

J. Ángel Velázquez Iturbide

**OptimEx Versión 1.4:
Manual de Usuario**

Número 2020-03

Serie de Informes Técnicos DLSI1-URJC

ISSN 1988-8074

Departamento de Lenguajes y Sistemas Informáticos I

Universidad Rey Juan Carlos

Índice

1	Introducción	1
2	Visión General de la Aplicación	2
1.1	Una Sesión de Usuario	2
1.2	Instalación	7
3	Interfaz de Usuario.....	8
3.1	Tabla de Datos.....	9
3.2	Tabla de Ejecuciones.....	10
3.3	Tabla de Resumen Numérico	13
3.4	Pestaña de Resumen Gráfico	15
4	Funciones de OptimEx	18
4.1	Menú Archivo	18
4.2	Menú Ejecutar	20
4.3	Menú Tablas.....	25
4.4	Menú Configuración	25
4.5	Menú Ayuda.....	25
	Referencias.....	26

OptimEx Versión 1.4: Manual de Usuario

J. Ángel Velázquez Iturbide¹

¹ Departamento de Informática y Estadística, Universidad Rey Juan Carlos,
Móstoles, Madrid, España
angel.velazquez@urjc.es

Abstract. OptimEx es un sistema para la experimentación con algoritmos de optimización con fines educativos. El sistema permite cargar una clase Java que contenga distintos algoritmos que resuelven el mismo problema de optimización y comparar los resultados que calculan con muchos casos de prueba. OptimEx da soporte para las tareas auxiliares necesarias para un experimento: fijación de las condiciones de comparación, generación de los casos de prueba, almacenamiento y comparación de los resultados, y análisis estadístico. En la versión 1.4, OptimEx también permite medir tiempos de ejecución siguiendo el mismo esquema experimental que para la optimalidad.

Keywords: problemas de optimización, algoritmos, experimentación, optimalidad, tiempos de ejecución.

1 Introducción

OptimEx es un sistema para la experimentación con algoritmos de optimización con fines educativos [1][2]. El sistema permite cargar una clase Java que contenga distintos algoritmos que resuelven el mismo problema de optimización y comparar los resultados que los algoritmos calculan sobre muchos casos de prueba. OptimEx da soporte para las tareas auxiliares necesarias para un experimento: fijación de las condiciones de comparación, generación de los casos de prueba, almacenamiento y comparación de los resultados, y análisis estadístico. También incluye algunas facilidades para su uso educativo, como la exportación de tablas y diagramas a fichero o el cambio de idioma. La versión 1.4, OptimEx también permite medir tiempos de ejecución siguiendo el mismo esquema experimental que para la optimalidad [3].

El informe solamente consta de dos apartados. El apartado 2 realiza una breve presentación de OptimEx, mostrando una supuesta y breve sesión de usuario, y explicando su descarga e instalación. El apartado 3 presenta las funciones de OptimEx con todo detalle. Algunas figuras se presentan varias veces, aunque con la misma numeración, para facilitar su lectura del manual.

2 Visión General de la Aplicación

En este apartado realizamos una introducción a OptimEx. En primer lugar, presentamos una breve sesión de usuario. Se presenta el modo de trabajo con OptimEx, sin entrar en detalle apenas en los elementos de la aplicación, que pueden encontrarse en el apartado 3. Después, se explica la descarga de OptimEx y su instalación.

1.1 Una Sesión de Usuario

Al abrir OptimEx, se contempla una interfaz de usuario “vacía”, como en la Fig. 1.

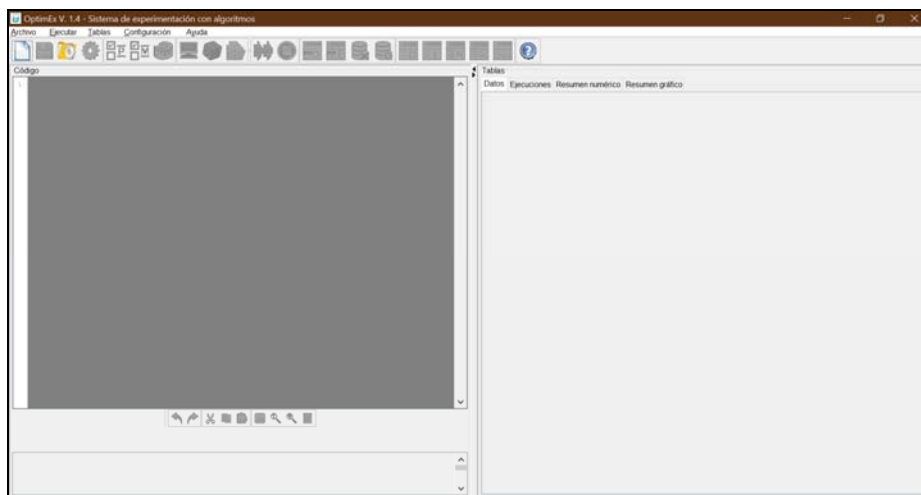


Fig. 1. Interfaz de usuario de OptimEx al inicio de sesión.

Puede observarse que la ventana consta de un menú principal y una barra de iconos en la parte superior y de dos paneles verticales en el área central. El panel izquierdo alberga un editor de texto y una pequeña área de mensajes debajo. El panel derecho alberga cuatro pestañas para tablas y diagramas.

Veamos una sesión ideal. El primer paso consiste en cargar una clase Java con los algoritmos a comparar (opción “Cargar y compilar clase...”). Supongamos que cargamos una clase llamada `LlenadoCajas.java`, que contenga varios algoritmos para las dos versiones (maximización y minimización) del problema de llenado de cajas [4, págs. 534-536]. Con esta única operación, se compila y extrae información sobre los métodos públicos de la clase, que se usará posteriormente, y se carga la clase en el editor. Si la clase produjera errores de compilación, se mostrarían uno o más mensajes explicativos en la zona de mensajes situada bajo el editor.

En un segundo paso se deben indicar las condiciones en las cuales se va a realizar el experimento. Se selecciona la función “Seleccionar problema...”, que presenta varias opciones, como muestra la Fig. 2.

El usuario debe seleccionar, de arriba abajo: (a) la signatura del problema de optimización que resuelven los métodos a analizar, (b) el criterio de análisis de los métodos, (c) en caso de comparar la optimalidad, si es un problema de maximización o de minimización, y (d) la posibilidad de cargar otros ficheros auxiliares, con juegos de prueba preparados y con restricciones de rangos para la generación aleatoria de juegos de prueba. En la Fig. 2, puede verse que se ha elegido la signatura de la versión de minimización del problema, y que la clase contiene tres algoritmos con dicha signatura. También puede verse que se va a comparar la optimalidad de los algoritmos, y a cargar los juegos de pruebas y las restricciones preparados.

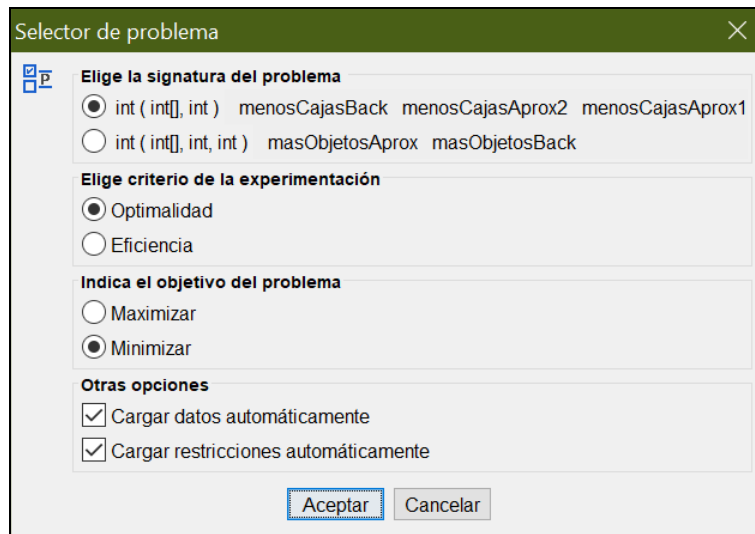


Fig. 2. Diálogo de selección del problema.

Al usuario se le mostrará un cuadro de mensaje informándole de las acciones realizadas y, en caso de haber cargado con éxito algunos juegos de datos, se mostrarán en una o varias filas de la tabla de datos.

Entonces, el usuario debe crear los juegos de prueba. Lo más frecuente es generarlos aleatoriamente. Si seleccionamos la opción "Generar datos aleatorios...", primero se abre un diálogo pidiéndonos el número de juegos de prueba a generar; supongamos que dejamos el valor por defecto, 100. A continuación, se abre un diálogo particularizado para el problema seleccionado en el paso anterior. En nuestro ejemplo, la versión de minimización del problema de llenado de cajas tiene dos parámetros: los pesos de los objetos a meter en cajas (un vector) y la capacidad de cada caja (un valor entero).

Véase la Fig. 3. Los valores mostrados son los cargados automáticamente durante la operación de selección del problema, en el paso anterior.

Los juegos de prueba generados se cargan en la tabla de datos, como puede verse en la Fig. 4. La columna primera numera cada juego de datos, mientras que cada una de las demás columnas está asociada a un parámetro. Obsérvese que la tabla de datos contiene una primera fila con su celda izquierda en verde y otras filas, en azul. La primera fila corresponde a un juego de datos cargado automáticamente al cerrar el

diálogo de “Seleccionar problema...”, incluyendo tras su número el valor optimal del problema para estos datos de entrada. Las restantes 100 filas se han generado automáticamente y solamente contienen datos de entrada.

