo mismo, hasta alcanzar un número de soluciones.

Definimos la función que muta:

mutate (generic function with 1 method)

```
function mutate(solution)
    result = copy(solution)
    n = length(solution)
    pos1 = rand(1:n)
    pos2 = rand(1:n)

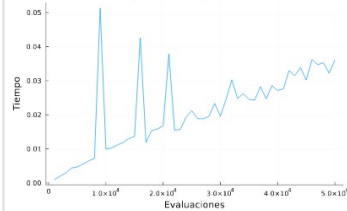
    # Comprueba que sea diferente de la primera
    while pos2 == pos1
        pos2 = rand(1:n)
    end

    # Intercambia posiciones
    result[pos1] = solution[pos2]
    result[pos2] = solution[pos1]
    return result
end
```

Midiendo el rendimiento

Ahora vamos a medir cómo de rápido es el algoritmo. Para ello vamos a probar los tiempos con distinto número máximo de evaluaciones (maxevals) y ver cómo funciona.

Tiempo con Búsqueda Aleatoria



```
begin
    # Recorro desde 1000 hasta 50000
    evals_time = collect(1:1000:50_000)
    local time_alg = Float64[]

    for evals in evals_time
        time = @elapsed optimize_random(Distances_cities, evals)
        push!(time_alg, time)
    end

    plot(evals_time, time_alg, legend=false, title="Tiempo con Búsqueda Aleatoria", xlabel="Evaluaciones", ylabel="Tiempo")
end
```

Tabla comparativa

Vamos a medir los algoritmos con distintas evaluaciones y medir los tiempos.

Resultado con 34335 evaluaciones

Algoritmo	Tiempo	Mejor Fit.
Greedy	2.584e-04	10200.895
Random	2.990e-02	25890.361
Búsqueda Local	2.178e-02	10524.587

```
begin
    function times(distances)
        df = DataFrame{name=String[], tiempos=Float32[], best=Float32[]}
        seed!(100)

        time_greedy = @elapsed _, fit_greedy = greedy(distances)
        push!(df, (name="Greedy", tiempos=time_greedy, best=fit_greedy))
        seed!(100)

        time_random = @elapsed _, fit_random = optimize_random(distances, evals_comp)
        push!(df, (name="Random", tiempos=time_random, best=fit_random))
        seed!(100)

        time_bl = @elapsed _, fit_bl = busquedaLocal(distances, evals_comp)
        push!(df, (name="Búsqueda Local", tiempos=time_bl, best=fit_bl))

        PrettyTables.pretty_table(df, headers=["Algoritmo", "Tiempo", "Mejor Fit."], title="Resultado con $(evals_comp) evaluaciones", formatters = (PrettyTables.ft_printf("%3e", [2]), PrettyTables.ft_printf("%2.3f", [3])))
    end

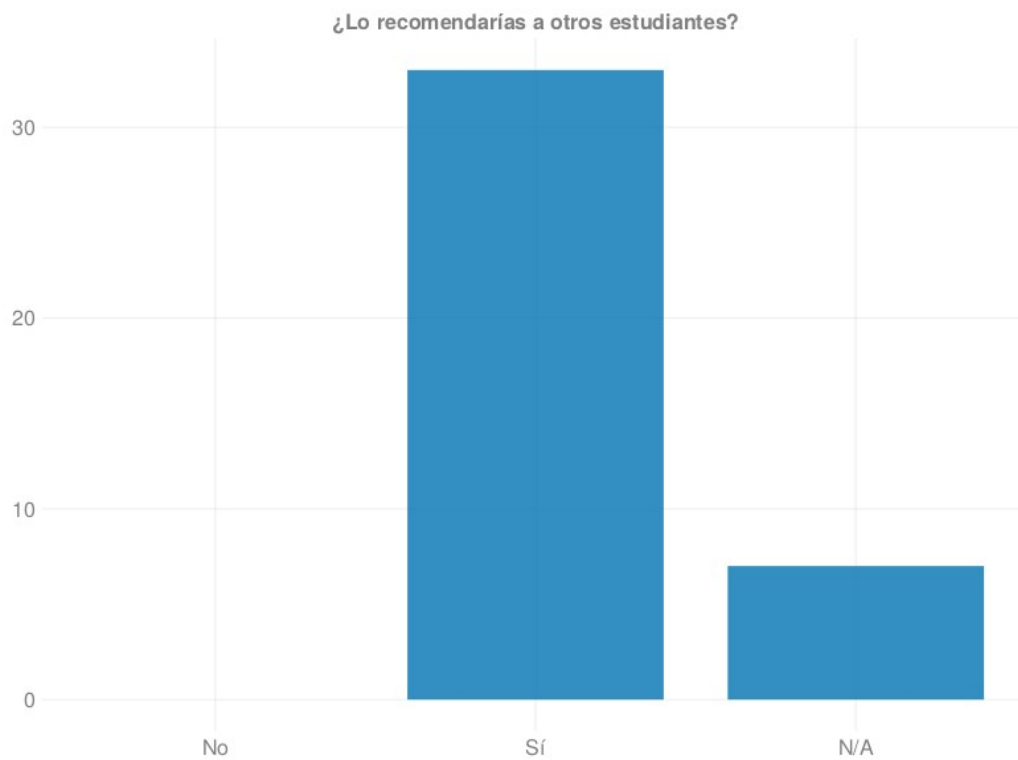
    times(@distances_cities)
end
```

