

Diseño y desarrollo de una herramienta de estimación de recursos

Carlos Navarro Aguila

Resumen— En este proyecto se propone una herramienta de estimación de recursos basada en un algoritmo de búsqueda local. El objetivo de este proyecto se centra en la creación de una herramienta que permita determinar una mejor gestión de los recursos de los que dispone una empresa para planificar sus proyectos con el menor coste posible. Se ha creado una función de coste para modelizar el problema y poder mostrar un óptimo. Además, se han realizado una série de pruebas para poder determinar el buen funcionamiento del algoritmo.

Palabras clave—Búsqueda Tabú, función de coste, habilidades, estado, óptimo, mínimo local, mínimo global.

Abstract— In this paper, a resource estimation tool based on a local search algorithm is proposed. The main objective of this project focuses on the creation of a tool that allows to determine a better management of the available resources to a company to plan their projects as lowest as possible. A cost function has been created to modelize the problema and show an optimum value to solve it. In addition, a series of tests have been performed to determine the proper functioning of the algorithm.

Index Terms—Tabu search, cost function, skills, state, optimum, local minimum, global minimum.



1 INTRODUCCIÓN

En este artículo se describe el Proyecto Final de Grado desglosando en su totalidad el desarrollo del proyecto, su funcionalidad, elementos que lo componen, la metodología utilizada en su desarrollo y los resultados obtenidos.

Son muchas las empresas que tienen que terminar haciendo reajustes de plantilla o recortes de presupuesto porque los proyectos le generan más gastos de los que creían. Un gasto importante viene dado por la mala planificación que se hace de los proyectos. No asignar a la persona adecuada provoca que el tiempo esperado para realizar la tarea aumente considerablemente.

La empresa en la que realizo mis prácticas busca crear una herramienta para poder estimar los recursos que dispone. Hasta ahora han utilizado herramientas que no les permitía vincular proyectos entre sí. Cada Project Leader podía ver sus recursos disponibles pero no los que estuvieran en otros proyectos.

En este proyecto vamos a centrarnos, no sólo en la creación de una herramienta que facilite este vínculo, sino que, además, optimice la planificación de los recursos en dichos proyectos.

Cada proyecto tiene unas tareas asignadas y cada trabajador varias habilidades que se relacionan con estas tareas. Los Project Leaders asignan un valor a cada habilidad que tengan los trabajadores basándose en su experiencia. Según este valor, asignar un trabajador a una tarea u otra puede hacer que el proyecto se demore y esto suponga pérdidas económicas para la empresa.

Los datos utilizados en el proyecto han sido elegidos para simular un entorno real.

La aplicación ha sido desarrollada de la forma más user-friendly posible, utilizando frameworks como Django y Semantic ui.

A continuación, se detallarán las secciones de este trabajo. En el apartado **Estado del arte** se muestra un resumen de proyectos similares que se han consultado de cara a reforzar el planteamiento. En el apartado **Metodología**, se explicará el algoritmo utilizado y el desarrollo de la interfaz. En la sección **Resultados**, se muestra y analiza el rendimiento del algoritmo usando diferentes configuraciones. Por último, en **Conclusiones**, se hará un análisis global del proyecto.

2 ESTADO DEL ARTE

Este apartado expone el estado del arte del problema MSRCPS (Multi-Skill Resource Constrained Project Scheduling Problem).

- E-mail de contacte: carlos.navarroa@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Robert Benavente Vidal (DCC)
- Curs 2017/18

2.1 MSRCPSP

En inteligencia artificial, y en otras áreas de la computación, muchos problemas pueden ser vistos como problemas de satisfacción de restricciones. Estos problemas están compuestos por:

- Conjunto finito de variables
- Conjunto de Dominios de cada variable
- Conjunto de restricciones

Éstos serían elementos básicos de un problema de satisfacción de restricciones (CSP). Sudoku, 8-reinas, serían algunos ejemplos.

Un problema de planificación de recursos en proyectos con restricciones (RCPSP)[1] sería una ampliación del problema anterior. El proyecto tiene unas tareas que han de ser asignadas a X recursos y no importa que recurso realice esta tarea. Cumplen ciertas restricciones como que un recurso no puede estar asignado a dos tareas en el mismo período, las tareas tienen una determinada duración, etc.

Cuando a esos recursos le asignamos un conjunto de habilidades, el problema se convierte en un MSRCPSP[2]. Aquí no sólo hemos de tener en cuenta que el trabajador no puede ser asignado a más de una tarea a la vez, sino que realmente es capaz de realizar esa tarea. El objetivo es decidir para cada tarea qué recursos deben ser asignados y cuándo.

Un ejemplo de este tipo de problema, lo encontramos en el trabajo realizado por Delgoshaei, Khairol y Hang[3]. Resuelven este problema utilizando un algoritmo genético.

3 OBJETIVOS

3.1 ¿En qué consiste la aplicación?

El objetivo principal de este proyecto es dar una solución a un problema de planificación de recursos en función de sus habilidades. Para ello se pretende desarrollar e implementar un algoritmo de búsqueda local. El algoritmo utilizado para resolver el problema será Tabú Search[4][5].

Para llevar a cabo el proyecto desarrollaremos los siguientes subobjetivos:

- Implementar una función de coste que nos relacione el nivel de habilidad que tiene cada trabajador.
- Implementar la Búsqueda Tabú para encontrar el estado óptimo.

