



Proyecto de Innovación

Convocatoria 2016/2017

Proyecto nº 176

Herramienta para escoger sistemáticamente combinaciones de ejercicios con mayor capacidad formadora y/o evaluadora

Ismael Rodríguez Laguna

Facultad de Informática
Departamento de Sistemas Informáticos y Computación

1. Objetivos propuestos en la presentación del proyecto

El objetivo del proyecto ha sido desarrollar una herramienta informática que permita a cualquier docente escoger automáticamente los ejercicios o ejemplos a mostrar en clase, de forma que dichos ejercicios enfrenten a sus alumnos a sus errores de concepto de la forma más eficiente posible. Adicionalmente, la herramienta también permite escoger automáticamente las preguntas que deben incluirse en un examen para que la pérdida de puntuación que suponga cada posible error de concepto del alumno (a lo largo de todas las preguntas del examen) se acerque lo más posible a la indicada expresamente por el docente.

Conforme a este concepto, que se introdujo en la primera sección de la propuesta del proyecto, la posterior sección de objetivos de dicha propuesta definió los objetivos del proyecto de la siguiente manera (a continuación se copia el contenido de dicha sección):

El objetivo del proyecto será desarrollar una herramienta informática de código abierto que, dada la batería de casos (ejemplos o ejercicios) de la que podemos escoger los que deseamos utilizar, y los errores de concepto que cubre cada uno, busque la combinación más eficiente de ellos que cumpla cualquiera de los dos objetivos siguientes, según solicite el usuario:

(a).- permitir al alumno enfrentarse a todos los errores de concepto considerados si se le enseñan dichos ejemplos en clase, de forma que cada error de concepto posible sea cubierto al menos el número de veces correspondiente solicitado por el docente. Además, se contemplará la posibilidad de considerar requisitos adicionales indicados por el docente, tales como respetar ciertas dependencias entre ejemplos (e.g. "el ejemplo B nunca debería mostrarse si no se ha mostrado antes el ejemplo A") o preservar que la dificultad de los ejemplos escogidos esté equilibrada;

(b).- confeccionar un examen donde todos los errores de concepto habituales estén cubiertos por las preguntas planteadas (es decir, donde todos los errores de concepto supongan fallar alguna de dichas preguntas) pero, a su vez, la puntuación numérica perdida por el alumno en función de sus errores de concepto sea aproximadamente la indicada por el docente en cada caso.

Desgraciadamente, a medida que la batería de casos disponibles (ejemplos o ejercicios) crece, la tarea de escoger la mejor combinación de ellos, conforme a cualquiera de los dos objetivos anteriores, se hace mucho más difícil. Ambos problemas generalizan el problema de la "Cobertura de conjuntos", cuya NP-completitud implica que ningún algoritmo podrá resolverlo de manera "rápida" (formalmente, "en tiempo polinómico") a no ser que $P=NP$ (una hipótesis improbable). Un algoritmo que esencialmente probase exhaustivamente todas las combinaciones posibles de e.g. 100 ejemplos posibles, tardaría más tiempo que la edad del universo en dar la solución óptima con un ordenador actual típico. Más aún, dada la Log-APX-completitud de dicho problema, a no ser que $P=NP$ tampoco podrá existir un algoritmo aproximado rápido que garantice que sus soluciones sean, en el caso peor, un $X\%$ peor que las correspondientes soluciones óptimas, para cierta X constante. Por todo ello, no esperamos que utilizar técnicas algorítmicas convencionales proporcione soluciones particularmente buenas para estos problemas. Para resolverlos de manera aproximada, utilizaremos una combinación de algoritmos voraces y algoritmos evolutivos o de enjambre, concretamente algoritmos genéticos y/o optimización basada en enjambre de partículas. Dichos algoritmos no garantizan ningún grado de optimalidad mínima en el caso peor, pero es ampliamente conocida su capacidad para

obtener buenas soluciones en media para instancias típicas de muchos problemas. Por ello, esperamos que también proporcionen soluciones razonables para nuestros problemas.