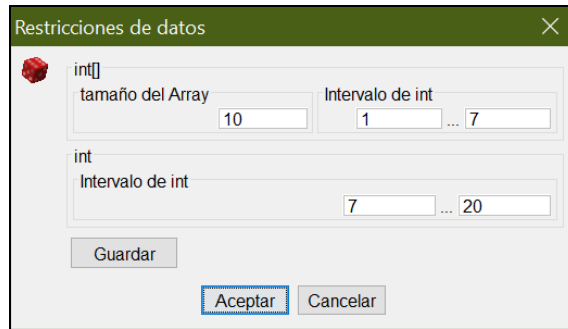


Fig. 3. Diálogo de generación aleatoria de datos con restricciones.

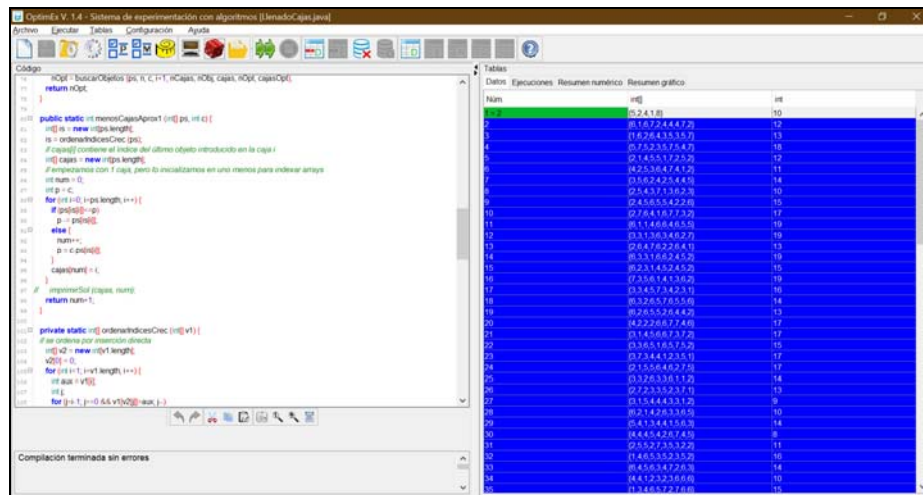


Fig. 4. Datos cargados y generados, almacenados en la tabla de datos.

El siguiente paso es seleccionar los juegos de datos a ejecutar. En la Fig. 4, las 100 filas generadas están en color azul porque, tras su generación aleatoria, quedan seleccionadas para que puedan usarse inmediatamente para ejecutar los algoritmos, sin necesidad de seleccionar sus filas.

A continuación, deben ejecutarse los algoritmos usando la opción “Ejecución”. La Fig. 5 muestra el contenido de la tabla de ejecuciones tras ejecutar los 100 juegos de datos anteriores con los tres algoritmos de minimización. Cada fila contiene los valores calculados por los algoritmos para el juego de prueba correspondiente en la tabla de datos. Cada columna contiene los valores calculados por un algoritmo. Las celdas coloreadas de verde oliva son las que tienen un valor optimal dentro de cada fila, estando coloreadas de verde claro las celdas con valores suboptimales. Si un algoritmo tiene valores optimales en todas las filas, los valores de su columna se colorean con fuente azulada, como es el caso de *menosCajasBack* en la Fig. 5.

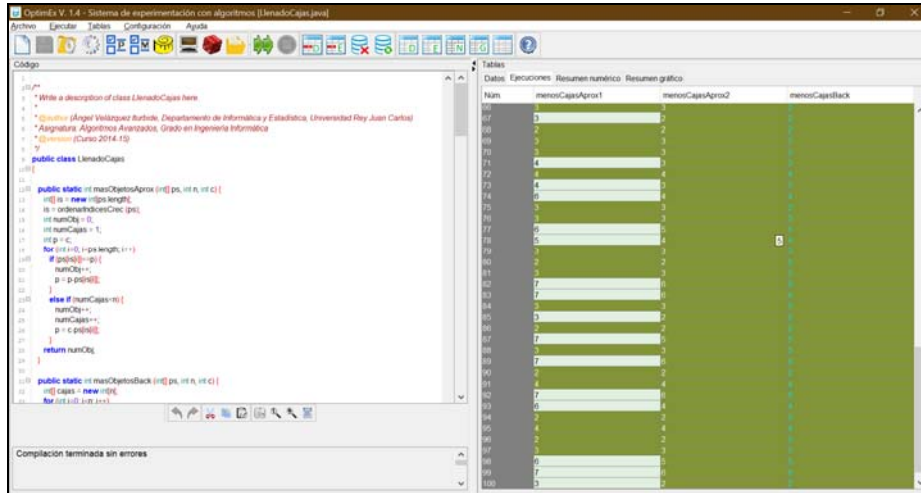


Fig. 5. Tabla de ejecuciones con los resultados calculados por los tres algoritmos.

Se tiene un resumen de los resultados del experimento en las dos pestañas restantes del panel derecho. El resumen numérico se muestra en la Fig. 6 y el gráfico, en la Fig. 7.

Medidas	menosCajasAprox1	menosCajasAprox2	menosCajasBack
Núm. ejecuciones	100	100	100
Núm. ejec. válidas del método	100	100	100
Núm. ejec. válidas en total	100	100	100
% soluciones subóptimas	36 %	1 %	0 %
% soluciones óptimas	64 %	99 %	100 %
% soluciones sobreóptimas	0 %	0 %	0 %
% valor medio subóptimo	131,25 %	133,33 %	0 %
% valor extremo subóptimo	150 %	133,33 %	0 %
% valor medio sobreoptimo	0 %	0 %	0 %
% valor extremo sobreóptimo	0 %	0 %	0 %

Fig. 6. Tabla de resumen numérico del experimento con la optimalidad.

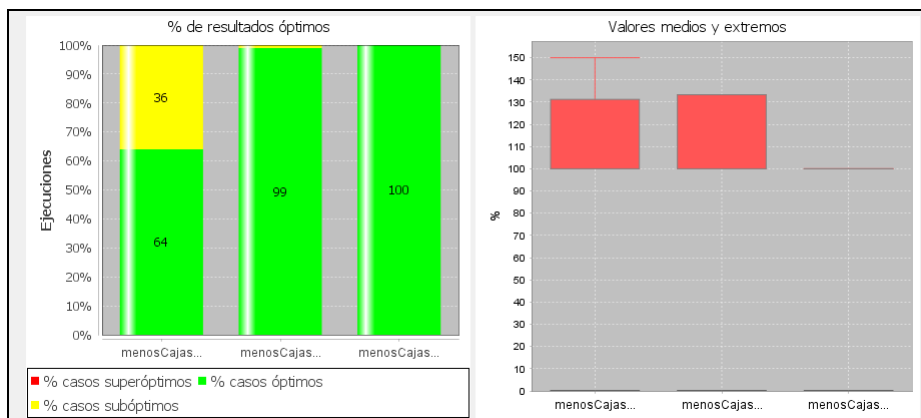


Fig. 7. Diagramas de resumen gráfico del experimento con la optimalidad.

Una vez obtenidos los resultados del experimento, los valores de todas las pestañas pueden exportarse a ficheros de distintos formatos (gráficos, Excel, texto, etc.).

En nuestro recorrido, vamos a cambiar el criterio de la experimentación, de la optimalidad a la eficiencia en tiempo. Para ello, volvemos a utilizar la opción “Seleccionar problema...” para cambiar el criterio de la experimentación a eficiencia. Véase en la Fig. 8 el diálogo con la nueva opción seleccionada.

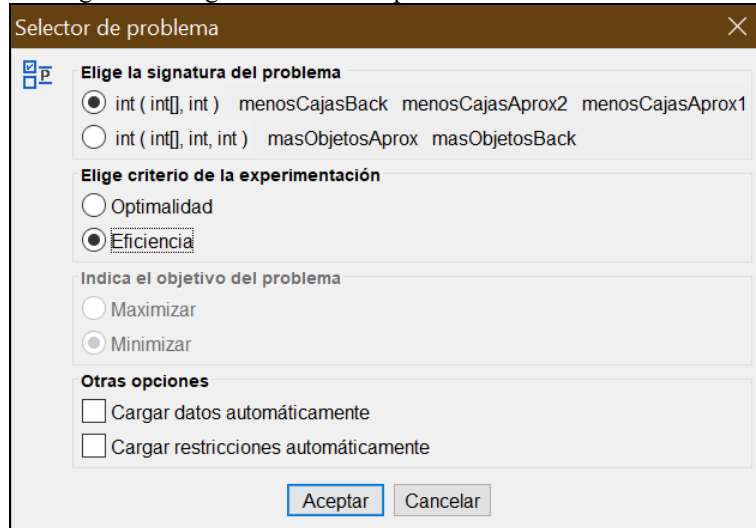


Fig. 8. Diálogo de selección de problema, con el criterio modificado.

Como resultado, se borran los resultados calculados y almacenados en la tabla de ejecuciones, así como los resúmenes. Debemos seleccionar de nuevo juegos de prueba y lanzar su ejecución. La Fig. 9 muestra los nuevos contenidos de la tabla de resultados, donde aparecen tiempos de ejecución. Todas las celdas tienen el mismo color verde claro porque no se compara su contenido.