Además, en este proyecto existe un objetivo secundario:

- Desarrollar e implementar una interfaz web para visualizar la planificación de los proyectos.
- Realizar experimentos para ver que cambios suponen cuando se alteran ciertos valores en el algoritmo o en la función de coste.

A largo plazo puede aportar datos de interés para futuras planificaciones y, además, reducir los gastos de la empresa.

Con todos los datos que la aplicación almacene, podemos analizarlos y obtener métricas. Gracias a estos datos estadísticos la empresa puede tener constancia de cómo se ve afectado el presupuesto u si un recurso le sale rentable o no, incluso si necesitan más o menos recursos.

3.3 Objetivos como trabajo de fin de grado

Por lo que se refiere al protecto de Trabajo Final de Grado, se establecen una serie de objetivos para darle sentido a la elaboración del proyecto. Un objetivo es el de poner en práctica lo aprendido en el grado y, más especialmente, en la mención de Computación. Se detalla la lista de conocimientos que se han puesto en práctica en la siguiente tabla:

Asignatura	Conocimiento
Tecnologies de Desenvolupament per a Internet i Web	Utilizar lenguajes de programación web (HTML, CSS, JS).
Laboratori de programació	Utilizar lenguajes de programación (Python).
Coneixement, raonament i incertesa	Uso de algoritmos de búsqueda local (Tabu Search).
Gestió de Projectos	Elaborar documentación
Anàlisi i disseny d'algoritmes	Seleccionar la mejor técnica de programación para resolver problemas complejos.
Aprentatge Computacional	Generar propuestas innovadores en el ámbito profesional.

Tabla 2. Relación entre asignatura y conocimiento puesto en práctica.

4 DESCRIPCIÓN DEL PROBLEMA

4.1 Descripción

El foco de este proyecto será centrarse en optimizar el coste total de los proyectos. El planteamiento del problema viene dado por los siguientes factores:

- El trabajador puede estar únicamente en un proyecto.

- El proyecto tiene unas tareas que cumplir (Python, Django, RobotFramework, etc.).
- La tarea la realizará el trabajador que tenga esa habilidad.
- El trabajador puede realizar únicamente una tarea.

Las habilidades son atributos que añade el mánager del departamento y pueden estar valoradas de 1 a 5, dónde 1 es el nivel más bajo de conocimiento y 5 el nivel más alto.

4.2 Modelo matemático

Para resolverlo, primero plantearemos el modelo matemático del problema. Por el momento, haremos un planteamiento básico. Más adelante, se pueden añadir mejoras cómo la retención a mover un trabajador u el coste que supone aprender una habilidad.

- C_p = Coste del proyecto p
- P = número de proyectos
- h_{ip} = horas de la tarea i del proyecto p
- w_{ip} = ID trabajador asignado a la tarea i del proyecto p
- $sk_i(\cdot)$ = nivel habilidad trabajador asignado a la tarea i
- $s(\cdot)$ = sueldo del trabajador
- M ≡ matriz de asignación de los recursos
- M_{ij} ≡ 1 si la tarea i se le asigna el trabajador j , sinó 0
- t_p = número de tareas del proyecto p
- N = porcentaje de penalización
- d_{ip} = max días que debe durar la tarea i del proyecto p
- $dias(\cdot)$ = días que tarda en hacer la tarea un trabajador
- Q_p = valor de penalización del proyecto p

Dadas estas variables, la función objetivo que queremos minimizar es la siguiente:

$$Min \sum_{p=1}^P C_p \tag{1}$$

$$C_p = \sum_{i=1}^{t_p} m + n \tag{2}$$

$$m = \left(h_{ip} + \left(N \cdot \left(3 - sk_i(w_{ip}) \right) \cdot h_{ip} \right) \right) \cdot s(w_{ip}) \tag{3}$$

$$n = \max \left(\left(dias(w_{ip}) - d_{ip} \right), 0 \right) \cdot Q \tag{4}$$

$$w_{ip} = \max_j M_{ij} \tag{5}$$

La función objetivo (1) calcula el coste de todos los proyectos. Función que queremos minimizar para encontrar la planificación que supone un menor coste. El coste de cada proyecto (2) viene dado por la suma de la función (3) y (4). La función (3) calcula el coste económico generado al asignar a un trabajador. Ese coste viene dado por el nivel de la habilidad que tiene el trabajador. Si el nivel de la habilidad es 3, únicamente multiplica el número de horas de la tarea por el coste que supone ese trabajador. Si el nivel de la habilidad es mayor que 3, reduce en N% el número de horas que durará la tarea. Sin embargo, si el nivel de la habilidad es inferior a 3, supondrá un aumento del N% el número de horas. De esta forma se consigue

que el nivel de la habilidad haga variar el coste asignándole un trabajador u otro. La función (4) aplicará una penalización Q_p (cada proyecto tendrá penalización propia) si el trabajador excede el número de días máximos que tiene para realizar la tarea.

4.3 Búsqueda Tabú

La Búsqueda Tabú es un algoritmo metaheurístico que usa búsqueda agresiva del óptimo del problema. Esto quiere decir que evita que la búsqueda quede atrapada en un óptimo local que no sea global.