2. Objetivos alcanzados

Hemos desarrollado una herramienta que permite resolver (de manera aproximada, dada su intratabilidad) los siguientes problemas:

1.- **Conjunto mínimo de problemas que alcance cierta cobertura conceptual:** Dados (a) una batería de ejercicios posibles para resolver en clase, donde para cada ejercicio se indica la duración aproximada que requiere resolverlo y los errores conceptuales a los que es sensible dicho ejercicio (es decir, tales que su resolución sería incorrecta si el método de resolución del alumno padeciera dicho error conceptual), y (b) los niveles de cobertura de los errores conceptuales que desea alcanzar el profesor con los ejercicios que resolverá en clase (e.g. el profesor desea que los ejercicios vistos cubran al menos 5 veces los errores conceptuales de tipo A, que cubran al menos 2 veces los de tipo B, etc), encontrar el conjunto de problemas cuya resolución completa requiera menor tiempo total y que cumpla, a su vez, que la cobertura de errores conceptuales lograda por los ejercicios incluidos sea, al menos, la solicitada en todos los casos (puede ser mayor).

2.- **Conjunto de problemas que maximice la cobertura dentro de un tiempo disponible dado:** Dados (a) y (b), y adicionalmente (c) el tiempo máximo que el profesor desea asignar a la resolución de problemas, encontrar el conjunto de problemas tal que su resolución en clase requiera un tiempo menor o igual que el asignado para la tarea, que alcance las coberturas deseadas para todos los tipos de error conceptual indicados y que, bajo ambas condiciones, maximice la suma de las coberturas sobrantes (es decir, la suma, para todos los errores conceptuales considerados, de la diferencia entre la cobertura de dicho error alcanzada por el conjunto de ejercicios propuesto y la cobertura solicitada).

3.- **Examen donde los errores conceptuales resten la puntuación deseada:** Dado (a) y adicionalmente (d) la proporción de ejercicios de examen que se desea que cada tipo de error conceptual provoque fallar (e.g. un alumno con el error conceptual de tipo A debería fallar un 30% de los ejercicios de examen, mientras que otro con un error de tipo B, más grave, debería fallar un 50% de los ejercicios) y (e) el tiempo máximo que la resolución del examen debe requerir, se pretende lo siguiente: encontrar el conjunto de ejercicios (examen) tal que su resolución requiera un tiempo menor que dicho tiempo máximo y que, a su vez, minimice la desviación entre la puntuación total que el profesor desea que cada alumno obtenga según sus errores conceptuales en cada caso y la puntuación que realmente se obtendría conforme a los ejercicios escogidos (es decir, que minimice la suma de las diferencias entre la proporción de preguntas del examen escogido que efectivamente se fallarían por culpa de cada error conceptual posible y la proporción de ellas que el profesor deseaba que se fallasen). En realidad, se consideran dos variantes del problema en función de la interpretación del dato de entrada (d): que este proporcione el porcentaje de ejercicios que cada error de concepto debe provocar fallar (como se indicó anteriormente), o que proporcione el número absoluto de ejercicios que debe provocar fallar. La segunda variante es apropiada si se desea una cantidad de cobertura absoluta, en lugar de simplemente una proporción.

Dada la intratabilidad computacional de los problemas anteriores, la herramienta encuentra soluciones aproximadas a los mismos utilizando algoritmos genéticos.

Como prueba de concepto y caso de estudio de la utilización de la herramienta, hemos creado una amplia batería de ejercicios de la asignatura Matemática Discreta y Lógica Matemática (primer curso de los grados en Ingeniería Informática, Ingeniería

del Software e Ingeniería de Computadores impartidos en la Facultad de Informática), clasificados tal y como se indica arriba para el dato de entrada (a): para cada problema, se identifica el tiempo que requiere aproximadamente su resolución, así como qué errores de concepto típicos en la asignatura harían que dicho problema en concreto fuera, probablemente, mal resuelto.

Basándonos en la amplia experiencia previa en la impartición de la asignatura de varios miembros del presente proyecto, hemos clasificado 261 problemas de la asignatura de dicha manera.

Utilizando como dato de entrada para nuestro algoritmo dichos datos extraídos de la docencia real, hemos ejecutado nuestra herramienta para la resolución de los problemas (1), (2) y (3) mencionados antes, con dos objetivos principales: (i) ajustar los parámetros de los algoritmos genéticos subyacentes para mejorar sus resultados al resolver instancias típicas procedentes del mundo real (es decir, para afinar y mejorar la propia herramienta) y (ii) comprobar la plausibilidad de los conjuntos de problemas y exámenes diseñados automáticamente por los mismos.