► Preguntas

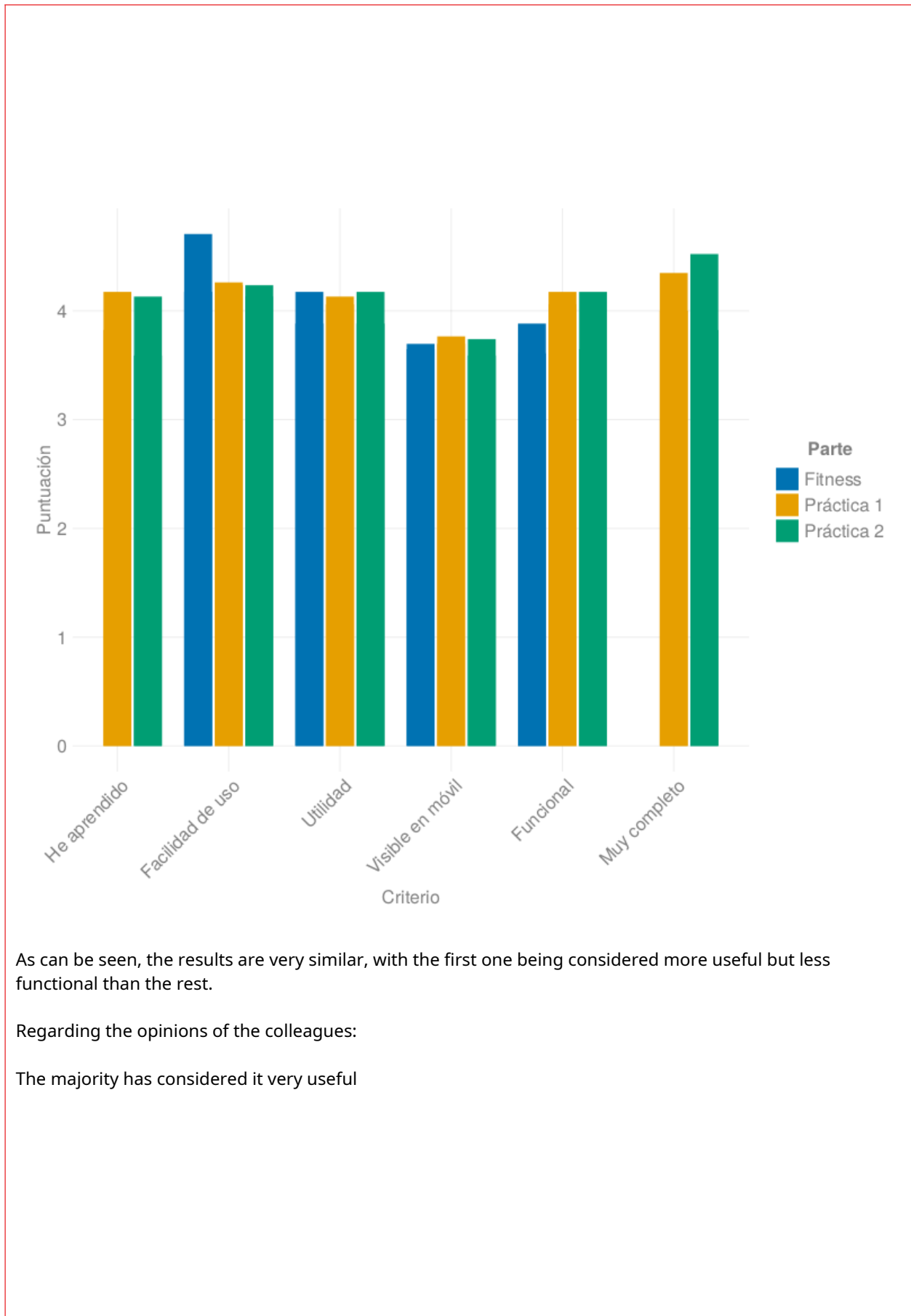
It can be seen that not only results are displayed (dynamic, the distribution of cities can change) but also code in a high-level language (Julia, similar to Python) and even questions to the students that the system checks if they are correct or not.