El principio de la Búsqueda Tabú podría resumirse como:

“Es mejor una mala decisión basada en información que una buena decisión al azar, ya que, en un sistema que emplea memoria, una mala elección basada en una estrategia proporcionará claves útiles para continuar la búsqueda. Una buena elección fruto del azar no proporcionará ninguna información para posteriores acciones.” [8]

La Búsqueda Tabú es un proceso iterativo que comienza en una solución X. En cada iteración, a partir de la solución actual, el proceso pasa a la mejor solución del vecindario. Se aceptará el movimiento incluso aunque este vecino se peor que la solución actual.

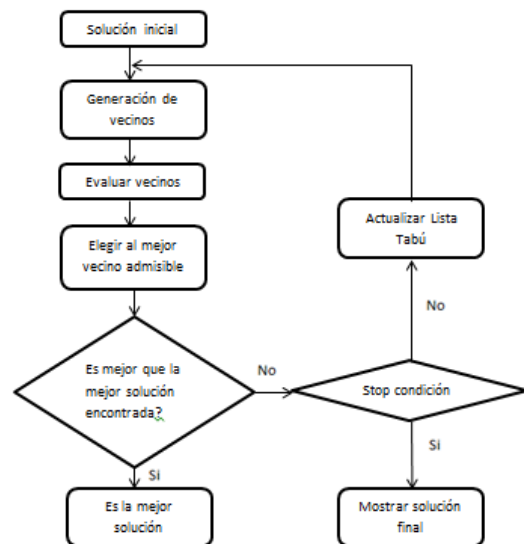


Figura1. Pseudocódigo Búsqueda Tabú 1

La Búsqueda Tabú utiliza memoria a corto plazo y memoria a largo plazo. Para la memoria a corto plazo, utiliza la Lista Tabú. En esta lista se guardan las soluciones previamente visitadas en un pasado cercano. De esta forma, se evita que la búsqueda se cicle. En la memoria a largo plazo, se guardan las soluciones descartadas, para evitar volver a elegir un vecino que no interesaba.

5 METODOLOGÍA

5.1 Metodología base: 'Cascada'

Después de una revisión a las diferentes metodologías posibles, se ha decidido que la más adecuada para este tipo de proyecto es la metodología en 'Cascada'. Esta metodología de trabajo establece una serie de fases por las que el proyecto debe fluir siguiendo una planificación temporal. En el caso de este proyecto, la planificación ha sido la siguiente:

Fase	Tiempo
Análisis de algoritmos	Desde: 1 de Marzo Hasta: 10 de Marzo
Implementación de la interfaz web	Desde: 11 de Marzo Hasta: 26 de Marzo
Montaje de la base de datos	Desde: 11 de Marzo Hasta: 26 de Marzo
Creación de funciones de intercambio de filas en matrices	Desde: 26 de Marzo Hasta: 22 de Abril
Implementación de la primera versión del algoritmo	Desde: 22 de Abril Hasta: 27 de Mayo
Añadir mejoras al algoritmo	Desde: 27 de Mayo Hasta: 17 de Junio
Pruebas y resultados	Desde: 17 de Junio Hasta: 02 de Julio

Tabla 3. Planificación del proyecto.

El diagrama de Gantt con las subtarefas se encuentra en el apéndice A1.

5.2 Aplicación del algoritmo

Cada proyecto tiene t_p tareas programadas de una determinada dedicación en horas y un límite de finalización expresado en días.

Hemos planteado el problema como se detalla a continuación. Utilizaremos una matriz para representar cada estado. Las filas de la matriz serán cada una de las tareas de cada proyecto, las columnas serán cada uno de los trabajadores que podrán o no hacer las tareas. La matriz

tendrá el valor '1' para asignarle esa tarea a un trabajador y '0' en el resto de casillas de la fila y la columna.

Se plantea un ejemplo para ayudar al entendimiento de la metodología aplicada. Siguiendo los pasos del pseudocódigo en el apartado 2.2.1, primero de todo, cogemos un estado inicial como solución.

Cada vez que se solicita la planificación se genera aleatoriamente una solución inicial diferente. De esta forma, evitamos que siempre se genere el mismo recorrido para llegar a la solución.

Paso 1. Dada una solución inicial

La inicialización consiste en asignar un trabajador aleatorio que pueda realizar esa tarea de entre todos los trabajadores que pueden realizarla. También creamos la Lista Tabú. En esta lista guardaremos los movimientos que efectuamos al generar un nuevo vecino y lo hacemos tabú durante 5 movimientos. Estos movimientos son las filas intercambiadas en la generación de vecinos.

Calculamos el coste y lo guardamos como la mejor solución encontrada.

Paso 2. Mientras no se verifique un cierto criterio de finalización

Este paso supone un bucle. El bucle tiene una única condición de parada. Ésta consiste en número máximo de iteraciones. Para el número de iteraciones se han elegido varios valores para estudiar que se verán en los experimentos realizados. Cada vez que se encuentra una mejor solución final, reiniciamos a 0 el número de iteraciones. En caso contrario, aumentamos en 1 el número de iteraciones. En el bucle se realizan los siguientes pasos:

Paso 3. Seleccionar una solución X' próxima a X , que no sea tabú o que satisfaga un cierto criterio de aspiración con mínimo coste en el vecindario de X

Generamos los vecinos de X . Cómo funciona la generación de vecinos se detalla en el siguiente apartado. De los vecinos generados elegimos aquellos que no estén en la lista tabú y calculamos el coste. Nos quedamos con el vecino que tenga mejor coste de entre todo el vecindario, aunque éste sea peor que le mejor solución encontrada. Si hubiera más de un vecino con la misma solución, elegimos uno de ellos de forma aleatoria.