En todos los casos, los miembros del proyecto expertos en la impartición de dicha asignatura han corroborado la idoneidad a priori de los conjuntos de problemas y exámenes obtenidos. Desgraciadamente, dado que la herramienta no ha sido completamente funcional hasta el final del segundo cuatrimestre, dichos conjuntos de problemas y exámenes se obtuvieron hace poco tiempo y no han sido puestos en práctica con los alumnos de la asignatura. En particular, antes de poder haber comprobado in situ la utilidad de los conjuntos de ejercicios obtenidos en la resolución de los problemas (1) y (2), considerábamos prematura la utilización directa de los exámenes diseñados en la resolución del problema (3) durante la reciente convocatoria de junio. No obstante, esperamos poner a prueba la utilidad de dichos conjuntos de ejercicios y exámenes en la docencia de los próximos cursos.

La herramienta ha sido desarrollada en el lenguaje de programación C++ y puede descargarse de la web <http://antares.sip.ucm.es/~ismael/pimcd176>. Al ejecutarse la herramienta, esta solicita al usuario los datos del problema a resolver. Primero se proporciona un fichero con la información sobre los ejercicios disponibles. En dicho fichero, la primera línea incluye el número de problemas disponibles y el número de errores conceptuales que se analizan. A continuación, cada línea del fichero incluye información sobre cada uno de los problemas disponibles. Así, cada línea incluye el tiempo necesario para resolverlo, su nivel de dificultad y la cobertura que proporciona para cada uno de los errores conceptuales que se quieren analizar.

Además del fichero de entrada anteriormente comentado, que incluye información sobre todos los problemas disponibles para la asignatura, la herramienta solicita un segundo fichero de entrada en el que se detalla el caso concreto que se quiere resolver. Es decir, si queremos encontrar un conjunto mínimo de problemas que proporcionen una cierta cobertura, si además queremos maximizar la cobertura dentro de un tiempo máximo, o si queremos crear un examen donde cada error conceptual reste lo que establezca el profesor. Además de indicar qué tipo de optimización queremos realizar, es necesario proporcionar las coberturas (o pesos) concretos que queremos para cada posible error, así como el tiempo máximo que puede durar el examen. Asimismo, se solicita al usuario que indique durante cuántas iteraciones quiere ejecutar el algoritmo genético.

Tras aportar el usuario todos los datos de entrada, la aplicación ejecuta el correspondiente algoritmo genético, y devuelve el conjunto de problemas o examen construido expresado en un formato muy simple, en el que para cada posible problema se indica un 1 si debe incluirse o un 0 si no debe incluirse.

3. Metodología empleada en el proyecto

Hemos desarrollado nuestra herramienta siguiendo un plan basado en las siguientes etapas, conforme a lo que se indicó en la presentación del proyecto.

1.- Inicialmente, todos los miembros del proyecto asistimos a una serie de reuniones en las que diseñamos los factores que deberían ser tenidos en cuenta en los problemas a resolver. Como dijimos antes, dichos problemas dependen de los errores de concepto cubiertos por cada ejemplo posible, así como de la puntuación de examen que se desea que pierda el alumno por cada error de concepto que pudiera cometer. De esta forma, fijamos la definición de los problemas (1), (2) y (3) tal y como se especifica en la sección anterior. Entonces, comenzamos la recopilación y clasificación de ejercicios de una asignatura concreta (Matemática Discreta y Lógica Matemática, como mencionamos anteriormente), para que dicha batería de ejercicios, ya clasificados en función de su tiempo de resolución y tipos de errores conceptuales cubiertos, fuera utilizada en las instancias reales de nuestros problemas a ser resueltas por la herramienta. A este respecto, la experiencia previa de algunos miembros del proyecto en la impartición de dicha asignatura fue fundamental.