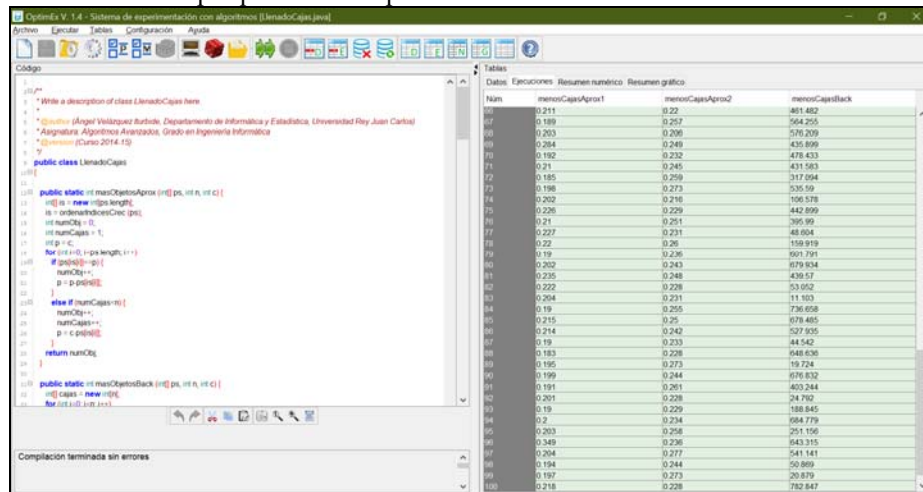


Fig. 9. Tabla de ejecuciones con tiempos de ejecución de los tres algoritmos.

La Fig. 10 y la Fig. 11 muestran los nuevos contenidos de la tabla de resumen numérico y diagramas de resumen gráfico. El resumen numérico solamente incluye tiempos máximos, mínimos y medios. Asimismo, el diagrama muestra estos tiempos de forma proporcional.

Medidas	menosCajasAprox1	menosCajasAprox2	menosCajasBack
Núm. ejecuciones	100	100	100
Núm. ejec. válidas del método	100	100	100
Núm. ejec. válidas en total	100	100	100
Tiempo máximo	25.501 ms	180.869 ms	1113.535 ms
Tiempo medio	0.803 ms	2.229 ms	403.577 ms
Tiempo mínimo	0.160 ms	0.193 ms	2.155 ms

Fig. 10. Tabla de resumen numérico del experimento con la eficiencia.

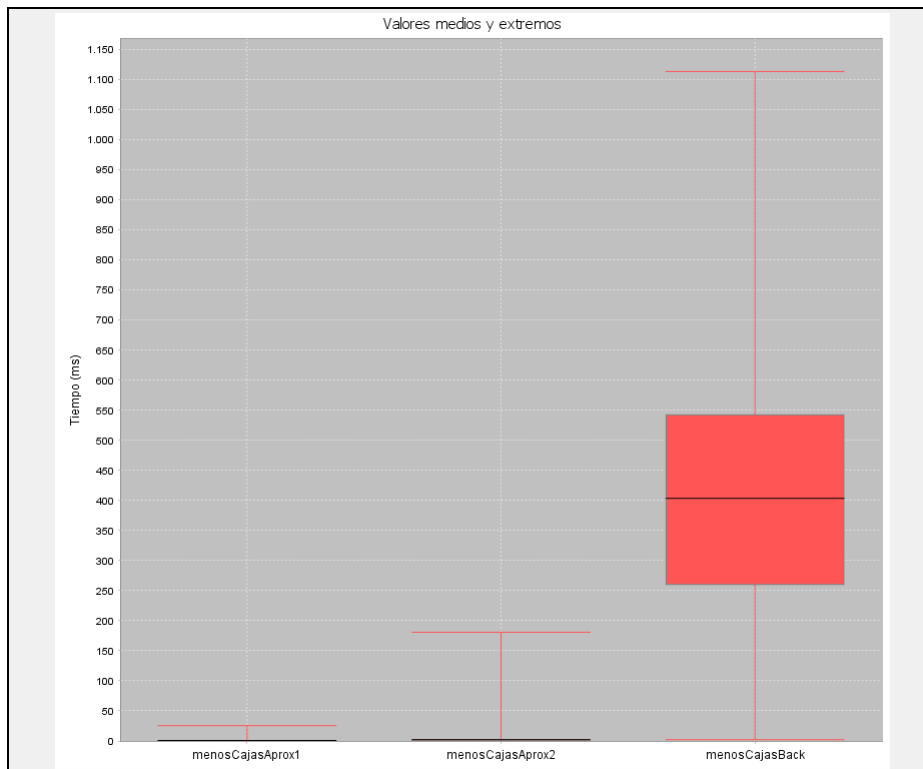


Fig. 11. Diagramas de resumen gráfico del experimento con la eficiencia.

1.2 Instalación

El sistema OptimEx es distribuye como un fichero JAR. Se encuentra disponible en la siguiente URL:

<https://lite.etsii.urjc.es/tolos/optimex>

Para ejecutar OptimEx en Windows, basta con hacer doble clic en el fichero JAR. Se necesita tener instalado un JDK (*Java Development Kit*) para poder compilar las clases de Java y ejecutar sus métodos.

La primera vez que se ejecuta, OptimEx busca automáticamente una JDK. Le sugiere una JDK al usuario, en un diálogo sencillo, dándole la posibilidad de navegar por el sistema de ficheros para elegir otra (véase Fig. 12).

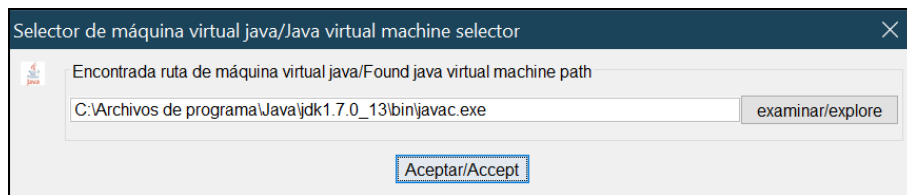


Fig. 12. Diálogo de selección de una máquina virtual de Java.

Al acabar la primera sesión de OptimEx, crea un fichero de configuración que incluye, entre otra información, el camino de la máquina virtual seleccionada. En consecuencia, este diálogo no se muestra en sesiones posteriores.

3 Interfaz de Usuario

La interfaz de usuario de OptimEx consta de tres zonas:

- Parte superior. Contiene el menú principal, con submenús desplegables, que agrupan 27 funciones. Como en otras aplicaciones, un título de función que termina en puntos suspensivos significa que se abrirá un diálogo. Debajo del menú principal se encuentra la barra de iconos, con 22 iconos. OptimEx es un sistema sencillo, por lo que la mayor parte de sus funciones se encuentran disponibles en la barra de iconos. Véase la Fig. 13.

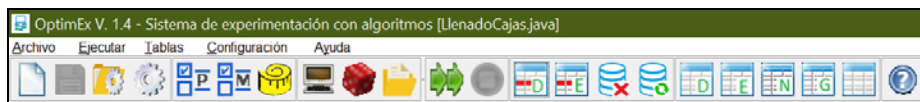


Fig. 13. Parte superior izquierda de la interfaz de usuario.

- Panel de código. Se ubica en la mitad izquierda de la pantalla. Agrupa el editor en su parte superior y una pequeña zona de mensajes en su parte inferior. El editor es multicolor para facilitar la lectura del código y presenta las líneas numeradas a su izquierda (véase Fig. 14). En las cabeceras de los métodos hay, entre los números de línea y el código, un botón etiquetado "+" o "-" que permite contraer o expandir, respectivamente, su código. El editor incluye 9 iconos en su parte inferior para operaciones de edición (véase Fig. 14). De izquierda a derecha, son: deshacer (atajo: Ctrl+Z), rehacer (Ctrl+V), cortar (Ctrl+X), copiar (Ctrl+C), pegar (Ctrl+V), ir a línea, buscar, buscar siguiente, y seleccionar todo (Ctrl+A).

El área de mensajes muestra los mensajes de error del compilador de Java. Cada mensaje se muestra descompuesto en tres partes para facilitar su comprensión. En la Fig. 14 se muestra un mensaje de error producido por la falta del signo de punto y coma en la línea 28.

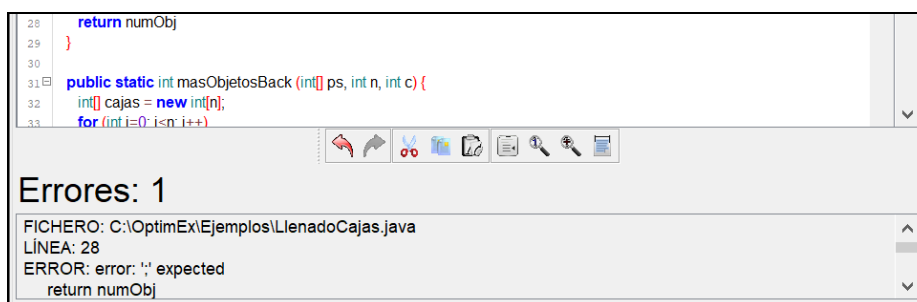


Fig. 14. Parte inferior del panel de código.

- Panel de tablas. Se ubica en la mitad derecha de la pantalla. Agrupa cuatro pestañas, de las cuales tres contienen tablas y una, diagramas. Dada su mayor complejidad, se presentan en detalle en los subapartados siguientes.

3.1 Tabla de Datos

La tabla de datos alberga los datos de entrada a usar juegos de prueba. Los valores de cada juego de prueba se almacenan en una fila distinta de la tabla. La columna primera contiene una numeración de los juegos de prueba, comenzando desde 1, y el resto de columnas, el valor de cada parámetro en los distintos juegos de prueba.

Los juegos de prueba preparados previamente constituyen un caso especial. Pueden cargarse mediante la función “Seleccionar problema...”. En la tabla de datos, se distinguen de los demás juegos de prueba en que su casilla de numeración está coloreada en color verde y además contiene su resultado optimal (véase Fig. 4). El resto de celdas de la tabla se producen con funciones del menú Archivo (véase apartado 4.1). Se presentan en color gris claro y sin resultado.

La operación de manipulación más frecuente sobre la tabla de datos consiste en seleccionar las filas que se van a usar como juegos de prueba en una operación de ejecución. Las filas seleccionadas se resaltan en color azul. La Fig. 4 muestra la tabla con todos los juegos de prueba seleccionados salvo el primero, que es un juego preparado.