Paso 4. Hacer $X=X'$ y actualizar la lista tabú y criterio de aspiración

Asignamos el mejor vecino como el estado actual (este estado pasará de X' a X en el Paso 1) y añadimos el movimiento a la lista tabú. También ac-

tualizamos la lista tabú restando uno a los movimientos tabú y si el valor es '0', lo eliminamos de la lista.

Comparamos también el mejor de los vecinos con la mejor solución encontrada, y si el coste es menor, hacemos que éste sea nuestra nueva mejor solución encontrada y reiniciamos a 0, también, el número de iteraciones. Así, si no encuentra una solución mejor dentro de unas iteraciones definidas previamente, detenemos la búsqueda.

Paso 5. Retornar la mejor solución encontrada

Cuando el bucle termina, devolvemos el estado que ha sido la mejor solución encontrada y su coste.

5.3 Generación de vecinos

El proceso elegido se basa en la aleatoriedad. Generamos dos números aleatorios entre 0 y el número de filas de la matriz, hasta que éstos sean diferentes. Estos números nos indican que filas hemos de intercambiar. Comprobamos también, que los trabajadores puedan hacer la tarea que estaba haciendo el otro. En caso de que no pueda, volvemos a generar uno de los números aleatorios, para hacer el intercambio con otra fila. Comprobamos, también, que el vecino elegido no haya sido descartado en otras iteraciones.

El intercambio de filas lo hacemos siempre partiendo del estado actual. De esta forma, los vecinos generados se diferencian del estado actual en un único intercambio de filas.

Suponer que el estado inicial es la siguiente matriz:

	Trab1	Trab2	Trab3	Trab4
P1-T1	0	1	0	0
P1-T2	1	0	0	0
P2-T1	0	0	0	1
P2-T2	0	0	1	0

Algunos intercambios serían:

	Trab1	Trab2	Trab3	Trab4
P1-T1	1	0	0	0
P1-T2	0	1	0	0
P2-T1	0	0	0	1
P2-T2	0	0	1	0

	Trab1	Trab2	Trab3	Trab4
P1-T1	0	1	0	0
P1-T2	0	0	0	1
P2-T1	1	0	0	0
P2-T2	0	0	1	0

6 RESULTADOS

Una vez implementada la primera versión del algoritmo, detallamos el esquema y los resultados obtenidos.

Existen dos proyectos con las siguientes tareas a realizar:

Proyecto 1

Tarea	Horas	Dias máximos
RQM	300	40
GPS	358	46
RobotFramework	100	13
Django	300	40
Jenkins	50	7
Bluetooth	60	9

Proyecto 2

Tarea	Horas	Dias máximos
Probe	222	30
GPS	300	40
CAN	120	17
Django	550	70
Jenkins	250	35
Backup	80	12

En una de las ejecuciones realizadas, la matriz inicial queda de la siguiente forma. Se lista por cada proyecto que trabajador (iniciales) realiza cada tarea.

Proyecto 1

- RQM → XA
- Django → CN
- GPS → IP
- Jenkins → JE
- RobotFramework → VL
- Bluetooth → FR

Proyecto 2

- Django → JG
- Jenkins → AB
- GPS → DQ
- CAN → MC
- Backup → AP
- Probe → NC

Con un coste de 383680€.

Después del algoritmo obtenemos la siguiente matriz:

Proyecto 1

- RQM → ~~XA~~ → CN
- Django → ~~CN~~ → VL
- GPS → ~~IP~~ → NC
- Jenkins → ~~JE~~ → MC
- RobotFramework → ~~VL~~ → AP

Bluetooth → FR → IP

Proyecto 2

Django → JG → XA

Jenkins → AB → JE

GPS → DQ

CAN → MC → AB

Backup → AP → JG

Probe → NC → FR

Con un coste de 114393,5€.

Se puede observar que el algoritmo realiza varios cambios respecto a la matriz inicial.

Hay que tener en cuenta que en esta primera versión sólo tiene en cuenta movimientos entre usuarios elegidos en la inicialización.

Destacar que la generación de vecinos elegida puede hacer que el algoritmo sea más lento u incluso el número de iteraciones elegidas para finalizar la búsqueda.

En el apéndice A4 se muestran capturas de pantalla de la matriz inicial, para observar que no siempre es la misma y de la matriz final.

6.1 Número de iteraciones

En este apartado estudiaremos qué pasa con la solución final cuando se modifican las iteraciones máximas en la generación de vecinos o como criterio de finalización de la búsqueda Tabú.

6.1.1 Iteraciones en la generación de vecinos

Elegimos diferentes números de iteraciones en la generación de vecinos. Esto implicará que el número de vecinos generados variará. Con este experimento queremos determinar si la generación de vecinos supone una gran ventaja para alcanzar el mínimo global.

Por cada valor haremos 10 pruebas y nos guardaremos el resultado final y el tiempo de ejecución. De los 10 resultados finales sacaremos el resultado medio y lo mismo para los tiempos de ejecución. Para cada prueba se han elegido 120 iteraciones en la búsqueda local.