Part 3:

In this third part two new algorithms are shown, which are requested in practice 2



In summary, students have been asked to rate different attributes of each part, and the ratings have been very good:

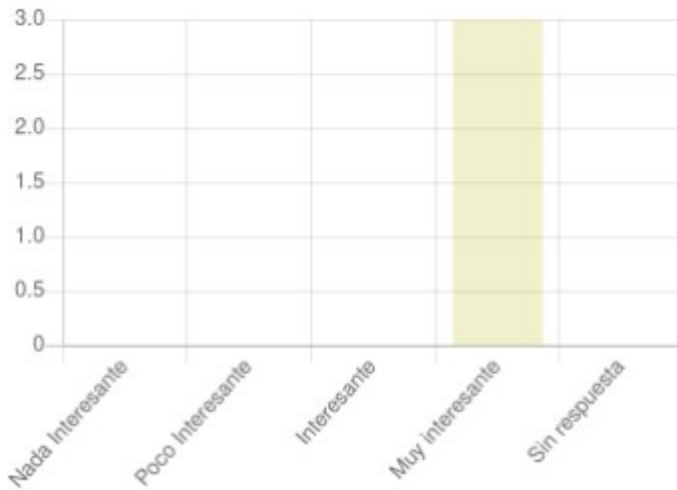


As can be seen, the results are very similar, with the first one being considered more useful but less functional than the rest.

Regarding the opinions of the colleagues:

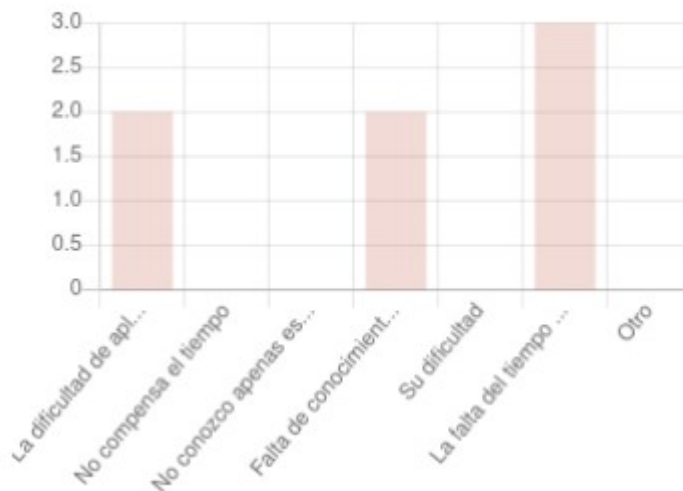
The majority has considered it very useful

¿Cómo valoras la iniciativa de implicar al estudiantado con herramientas interactivas para la docencia?



They are very much in agreement with this type of tool, but although they consider it very interesting to be applied, there are different reasons why they do not do so:

¿Cuáles son los principales problemas para el uso de este tipo de herramientas?



In other words, the main reason for not using them is considered to be the lack of teacher time, followed by the difficulty of application (which requires a lot of time) and lack of knowledge on the part of the teacher to use them.

E. Difusión y aplicación del proyecto a otras áreas de conocimiento y universidades

Tal y como se ha indicado en el apartado D, el software resultado del presente proyecto de innovación docente está libremente disponible para ser usado por todo docente interesado.

Aparte, se ha difundido personalmente a un conjunto del profesorado que trabajan en el mismo área, y se

les ha pasado un cuestionario, cuyos resultados se adjuntan.

Una vez terminado el proyecto, se espera poder difundirlo en congresos temáticos, con lo que se difundiría a docentes de otras universidades.

También se hará difusión como herramienta complementaria en una charla divulgativa de la Noche de los Investigadores.

Dissemination and application of the project to other areas of knowledge and universities (In English)

As indicated in section D, the software resulting from this teaching innovation project is freely available for use by all interested teachers.

In addition, it has been personally disseminated to a group of teachers working in the same area, and a questionnaire has been sent to them, the results of which are attached.

Once the project is completed, it is expected to be disseminated in thematic congresses, which would be disseminated to teachers from other universities.

It will also be disseminated as a complementary tool in an informative talk at the "Noche de los Investigadores".

F. Estudio de las necesidades para incorporación a la docencia habitual

Tras una pequeña formación de uso al profesorado se podrá utilizar tanto en clases de teoría como de práctica de la asignatura. Se ha planteado el proyecto como una herramienta a utilizar directamente en la docencia habitual.

G. Puntos fuertes, las dificultades y posibles opciones de mejora

Puntos fuertes:

- Software interactivo.
- Elementos visuales con controles interactivos.
- Fácil instalación, configurado para ser utilizado fácilmente con sistemas virtuales (contenedores).
- Fácil de extender a nuevos algoritmos.

Dificultades:

- El servidor debe de tener mayor memoria, en otros proyectos se plantea pedir dinero para alquilar servidor para ello.
- Para extender el software hay que aprender la tecnología sobre la que se ha construido el software.

Posibles opciones de mejora:

- Versión multi-idioma para tener versión en inglés para estudiantes Erasmus.
- Añadir más algoritmos.
- Mejorar el software en base al resultado de las distintas encuestas.