Pueden realizarse tres operaciones de selección con el ratón:

- Pulsar en cualquier lugar de una fila. Queda seleccionada dicha fila.
- Pulsar en una fila y arrastrar hacia arriba o hacia abajo. Quedan seleccionadas todas las filas sobre las cuales se ha arrastrado el ratón.
- Pulsar en la cabecera de la tabla. Si no están seleccionadas todas las filas, produce la selección de todas. Si estuvieran seleccionadas todas, dejan de estarlo.

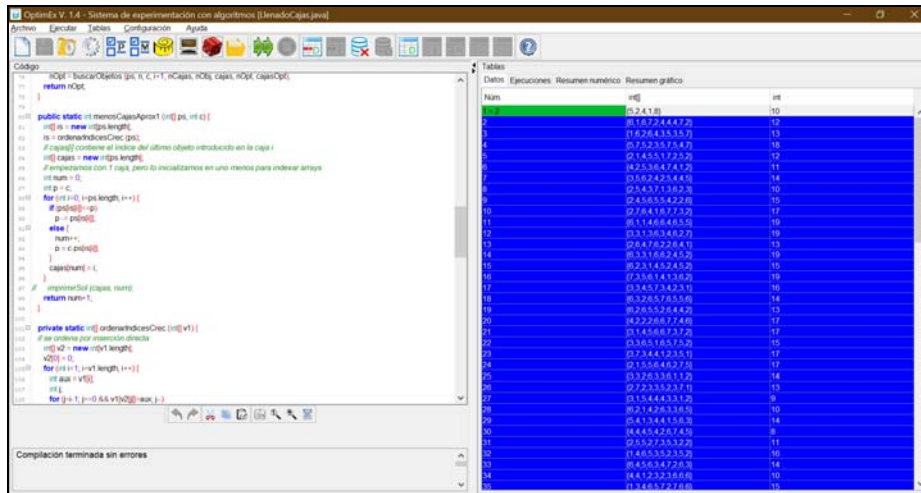


Fig. 4. Datos cargados y generados, almacenados en la tabla de datos.

3.2 Tabla de Ejecuciones

La tabla de ejecuciones alberga los resultados o los tiempos de ejecución de los algoritmos, tras ejecutarse con los juegos de prueba y según el criterio de experimentación seleccionados (véanse Fig. 5 y Fig. 9).

La columna primera tiene color gris oscuro y contiene la misma numeración que los juegos de prueba de la tabla de datos. El resto de columnas contienen los valores o los tiempos obtenidos al ejecutar cada algoritmo. Las celdas se colorean de forma distinta, dependiendo de que el criterio de la experimentación sea optimalidad o eficiencia en tiempo.

Si el criterio de la experimentación es la optimalidad (véase Fig. 5), el valor calculado por cada algoritmo para cada juego de datos se almacena en la fila y columna correspondiente. Dichos valores se comparan dentro de cada fila según el criterio de optimización seleccionado (maximizar o minimizar) y se determina cuáles son óptimas para el juego de prueba correspondiente. Las celdas de una fila que contengan valores óptimos se colorean en verde oliva y los demás, en verde claro. Las columnas que tengan todas sus celdas de color verde oliva presentan sus valores en fuente azul claro para destacar que, según el estado actual del experimento, se trata de un algoritmo exacto.

En la Fig. 5, los dos primeros algoritmos han producido resultados subóptimos con algunos casos de prueba, mientras que el algoritmo tercero tiene un 100% de casos óptimos. Aunque los casos mostrados pueden dar la impresión de que el segundo algoritmo es exacto, la fuente blanca indica que hay casos subóptimos, aunque no visibles en la figura.

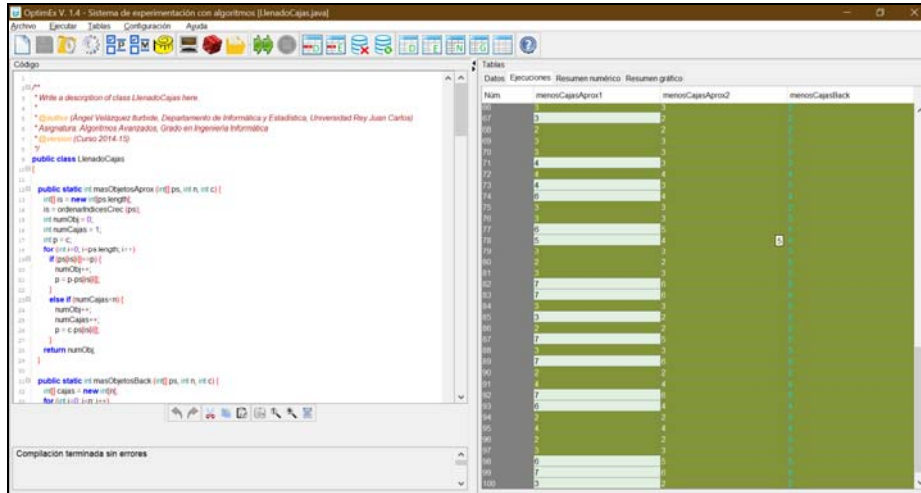


Fig. 5. Tabla de ejecuciones con los resultados calculados por los dos algoritmos.

Las celdas también pueden contener información de otros dos tipos. Por un lado, el usuario puede indicar con la función “Seleccionar método...” que un método dado es exacto. Obsérvese que en una situación como la descrita, en la que todos los valores de una fila se comparan entre sí para determinar el máximo (o el mínimo en un problema de minimización), solamente existe el valor optimal y, en todo caso, valores subóptimos. Sin embargo, al marcar un método como exacto, se considera que todos los valores que calcula son optimales. Por tanto, podría darse la situación en la que existan otros valores mayores (o menores en problemas de minimización) en la misma fila. Estas celdas corresponden a situaciones que no deberían darse, por lo cual se colorean rojizos. Se dará más información sobre estas situaciones en la tabla de resumen numérico (apartado 3.3).

Supongamos que, en nuestro ejemplo, se ha seleccionado el algoritmo inexacto *menosCajasAprox2* como exacto, lo cual produce que el algoritmo *menosCajasBack* genere un valor mayor en un caso. La Fig. 15 muestran la tabla de resumen numérico, con la situación anormal de que hay un valor “sobreóptimo”.

Medidas	menosCajasAprox1	menosCajasAprox2	menosCajasBack
Núm. ejecuciones	100	100	100
Núm. ejec. válidas del método	100	100	100
Núm. ejec. válidas en total	100	100	100
% soluciones subóptimas	35 %	0 %	0 %
% soluciones óptimas	65 %	100 %	99 %
% soluciones sobreóptimas	0 %	0 %	1 %
% valor medio subóptimo	131,19 %	0 %	0 %
% valor extremo subóptimo	150 %	0 %	0 %
% valor medio sobreóptimo	0 %	0 %	75 %
% valor extremo sobreóptimo	0 %	0 %	75 %

Fig. 15. Tabla de resumen numérico del experimento con la optimalidad, con casos sobreóptimos.

Asimismo, la Fig. 16 muestra el resumen gráfico. Para *menosCajasBack*, el diagrama de casos presenta en color rojo el 1% de porcentaje de casos “sobreóptimos”, mientras que el diagrama de desviación muestra que calcula valores por debajo del 100%.

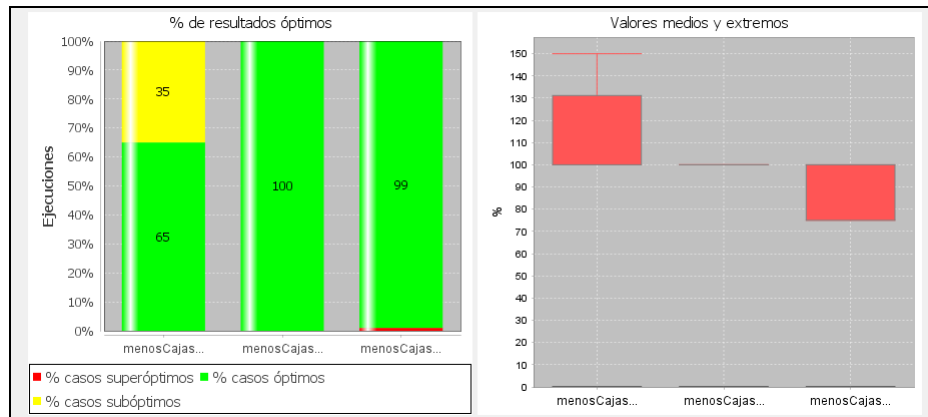


Fig. 16. Diagramas de resumen gráfico del experimento con la optimalidad, con casos sobreóptimos.

Por último, si algún algoritmo produjera un error de ejecución para algún juego de datos, su celda correspondiente contendrá el mensaje “Tiempo excedido” y se coloreará de gris claro.

Si el criterio de la experimentación es la eficiencia (véase Fig. 9), los tiempos medidos no se comparan entre sí. Por tanto, todas las celdas se colorean del mismo color, verde claro. Los tiempos aparecen expresados en milisegundos.