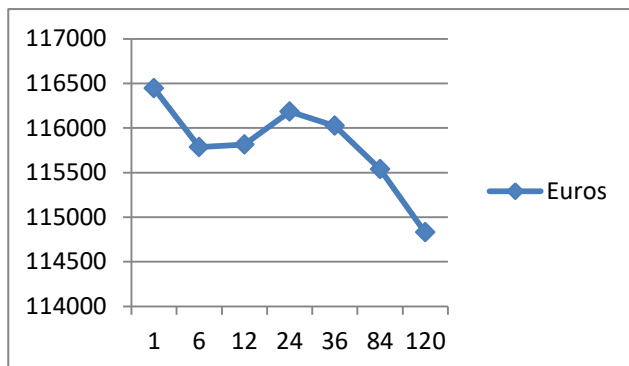


Figura2. Coste generación de vecinos

En la Figura2 podemos observar que el coste se acerca a valores cercanos al mínimo global cuando el número de iteraciones en la generación de vecinos aumenta. Al generar más vecinos aumentan las posibilidades de encontrar mejores planificaciones. Se ha podido observar también que el valor 114393.5 aparece en la mayoría de ocasiones y esto nos hace indicar que ese valor es un el mínimo global.

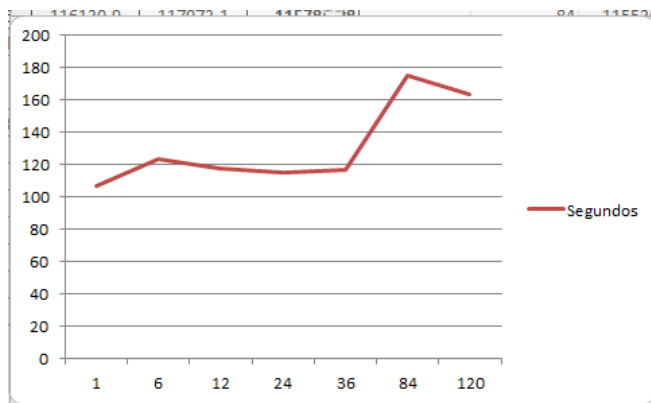


Figura3. Tiempo generación de vecinos

En la Figura3 se observa que al aumentar el número de iteraciones en la generación de vecinos aumenta el tiempo de ejecución del algoritmo. Ésto puede venir dado porque al poder elegir entre más vecinos, a las pocas iteraciones se encuentra una solución mejor a la mejor encontrada. - Ésto hace que el número de iteraciones para finalizar la búsqueda tabú se vaya reiniciando y se tarde más en finalizar el algoritmo.

6.1.2 Iteraciones en la búsqueda tabú

Elegimos diferentes números de iteraciones en el algoritmo principal de la búsqueda tabú. Con este experimento queremos determinar cómo afecta el criterio de finalización con un gran número de vecinos generados.

Por cada valor haremos 10 pruebas y nos guardaremos el resultado final y el tiempo de ejecución. De los 10 resultados finales sacaremos el resultado medio y lo mismo para los tiempos de ejecución. Para cada prueba se han elegido 120 iteraciones en la generación de vecinos.

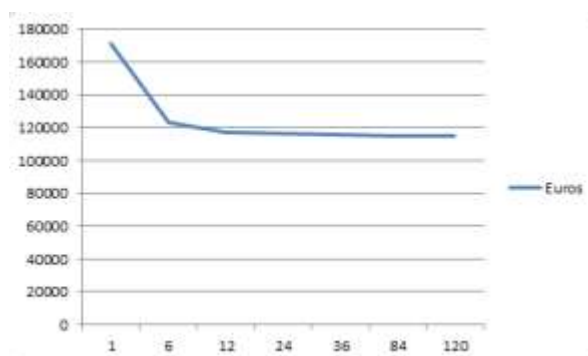


Figura4. Coste búsqueda tabú

En la Figura4 se observa cómo para valores altos nos acercamos a valores cercanos al mínimo global. En cambio, cuando sólo hacemos una iteración vemos que los valores se alejan bastante. Esto se debe a que la búsqueda tabú acepta valores peores al estado actual en la generación de vecinos y eso haría que se cumpliera el criterio de finalización cuando el número de iteraciones es 1.

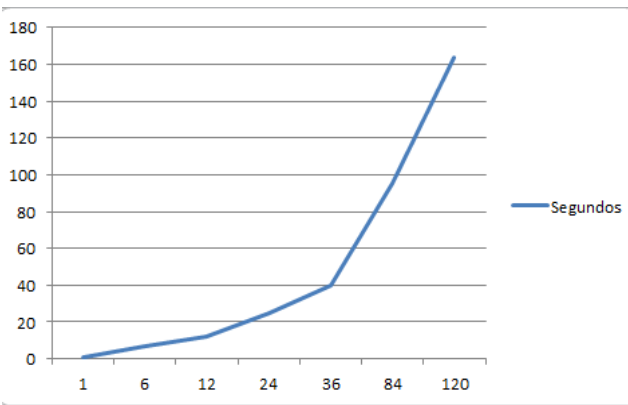


Figura5. Tiempo búsqueda tabú

En la Figura5 se observa como el tiempo aumenta conforme vaoms aumentando el número de iteraciones en la búsqueda tabú. Esto tiene mucho sentido ya que al retrasar el final del algoritmo podemos generar más vecinos que tengan un coste menor al que hemos encontrado.