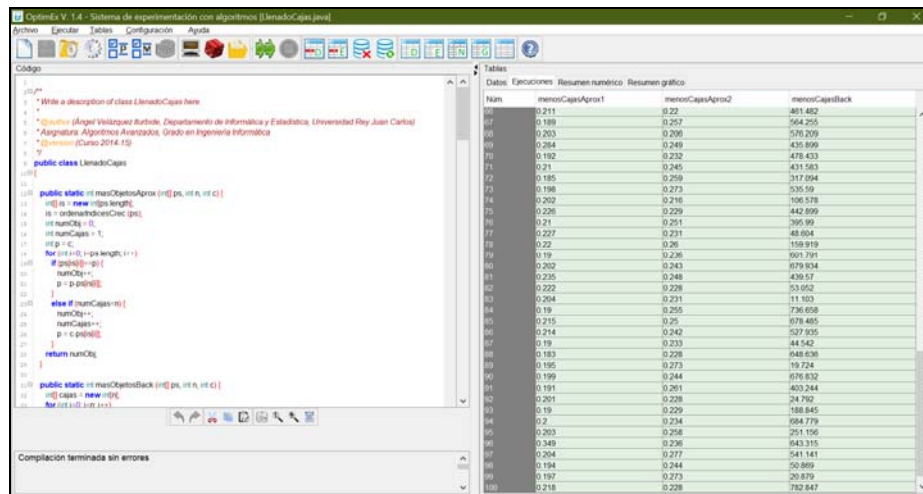


Fig. 9. Tabla de ejecuciones con tiempos de ejecución de los dos algoritmos.

3.3 Tabla de Resumen Numérico

La tabla de resumen numérico alberga un resumen estadístico de los resultados o los tiempos de ejecución medidos para los algoritmos, según el criterio de la experimentación seleccionado (véanse Fig. 6 y Fig. 10). La columna primera identifica la medida incluida en cada fila y el resto de columnas, los valores que toma dicha medida para cada algoritmo.

Cuando el criterio de experimentación es la optimalidad, se presentan 10 medidas (véase Fig. 6).

Medidas	menosCajasAprox1	menosCajasAprox2	menosCajasBack
Núm. ejecuciones	100	100	100
Núm. ejec. válidas del método	100	100	100
Núm. ejec. válidas en total	100	100	100
% soluciones subóptimas	36 %	1 %	0 %
% soluciones óptimas	64 %	99 %	100 %
% soluciones sobreóptimas	0 %	0 %	0 %
% valor medio subóptimo	131,25 %	133,33 %	0 %
% valor extremo subóptimo	150 %	133,33 %	0 %
% valor medio sobreoptimo	0 %	0 %	0 %
% valor extremo sobreóptimo	0 %	Ejecución nº: 0	0 %

Fig. 6. Tabla de resumen numérico del experimento con la optimalidad.

Las medidas son de tres tipos:

- Números de ejecuciones:
 - Número de ejecuciones. Es el número de juegos de prueba sobre los que se han ejecutado todos los algoritmos seleccionados. Su valor es igual en todas las columnas.
 - Número de ejecuciones validas del método. Para cada método, es el número de juegos de prueba que han terminado su ejecución normalmente, sin un error en tiempo de ejecución. Se considera un error en tiempo de ejecución que el algoritmo haya lanzado una excepción o que OptimEx haya detenido una ejecución porque ha detectado que tardaba un tiempo excesivo.
 - Número de ejecuciones válidas en total. Es el número de juegos de prueba para los que todos los algoritmos han terminado su ejecución normalmente.

En la Fig. 6 no se ha producido ningún error en tiempo de ejecución, por lo cual los tres algoritmos presentan 100 ejecuciones en las tres filas.

- Porcentajes de soluciones válidas:
 - % soluciones subóptimas. Es el porcentaje de soluciones válidas de un algoritmo que han sido suboptimales. Una solución es subpotimal en un problema de maximización si es menor que una solución optimal para el mismo juego de datos. Asimismo, es suboptimal en un problema de minimización si es mayor (como sucede en el ejemplo mostrado).
 - % soluciones óptimas. Es el porcentaje de soluciones válidas de un algoritmo que han sido optimales.

- % soluciones sobreóptimas. Damos el nombre de soluciones “sobreóptimas” a ciertas situaciones que no deberían suceder. Solamente aparecen casos “sobreóptimos” si el usuario marca con la función “Seleccionar métodos...” que un algoritmo es exacto. Entonces, en cada fila se toman como valores de referencia los valores calculados por dicho algoritmo. Obviamente, el algoritmo marcado tendrá un 100% de soluciones optimales. Sin embargo, puede suceder que algún otro algoritmo produzca en alguna fila un valor mayor que la solución “óptima” si el problema es de maximización o inferior si es de minimización. Este campo mide el porcentaje de soluciones válidas de cada algoritmo que corresponden a estos casos, es decir que son soluciones “sobreóptimas”.

En la Fig. 6 puede verse que el tercer algoritmo es exacto, con 100% de casos optimales, mientras que el primero y el segundo son inexactos, con 64% y 99%, respectivamente, de casos optimales y 36% y 1%, respectivamente, de casos suboptimales.

- Desviaciones respecto a las soluciones optimales:
 - % valor medio subóptimo. Se considera que la solución óptima de cada fila corresponde al 100%, A cada valor subóptimo o sobreóptimo situado en la misma fila le corresponde un porcentaje proporcional. Esta medida calcula el valor medio del porcentaje correspondiente a los casos válidos suboptimales. Obsérvese que, para problemas de maximización, será un valor inferior a 100% y para problemas de minimización, superior.
 - % valor extremo subóptimo. Al igual que en la medida anterior, se considera que la solución óptima de cada fila corresponde al 100%, Esta medida calcula el valor más alejado del 100% en los casos válidos suboptimales. De nuevo, para problemas de maximización, será un valor inferior a 100% y para problemas de minimización, superior.
 - % valor medio sobreóptimo. Se define se forma análoga al “% valor medio subóptimo” pero para los casos sobreóptimos.
 - % valor extremo sobreóptimo. Se define se forma análoga al “% valor extremo subóptimo” pero para los casos sobreóptimos.

En la Fig. 6 puede verse que los casos suboptimales del algoritmo primero alcanzan en media el 132,25% del valor óptimo, aunque en el caso más desfavorable asciende hasta el 150%. También puede observarse que, cuando el ratón se sitúa encima de alguna de las filas de valores extremos, OptimEx muestra el número de los casos de prueba correspondientes.

Cuando el criterio de experimentación es la eficiencia, se presentan 6 medidas (véase Fig. 10).

Medidas	menosCajasAprox1	menosCajasAprox2	menosCajasBack
Núm. ejecuciones	100	100	100
Núm. ejec. válidas del método	100	100	100
Núm. ejec. válidas en total	100	100	100
Tiempo máximo	25.501 ms	180.869 ms	1113.535 ms
Tiempo medio	0.803 ms	2.229 ms	403.577 ms
Tiempo mínimo	0.160 ms	0.193 ms	2.155 ms

Fig. 10. Tabla de resumen numérico del experimento con la eficiencia.

Las medidas son de dos tipos:

- Números de ejecuciones. Se definen igual que con el criterio de optimalidad:
 - Número de ejecuciones.
 - Número de ejecuciones validas del método.
 - Número de ejecuciones válidas en total.
- Tiempos de ejecución:
 - Tiempo máximo. Es el tiempo máximo tardado por el algoritmo en alguna ejecución válida.
 - Tiempo medio. Es el tiempo medio tardado por el algoritmo en todas sus ejecuciones válidas.
 - Tiempo mínimo. Es el tiempo mínimo tardado por el algoritmo en alguna ejecución válida.

En la Fig. 10 puede verse que los tiempos de ejecución del algoritmo tercero (de vuelta atrás) son mucho mayores que los otros dos (algoritmos aproximados). No obstante, el algoritmo segundo también tiene un tiempo relativamente alto en el caso peor.

3.4 Pestaña de Resumen Gráfico

La pestaña de resumen gráfico presenta en formato gráfico los mismos resultados o tiempos de ejecución presentados en la tabla de resumen numérico, salvo los números de ejecuciones (véanse Fig. 7 y Fig. 11). Presenta uno o dos diagramas por algoritmo.

Cuando el criterio de experimentación es la optimalidad y algún algoritmo es exacto, se presentan 2 diagramas por algoritmo (véase Fig. 7).

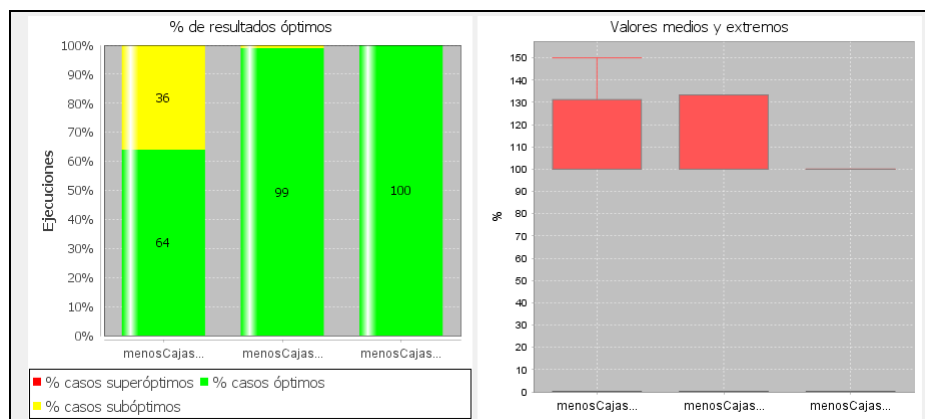


Fig. 7. Diagramas de resumen gráfico del experimento con la optimalidad.

Veamos cada diagrama:

- Diagrama de casos. Está situado a la izquierda de la pestaña. Recoge los porcentajes de soluciones válidas que son subóptimas, óptimas y sobreóptimas

para cada algoritmo. Tiene el mismo formato que el diagrama de “columna 100% apilada” de Excel.

Cada algoritmo tiene asociada una barra cuya altura total suma el 100% de casos. La barra se parte en tres zonas, que corresponden a los porcentajes de casos subóptimos (en amarillo), óptimos (en verde) y sobreóptimos (en rojo). Si algún porcentaje es nulo, no se dibuja. Para mayor precisión, cada zona contiene también su porcentaje correspondiente, siempre que quepa en la misma.

Supongamos que un diagrama de casos presenta tres porcentajes. En un problema de maximización, se presentan, de arriba abajo, los casos sobreóptimos, óptimos y subóptimos. En un problema de minimización, los casos sobreóptimos y subóptimos aparecen invertidos, es decir, los casos subóptimos se presentan arriba (calculan costes mayores) y los sobreóptimos debajo (calculan costes menores).

Véase la Fig. 7, con porcentajes subóptimos y óptimos solamente, y la Fig. 16, con porcentajes de los tres tipos.

- Diagrama de desviación. Está situado a la derecha de la pestaña. Es un diagrama basado en el diagrama estadístico de cajas o bigotes (*boxplot*).

Supongamos que es un problema de maximización y que el algoritmo tiene casos subóptimos. El diagrama de desviación presenta un rectángulo que va desde el 100% al “% valor medio subóptimo”. Además, se prolonga una línea vertical hasta una línea horizontal situada en la posición correspondiente a “% valor extremo subóptimo”. De igual forma, si el algoritmo presenta valores sobreóptimos, se prolongará hacia arriba una representación de las dos medidas de dispersión correspondientes. Si el problema es de minimización, como en el ejemplo, la representación es similar, pero los casos subóptimos se dibujan por encima del 100% y los sobreóptimos, por debajo.

Véase la Fig. 7, con casos subóptimos solamente, y la Fig. 16, con casos subóptimos y sobreóptimos. Recuérdese que solamente presentan desviación los algoritmos inexactos.

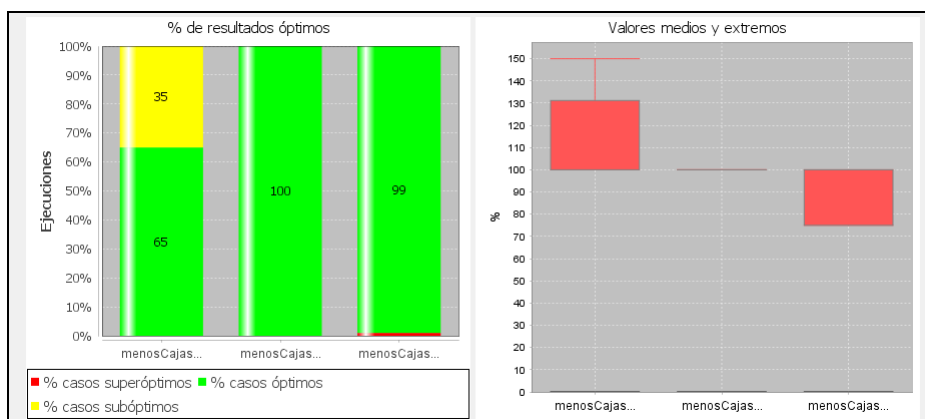


Fig. 16. Diagramas de resumen gráfico del experimento con la optimalidad, con casos sobreóptimos.

Cuando el criterio de experimentación es la eficiencia en tiempo, se presenta un solo diagrama por algoritmo (véase Fig. 11). De nuevo, el diagrama está basado en el diagrama estadístico de cajas.

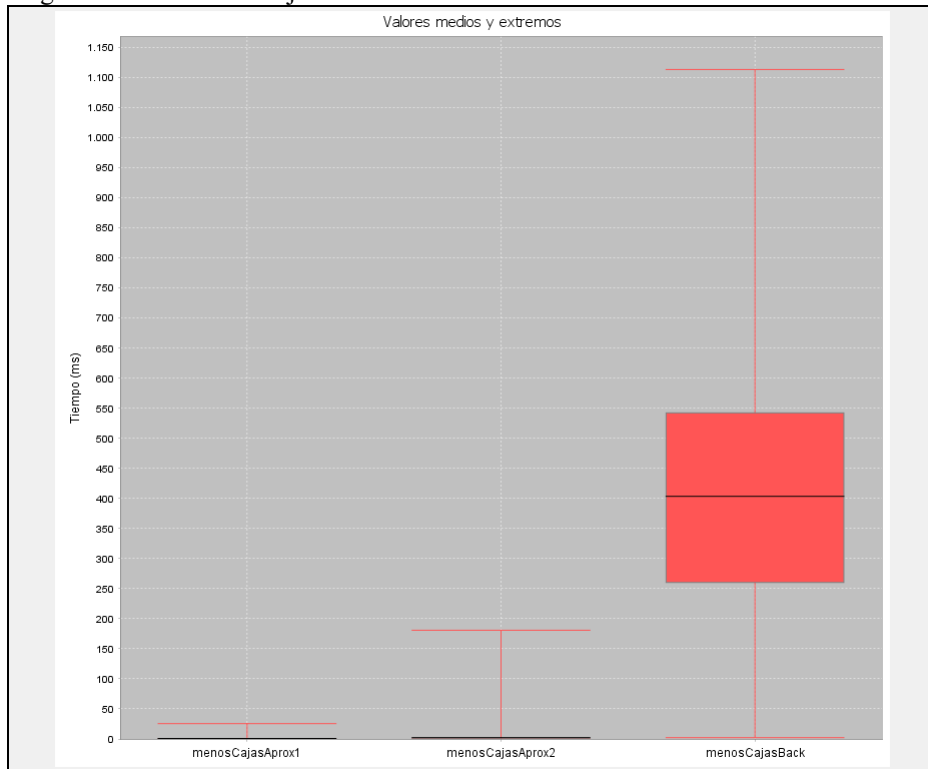


Fig. 11. Diagramas de resumen gráfico del experimento con la eficiencia.

El significado de los elementos del diagrama es el siguiente:

- Los tiempos máximo y mínimo corresponden con las líneas horizontales rojas situadas en las partes superior e inferior del diagrama.
- La caja sólida, roja, engloba el 50% de los tiempos medidos, en concreto los tiempos comprendidos entre los 25% tiempos menores y el 75%.
- Una línea horizontal negra representa el tiempo medio. Obsérvese en la Fig. 7(b) que no coincide con el tiempo situado en la posición 50%, ya que este sería la mediana y estaría dentro de la caja. Obsérvese que esta línea negra corresponde al tiempo medio, no a la mediana de los tiempos medidos, por lo que puede aparecer fuera de la caja.

El resumen gráfico permite interaccionar mediante zoom porque es frecuente que el rango de tiempo medido sea grande, resultando en un diagrama donde es difícil ver los detalles.

Conviene avisar de que algunos tiempos medidos son desmesuradamente mayores que los tiempos medidos para otros datos del mismo tamaño. Esto se debe a peculiaridades de la máquina virtual de Java.

4 Funciones de OptimEx

Veamos las funciones de OptimEx siguiendo la estructura del menú principal. No se explican apenas algunas funciones cuya etiqueta es autoexplicativa o que ya se han explicado suficientemente en los apartados anteriores.

4.1 Menú Archivo

El menú Archivo agrupa las funciones relacionadas con el manejo de ficheros o entrada de juegos de prueba. Incluye 10 opciones (organizadas en 4 grupos):

- “Nueva clase”. Se crea una clase vacía para que usuario la edite y guarde. En caso de que ya se haya realizado un experimento en esta sesión, OptimEx pide confirmación al usuario porque se perdería el contenido de todo el panel de tablas.
- “Guardar clase...”. Guarda en disco la clase cargada en el editor.
- “Cargar y compilar clase...”. Se navega por el sistema de ficheros, se selecciona un fichero JAVA, se compila y se carga en el editor. También pide confirmación en caso de haber realizado un experimento.
- “Compilar clase”. Compila la clase cargada en el editor. También pide confirmación en caso de haber realizado un experimento.
- “Introducir datos por teclado”. Se presenta un diálogo a medida para que el usuario introduzca un juego de prueba. Indica el tipo de datos de cada parámetro y los valores válidos, según las restricciones indicadas. La Fig. 17 muestra el diálogo para la versión de minimización del problema del llenado de cajas.

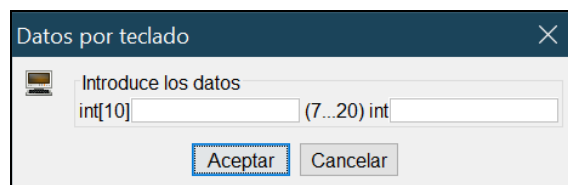


Fig. 17. Diálogo para introducir datos por teclado.

- “Generar datos aleatorios...”. Se presentan sucesivamente dos diálogos. En el primer diálogo se pide el número de casos de prueba a generar. Por defecto, se generan 100. El segundo diálogo se presenta a medida para que se introduzca la longitud de los arrays y los rangos de valores de las variables simples. Por defecto, la longitud de los arrays es 12 y los valores enteros varían entre 0 y 30. La Fig. 18 muestra el diálogo que se presenta para el problema de ejemplo en caso de no haber cargado automáticamente fichero de restricciones.

El segundo diálogo incluye una opción Guardar (véase la Fig. 18) que permite que los valores de los campos del diálogo se almacenen como restricciones para generar aleatoriamente datos para mismo problema en futuras sesiones. Un fichero de restricciones es un fichero de texto con

extensión CNSTR cuyo formato se ve en el apartado 3.3 (función “Selección de problema...”).

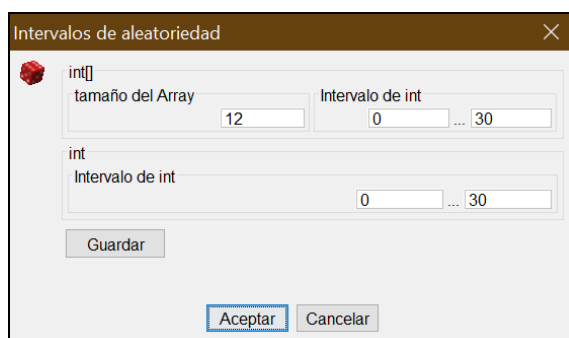


Fig. 18. Diálogo de generación aleatoria de datos sin restricciones.