Se ha observado también que para 120 iteraciones se repite en varias ocasiones el valor 114393.5, considerado como el mínimo global. El problema es que el tiempo aumenta considerablemente para 120 iteraciones.

7 INTERFAZ WEB

La web ha sido desarrollada usando Django y Semantic Ui. Django permite la creación de modelos para la creación de la base de datos. Estos modelos son *non-database-dependendant*.

El primer paso después de reunir la información necesaria ha sido el de diseñar la base de datos. Django te facilita mucho esta tarea porque únicamente has de tener muy claras las tablas que quieres que tenga la base de datos y crear las clases necesarias para el modelo[Apéndice A3].

Después de crear la base de datos, se ha de realizar un *mockup* para el diseño de la interfaz, reunirte con cliente y ver si es lo que quiere.

Una vez hechos estos dos pasos, empieza la búsqueda de plugins que puedan ser útiles (Datatables, Highcharts), la creación de las urls, los templates, los archivos javascript, css y las imágenes.

Para loguearse en la página se ha implementado un acce-

so usando el LDAP que utiliza Ficosa.

Se han creado varias secciones y vamos a detallar qué hace cada una de ellas:

Forecast

En esta sección podemos observar los recursos asignados a cada proyecto y además añadir nuevos recursos o editar los que ya estaban. Para poder reservar un recurso, éste debe haber sido añadido previamente al proyecto. La información se muestra en un mapa de calor usando *Highcharts*.



Figura6. Forecast web

Projects

Al entrar en esta sección y elegir un proyecto, veremos la información básica de ese proyecto: releases, PCR (Product Change Request), dos gráficas para mostrar el budget del proyecto y una tabla resumen de los recursos asignados. Para la tabla se usa *Datatables*.



Figura7. Project web

Users

Se muestra la información del usuario. Está restringida según los privilegios que tenga el usuario logueado. Con el máximo de permisos podrá ver: información básica, comentarios sobre el usuario, habilidades y nivel de éstas,

equipo o supervisor, y una tabla resumen de los proyectos en los que está asignado[Apéndice A2].

Management

Esta sección únicamente la puede ver el Administrador de la página web. En ella se pueden hacer las siguientes funciones:

- Asignar usuarios a departamentos
- Asignar roles a usuarios
- Resincronizar usuarios de LDAP
- Crear unidades de tiempo
- Asignar usuarios a un supervisor
- Asignar usuarios a proyectos
- Crear skills

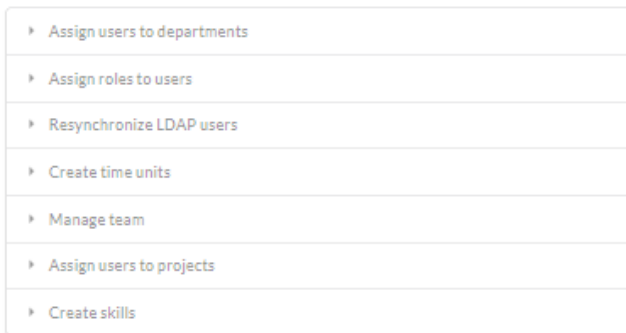


Figura8. Management web

Search

Aquí es dónde se muestra el funcionamiento del algoritmo implementado y se ve la tabla resumen. Via Ajax (ya que el proceso, como hemos visto, puede ser muy lento) se lanza el algoritmo y al acabar se muestra la planificación en tablas. Una tabla por cada proyecto. Se muestra también el valor de la planificación actual y se compara con el valor de la planificación calculada por el algoritmo.

Además, se permite una herramienta de búsqueda al usuario para poder buscar los usuarios que tienen ciertas habilidades con un cierto nivel.

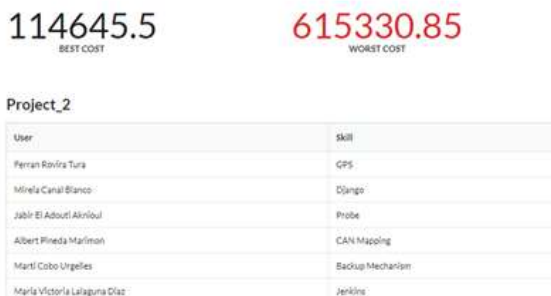


Figura9. Tabu Search web

8 CONCLUSIONES

En la recta final del proyecto, se puede dar por hecho que los objetivos planteados al inicio se han ido superando progresivamente, tanto a nivel técnico de la aplicación, como a nivel formativo del Trabajo Final de Grado. La elaboración de este proyecto ha requerido poner en práctica los conocimientos adquiridos durante la carrera, tanto de asignaturas troncales como de las de la mención de Computación.

El proyecto se ha centrado en resolver un problema de planificación utilizando un algoritmo de búsqueda local como es la Búsqueda Tabú.

Con los resultados obtenidos se puede decir que el algoritmo converge encontrando varias veces el que parece ser el mínimo global. Además, podemos concluir que un número elevado de iteraciones en la búsqueda tabú ayuda a que el algoritmo llegue al mínimo global.

Dificultades encontradas

Al principio del proyecto surgían algunas dudas en el planteamiento del problema, especialmente en cómo juntar todos los datos en una función de coste. Esto ha supuesto que la planificación se retrasara y se tuviera que aumentar la carga de trabajo en las últimas semanas.