- “Cargar datos de fichero...”. Permite cargar juegos de prueba de un fichero TXT o XML, bien creados manualmente bien resultantes de exportar la tabla de datos. La Fig. 19 muestra el formato de los datos para el problema de ejemplo. Puede verse que consta de varias líneas, mostrando la primera los tipos de datos de los parámetros y las siguientes, los distintos juegos de prueba. En la Fig. 19 se muestra el comienzo de los datos mostrados en la Fig. 4. El primero es un juego de prueba de longitud 5 cargado al seleccionar el problema, mientras que el segundo y posteriores son juegos de longitud 10 generados aleatoriamente.

```
int[] int
{5,2,4,1,8} 10
{6,1,6,7,2,4,4,4,7,2} 12
{1,6,2,6,4,3,5,3,5,7} 13
{5,7,5,2,3,5,7,5,4,7} 18
```

Fig. 19. Fichero de texto con datos de entrada.

- “Exportar datos...”. Exporta la cabecera y las filas de la tabla de datos en un fichero. Sin embargo, no se exporta la columna de numeración. El usuario puede elegir como formato del fichero entre un fichero de texto (TXT), de marcas (XML) o un fichero gráfico (GIF, JPG o PNG). Puede encontrarse más información sobre el formato XML en la ayuda interactiva de OptimEx.
- “Exportar ejecuciones...”. Exporta la cabecera y las filas de la tabla de ejecuciones. Sin embargo, no se exporta la columna de numeración. El usuario puede elegir como formato del fichero entre un fichero Excel (XLS) o un fichero gráfico (GIF, JPG o PNG).
- “Exportar resumen numérico...”. Exporta la cabecera y las filas de la tabla de resumen numérico. El usuario puede elegir como formato del fichero entre un fichero Excel (XLS) o un fichero gráfico (GIF, JPG o PNG).

- “Exportar resumen gráfico...”. Exporta el contenido de la pestaña de resumen gráfico. El usuario puede elegir un formato gráfico (GIF, JPG o PNG).
- “Exportar todas las tablas...”. Se exporta el contenido de las cuatro pestañas del panel de tablas en cuatro hojas de un fichero Excel, llamadas Datos, Ejecuciones, ResumenNumerico y Graficas. En los tres primeros casos, se exporta tanto la cabecera de la tabla como sus filas, pero no se exporta la columna de numeración.
- “Salir”. Si se ha experimentado, pide una confirmación al usuario.

4.2 Menú Ejecutar

El menú Ejecutar agrupa las funciones relacionadas con la preparación y realización del experimento. Incluye 4 opciones, que presentamos a continuación (con “otras opciones” desglosado):

- “Seleccionar problema...”. El diálogo permite configurar la experimentación de forma adecuada (véase Fig. 2). Si esta función de selección se invoca y modifica alguna opción tras haber experimentado, el sistema nos avisará de que se perderán los datos calculados. En caso de confirmar la acción, se conserva el contenido del editor y de la tabla de datos, pero se borrará el contenido del resto de tablas.

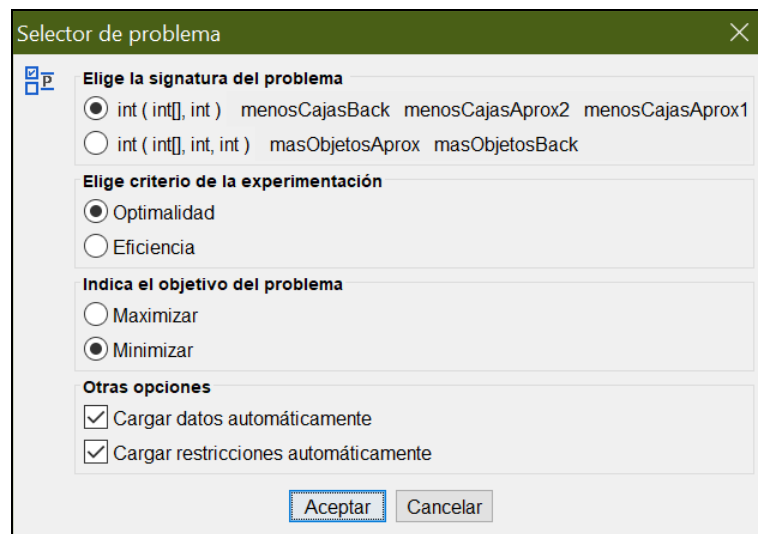


Fig. 2. Diálogo de selección de problema.

El diálogo consta de las siguientes partes:

- Signatura. Presenta las signaturas de los métodos públicos de la clase. Se presenta el tipo de datos del resultado, seguido de los tipos de datos de los parámetros, separados por blancos y entre paréntesis. Si hubiera

una única signatura, esta selección está deshabilitada y la signatura está preseleccionada. Para facilitar la selección, a la derecha de cada signatura se muestra el identificador de los métodos que tienen dicha signatura.

- Criterio de la experimentación. Permite elegir entre optimalidad o eficiencia (tiempos de ejecución).
- Objetivo a maximizar. Esta opción sólo se habilita cuando el criterio de la experimentación es optimalidad, permitiendo indicar si la función objetivo del problema es de maximización o de minimización.
- Carga de casos de prueba. Permite cargar automáticamente juegos de pruebas preparados. Se ha concebido para cargar casos de prueba cuyo resultado óptimo se conoce, como ejemplos incluidos en el enunciado de un problema. En caso de marcar esta casilla, OptimEx busca, en el mismo directorio donde se encontraba la clase cargada en el editor, un fichero de texto el mismo nombre que la clase y extensión DATA.

El fichero puede contener casos de prueba para todas las signaturas de métodos públicos de la clase. La Fig. 20 muestra un ejemplo para el problema del llenado de cajas. Puede observarse que aparecen los casos de prueba para dos problemas, separados entre sí por una línea con un punto y coma. El fichero comienza con un caso de prueba para la versión de maximización del problema y sigue con otro caso para la versión de minimización.

```
int[] int int -> int
{2,3,4,5} 2 7 -> 4
;
int[] int -> int[]
{5,2,4,1,8} 10 -> 2
```

Fig. 20. Fichero de texto de carga automática, con datos de entrada.

Para cada problema, debe escribirse su signatura en una primera línea. Se escribirá el tipo de datos de cada parámetro, separados por blancos, seguido de una flecha (“->”) y el tipo de datos del resultado. Después, puede escribirse uno o más casos de prueba, cada uno en una línea distinta. Cada caso de prueba se escribe con una sintaxis similar a la de la signatura: valores de los parámetros, separados por blancos, una flecha y valor óptimo del problema para dichos datos de entrada. La carga de casos de prueba preparados produce que, cuando el usuario elige alguno de los problemas para los que existen, se cargan automáticamente en la tabla de datos (véase apartado 3.1).

- Carga de restricciones. Permite cargar automáticamente los rangos de valores a usar para la generación aleatoria o la introducción interactiva de juegos de prueba. Dado que los diálogos de producción de datos recuerdan los últimos rangos usados, esta función está concebida para la experimentación con un mismo problema en varias sesiones. En caso de marcar esta casilla, OptimEx busca, en el mismo directorio donde se

encontraba la clase cargada en el editor, un fichero de texto con el mismo nombre que la clase y extensión CNSTR.

El fichero puede contener restricciones para las distintas firmas de métodos públicos de la clase. La Fig. 21 muestra un ejemplo para el problema del llenado de cajas. Puede observarse que aparecen las restricciones para dos problemas, separados por una línea con un punto y coma. El fichero comienza con las restricciones para la versión de maximización del problema y sigue con las restricciones de la versión de minimización.

Para cada problema, debe escribirse su firma en una primera línea. Se escribirá el tipo de datos de cada parámetro, separados por blancos, seguido de una flecha y el tipo de datos del resultado. Después, se escriben las restricciones de cada parámetro, cada uno en una línea distinta. Los valores a escribir para cada parámetro son los mismos que aparecen en el diálogo de generación aleatoria de datos (compárense los valores de la Fig. 21 y la Fig. 3).

```
int[] int int
10 1 7
1 10
7 20
;
int[] int
10 1 7
7 20
```

Fig. 21. Fichero de texto de carga automática, con juegos de prueba preparados.

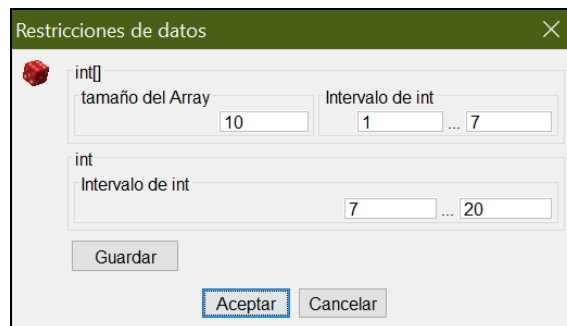


Fig. 3. Diálogo de generación aleatoria de datos con restricciones.

- “Seleccionar métodos...”. Una vez que se ha elegido un problema en OptimEx, por defecto se usan todos los métodos públicos de la clase que tengan dicha firma. Sin embargo, el usuario puede querer dejar aparte del experimento algunos métodos, por ejemplo, por estar inacabados o dar alguna clase de problemas.

La Fig. 22 muestra el diálogo cuando el criterio de experimentación es optimalidad. Puede observarse que se presentan todos los métodos con la

signatura seleccionada, de arriba abajo. Para cada método, pueden marcarse dos opciones: si se quieren comparar sus resultados y si se quiere tomar el método como referencia para las comparaciones. Por defecto, todos los métodos están seleccionados y no se considera que ninguno sea óptimo, sino que deben compararse sus resultados para determinar qué valor es óptimo para cada juego de prueba.