El futuro de la aplicación

La aplicación puede ser de gran ayuda y aun quedan muchas cosas por pulir que permitirían dar datos más reales a las empresas.

Los puntos más importantes sería modificar la función de coste para que aplicara a otros detalles que no se han tenido en cuenta en este proyecto. Por ejemplo, se podría añadir la reticencia que tiene un Project Leader a mover uno de sus empleados, el coste que supone que un trabajador pase de un proyecto a otro, etc.

También se puede probar varias configuraciones diferentes para la generación de vecinos que no se basen en la aleatoriedad y ver sí, realizando los mismos experimentos, obtenemos mejores resultados.

AGRADECIMIENTOS

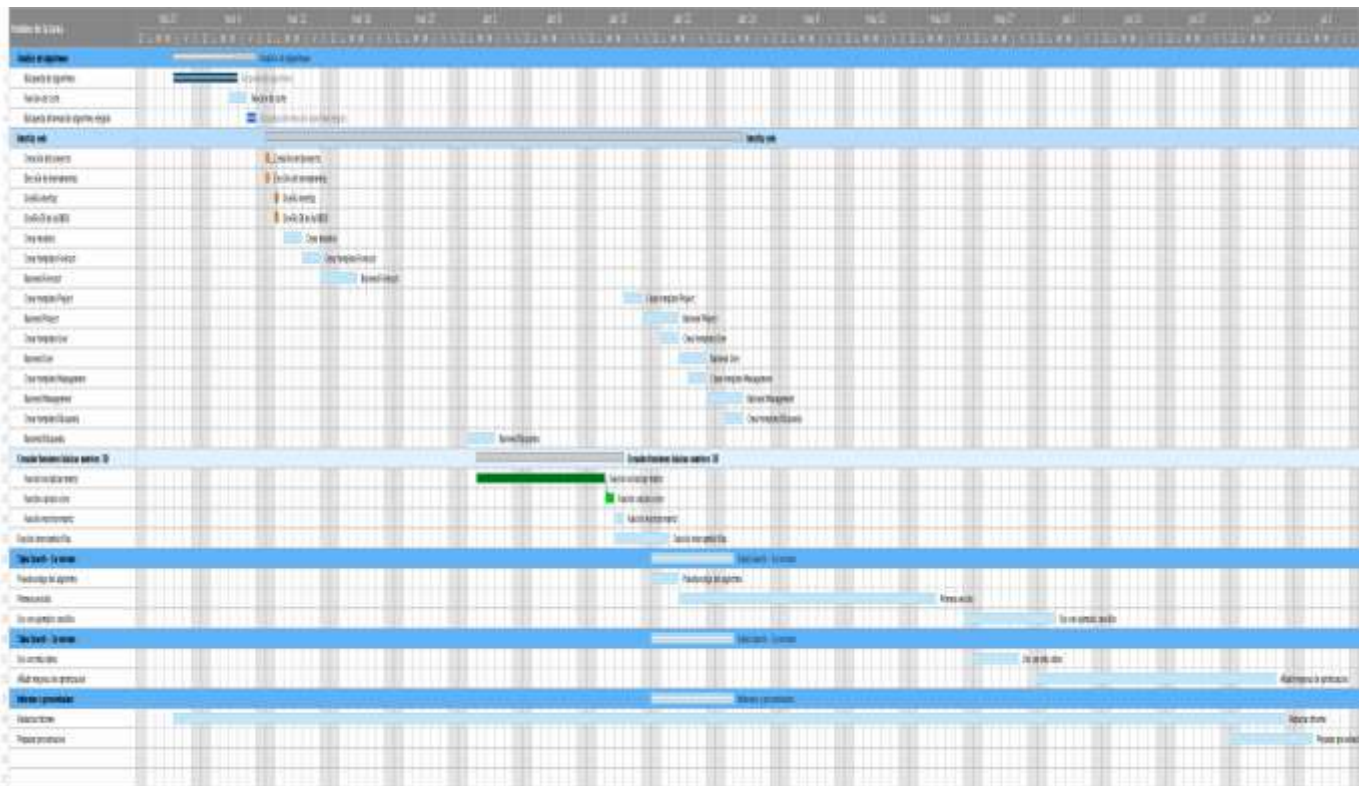
Me gustaría agradecer la ayuda recibida por mi tutor de proyecto Robert Benavente, ya que, gracias a él, las dudas y problemas obtenidos durante el desarrollo han sido solucionados.