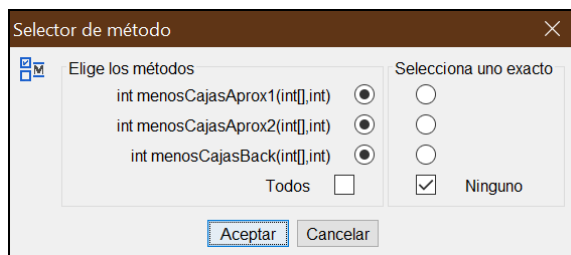


Fig. 22. Diálogo de selección de métodos.

- “Seleccionar medidas...”. Es un sencillo diálogo que permite elegir entre medidas de desviación absolutas y relativas cuando el criterio de experimentación es la optimalidad. Las medidas de desviación absolutas miden la diferencia (real o entera) entre las soluciones óptimas y las demás. Las medidas de desviación relativas miden la proporción entre las soluciones óptimas y las demás, suponiendo que una solución óptima corresponde al 100%. Por defecto, se toman medidas relativas.

La selección de medidas absolutas afecta a los resúmenes numérico y gráfico. El resumen numérico utiliza medidas de desviación absolutas (véase Fig. 23). Su nombre y definición son parecidos a los usados para medidas relativas.

- Diferencia media subóptima.
- Diferencia extrema subóptima.
- Diferencia media sobreóptima.
- Diferencia extrema sobreóptima.

Medidas	menosCajasAprox1	menosCajasAprox2	menosCajasBack
Núm. ejecuciones	100	100	100
Núm. ejec. válidas del método	100	100	100
Núm. ejec. válidas en total	100	100	100
% soluciones subóptimas	36 %	1 %	0 %
% soluciones óptimas	64 %	99 %	100 %
% soluciones sobreóptimas	0 %	0 %	0 %
Diferencia media subóptima	1,11	1	0
Diferencia máxima subóptima	2	1	0
Diferencia media sobreóptima	0	0	0
Diferencia máxima sobreóptima	0	0	0

Fig. 23. Tabla de resumen numérico del experimento con la optimalidad con medidas absolutas.

El diagrama gráfico de desviación muestra las medidas de desviación absolutas de forma similar a las relativas, con la única diferencia de que la escala de valores no mide porcentajes sino valores numéricos (véase Fig. 24).

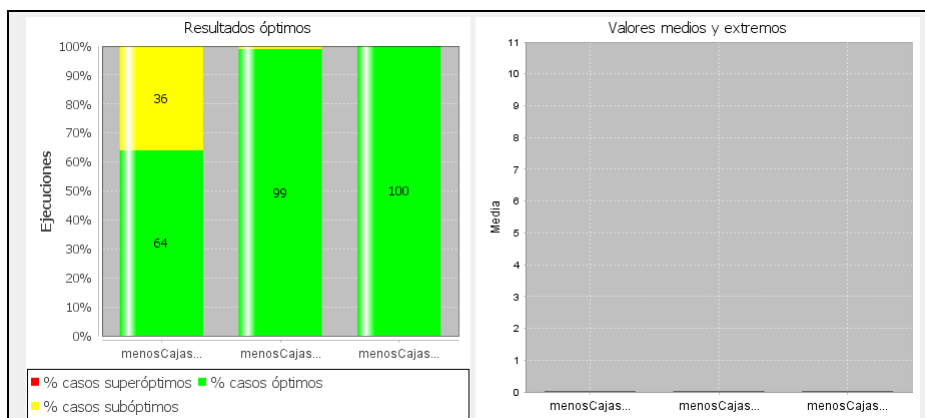


Fig. 24. Diagramas de resumen gráfico del experimento con la optimalidad con medidas absolutas.

- “Ejecución”. Produce la ejecución de los métodos sobre los juegos de prueba seleccionados. Aquellos juegos de datos que ya se ejecutaron no se ejecutan de nuevo.

Si algún caso de prueba está tardando un tiempo excesivo en terminar, OptimEx lo detecta y consulta si debe parar su ejecución. De esta forma, pueden evitarse esperas muy largas e incluso ejecuciones infinitas. Al usuario se le ofrece las opciones de continuar y de parar, aprovechando para preguntar si su respuesta se aplicará a este caso solamente o también otros casos que puedan detectarse (véase Fig. 25).

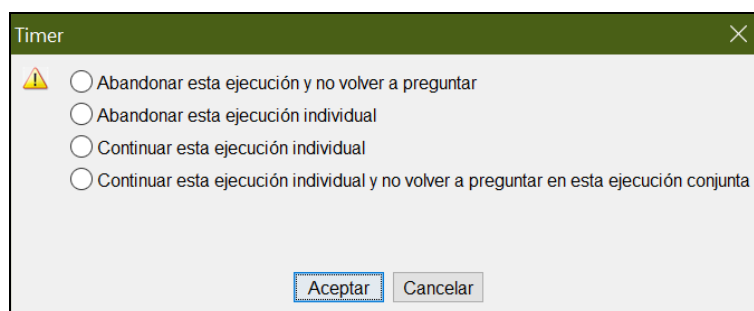


Fig. 25. Diálogo de consulta por tiempo excesivo de ejecución de un caso de prueba.

- “Parar ejecución”. Permite para la ejecución en marcha. Está concebido para aquellos casos en los que la ejecución avanza muy lenta, bien porque son muchos casos de prueba bien porque algún algoritmo tarda bastante en cada ejecución individual. La parada se realiza de forma ordenada, de forma que se hayan medido todos los algoritmos para los casos de prueba ejecutados.

4.3 Menú Tablas

El menú Tablas agrupa algunas funciones de manejo (de borrado) de tablas. Incluye 4 opciones:

- “Borrar fila en tabla de datos”. Borra la fila seleccionada de la tabla de datos. Si hay varias filas seleccionadas, borra la primera. El borrado de una fila se hace manteniendo la coherencia entre las tablas. Por tanto, también se borra la fila correspondiente de la tabla de ejecuciones y se actualizan los resúmenes numérico y gráfico.
- “Borrar fila en ejecuciones”. Borra la fila seleccionada de la tabla de ejecuciones. Sólo puede seleccionarse una fila. El borrado de una fila se hace manteniendo la coherencia entre las tablas, modificando de forma coherente la tabla de datos, la tabla de resumen numérico y la pestaña de resumen gráfico.
- “Borrar datos de sesión”. Se borra el contenido de todas las pestañas del panel de tablas.
- “Borrar resultados, mantener datos”. Se mantiene el contenido de la tabla de datos, pero se borra el contenido de las demás pestañas del panel de tablas.

4.4 Menú Configuración

El menú Configuración agrupa las funciones de configuración de la aplicación. Incluye 3 opciones:

- “Idioma...”. Selecciona el idioma de los textos usados en la interfaz de usuario. Actualmente, puede elegirse entre español e inglés.
- “Máquina virtual Java...”. Permite cambiar de máquina virtual navegando por el sistema de ficheros. Es el mismo diálogo que surge al ejecutar OptimEx por primera vez (véase Fig. 12).
- “Resumen gráfico...”. Permite elegir entre dos opciones: “Actualizar según se hacen las ejecuciones” y “Sólo actualizar al terminar”. Si está activa la primera opción, cada vez que se completa la ejecución de una fila de la tabla de ejecuciones, se actualizan los dos resúmenes. Si está activa la segunda opción, solamente se actualizan los resúmenes una vez terminada la ejecución de todos los casos de prueba seleccionados. Esta segunda opción está concebida para acelerar el proceso experimental. Por defecto, está activa la opción primera.

4.5 Menú Ayuda

El menú Ayuda agrupa las funciones de ayuda e información sobre OptimEx. Incluye 2 opciones:

- “Ayuda sobre OptimEx”. Contiene la ayuda interactiva de OptimEx, organizada de varias formas con un índice completo (véase Fig. 26).



Fig. 26. Diálogo de generación aleatoria de datos sin restricciones.

- “Sobre OptimEx”. Contiene un reconocimiento de los desarrolladores de las distintas versiones de OptimEx y la licencia de uso.

Agradecimientos. Este trabajo se ha financiado con los proyectos de investigación TIN2015-66731-C2-1-R del Ministerio de Economía y Competitividad y S2018/TCS-4307 de la Comunidad Autónoma de Madrid. Este último también está financiado con fondos estructurales (FSE y FEDER).

Referencias

1. Velázquez Iturbide, J.Á.: Design and evaluation of OptimEx, an experimentation system for optimization algorithms. En: ICT in Education - Multiple and Inclusive Perspectives, M. J. Marcelino A. J. Mendes y C. Azevedo Gomes (eds.), Springer (2016) 51-68, DOI 10.1007/978-3-319-22900-3_4
2. Velázquez-Iturbide, J.Á.: GreedEx and OptimEx: Two tools to experiment with optimization algorithms. International Journal of Engineering Education, 32, 3A (2016) 1.097-1.106
3. Velázquez Iturbide, J.Á., y Escudero Angulo, J.J.: Extending instructional support in an algorithm benchmarking system. En: Proceedings of the 21st International Symposium on Computers in Education (SIIE 2019), Gonçalo Marques, C., Pereira, I., y Pérez, D. (eds.). IEEE Xplore, 2019, 6 págs., DOI [10.1109/SIIE48397.2019.8970134](https://doi.org/10.1109/SIIE48397.2019.8970134)
4. Brassard, G., Bratley, P.: Fundamentals of Algorithmics. Englewood Cliffs, NJ: Prentice-Hall, 1996