BIBLIOGRAFIA

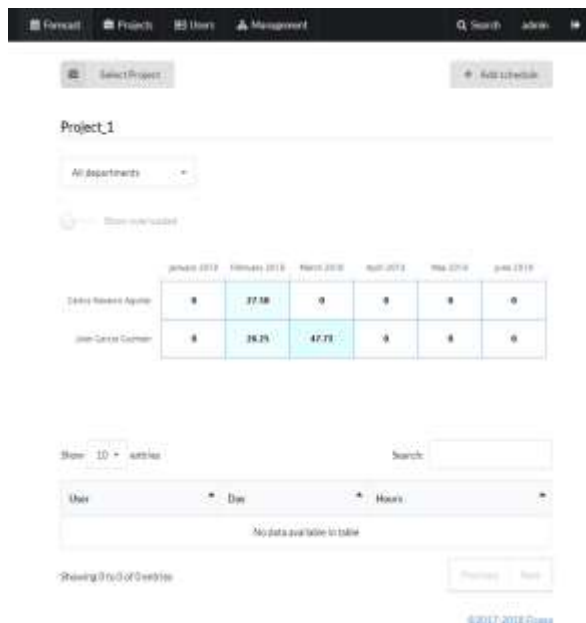
- [1] Morillo, D., Moreno, L. Y Diaz, L. (2014) "Metodologías Analíticas y Heurísticas para la Solucion del Problema de Programación de Tareas con Recursos Restringidos (RCPS): una revisión. Parte 1" Revista Ingeniería y Ciencia Vol.10, nº19:25.
- [2] Correia, L., Lampreia, L. y Saldanha-da-Gama, F. (2008) "Multi-Skill Resource-Constrained Project Scheduling: Formulation and Inequalities" Centro de Investigaçao Operacional.
- [3] Delgoshaei, A., Khairol, M. y Hang, B.T.(2016) "Pre-emptive Resource-Constrained Multimode Project Scheduling Using Genetic Algorithm: A Dynamic Forward Approach". University Putra Malaysia.
- [4] Melián Batista, B. y Glover, F. (2007) "Introducción a la Búsqueda Tabú." Rect@, Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA:35.
- [5] Glover, F. y Laguna, M.(2005) "Principles of Tabu Search" Chapman Hall/CRC.
- [6] Kolisch, R. y Hartmann, S. (1999) "Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. International Series in Operations Research & Management Science, vol 14. Springer, Boston, MA.
- [7] Moreno, L.F, Diaz, J., Peña, G. y Rivera, J. (2006) "A comparative analysis between two heuristic algorithms for solving the resource-constrained project scheduling problema (RCPS).Dyna. Vol.74 pp. 171-183.
- [8] Martí, R. (2003) "Procedimientos metaheurísticos en optimización combinatoria. Matemàtiques, vol.1, No 1, p.3-62.

APÈNDIX

A1. DIAGRAMA DE GANTT



A2. CAPTURAS DE LA WEB



Forecast Projects Users Management Search admin

Select User Edit user

Information

Surname: Navarro Aguilár

Name: Carlos

Workday: 4

Department: Test and Validation

Mail: carlosnavarro.aguilar@fossa.com

Skills

- Django: ★★★★★
- RobotFramework: ★★★★☆
- Jenkins: ★★★☆☆
- Backup Mechanism: ★★★★☆
- CAN Mapping: ★★★★☆
- Probe: ★★★★☆
- Bluetooth Call: ★★★★☆
- GPS: ★★★☆☆

Team

- Joan Garcia Guzman
- David Quiros Perez

User availability

	April 2018	May 2018	June 2018	July 2018	August 2018	September 2018	October 2018
Total	0	0	0	0	0	0	0

Forecast Projects Users Management Search admin

Users by skills

Skill: Django Minimum rate: ★★★★★ Search

114645.5

BEST COST

615330.85

WORST COST

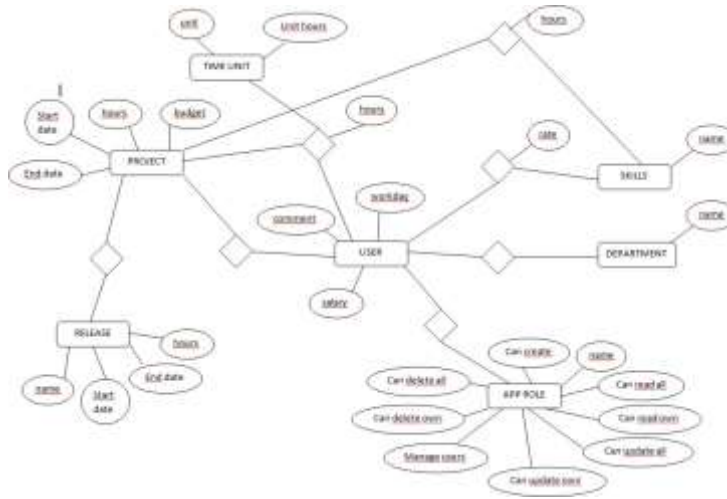
Project_2

User	Skill
Ferran Rovira Turá	GPS
Mireia Canal Blanco	Django
Jabir El Adouli Akroul	Probe
Albert Plineda Marimon	CAN Mapping
Marti Cobo Urgelles	Backup Mechanism
María Victoria Laguna Diaz	Jenkins

Project_1

User	Skill
Carlos Navarro Aguilár	RobotFramework
Joan Garcia Guzman	RQM
Antoni Boix Requesens	Jenkins
Xabier Arbulu Inausti	Django
Ignacio Palomo Ruiz	Bluetooth Call
David Quiros Perez	GPS

A3. DIAGRAMA ER



A4. CAPTURAS DE LA MATRIZ(VARIOS EJEMPLOS)

```
[ [0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. ]
[ 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. ]
[ 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ] ]
82694.4
72153.6
[[ [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. ]
[ 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. ] ]
[26/May/2018:17:34:00] "GET /ficciones/
```

```
[ [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. ]
[ 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. ]
[ 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. ] ]
83091.2
71961.6
[[ [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. ]
[ 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. ]
[ 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. ] ]
```

A5. DRIVE DEL PROYECTO

<https://drive.google.com/open?id=1VsDR2xsU1nC6lHUPSeOrn2BWqDSwvig5>