2.- Una vez que los problemas a resolver fueron plenamente modelados, y las instancias basadas en datos reales a ser resueltas por nuestra herramienta fueron construidas, se procedió a la implementación de los algoritmos que resuelven de manera aproximada dichos problemas. Como dijimos antes, para resolver nuestros tres problemas utilizamos algoritmos genéticos, los cuales forman parte del área de investigación de algunos miembros del proyecto. La implementación de la herramienta se ha desarrollado en C++.

3.- Tras finalizar la realización de la herramienta, ésta se ha hecho disponible públicamente a todos los miembros de la comunidad universitaria a través de la web <http://antares.sip.ucm.es/~ismael/pimcd176>. En dicha dirección puede encontrarse el fichero ejecutable de la aplicación, su código fuente en C++, y los ficheros de entrada a la aplicación que hemos utilizado en nuestro caso de estudio, incluidos los que representan, dentro del formato solicitado por la aplicación, los 261 ejercicios de Matemática Discreta y Lógica Matemática considerados.

4.- Además, hemos llevado a cabo algunas comprobaciones manuales para analizar la idoneidad de los conjuntos de ejemplos/ejercicios seleccionados por la herramienta, basadas en la experiencia previa de algunos miembros del proyecto en la impartición de la asignatura Matemática Discreta y Lógica Matemática. Su opinión es que los conjuntos de ejercicios y exámenes diseñados por la herramienta resultan adecuados para su utilización en clase. Desgraciadamente, como indicamos en la sección anterior, la herramienta y las instancias de nuestros problemas basadas en dicha asignatura han estado plenamente disponibles sólo al final del segundo cuatrimestre, por lo que no ha dado tiempo a comprobar la idoneidad de los conjuntos de ejercicios y exámenes diseñados por la herramienta aplicándolos en un entorno docente con alumnos reales. En los cursos siguientes deseamos comparar la capacidad de los alumnos para resolver problemas tras habersele enseñado la técnica necesaria para resolver dichos problemas de dos maneras posibles: (a) viendo con su profesor los ejemplos seleccionados por la herramienta; o bien (b) viendo ejemplos seleccionados manualmente por el docente, de la manera tradicional. Esto podrá llevarse a cabo utilizando cada una de dichas metodologías en un grupo o desdoble de problemas diferente.

4. Recursos humanos

Nuestro proyecto de innovación docente ha sido interfacultativo, al incluir miembros de la Facultad de Educación (Departamento de Didáctica de las Matemáticas) y de la Facultad de Informática (Departamento de Sistemas Informáticos y Computación).

La diferente formación y especialización de los miembros procedentes de ambas facultades ha resultado de especial utilidad para el desarrollo del proyecto. Por un lado, los miembros de la Facultad de Educación han aportado sus conocimientos a la hora de definir qué factores concretos deberían ser explícitamente representados en la definición de los tres problemas a resolver, de forma que los objetivos de dichos problemas realmente representasen los objetivos realistas de un docente conforme a la información de que habitualmente dispone (por ejemplo, no sería realista exigir al docente incluir, en la clasificación de los ejercicios, demasiadas características de cada uno, pues esto habría hecho demasiado tediosa dicha clasificación). Además, sus opiniones han sido clave a la hora de decidir cómo debían ser clasificados los problemas disponibles de la asignatura objetivo de nuestros experimentos (e.g. a qué nivel de granularidad deberíamos considerar los errores de concepto, cómo medir el esfuerzo que previsiblemente requiere la resolución de cada ejercicio, homogeneización de criterios, etc).

Por su parte, los miembros del proyecto de la Facultad de Informática se han dividido en dos grupos principales (no disjuntos), según su aportación. Por un lado, los miembros con mayor experiencia en la importación de la asignatura Matemática Discreta y Lógica Matemática han aplicado los criterios anteriormente fijados para clasificar sistemáticamente la batería de ejercicios disponible en función del tiempo de resolución y de los errores de concepto cubiertos por cada uno. También ha sido su responsabilidad evaluar, en base a su experiencia previa, si los conjuntos de problemas y exámenes obtenidos por la herramienta parecían ser apropiados para su utilización en clase.

Por otro lado, los miembros de la Facultad de Informática con mayor experiencia en el desarrollo de algoritmos evolutivos se han centrado en el desarrollo de la herramienta y de sus algoritmos subyacentes. Antes de comenzar la implementación de los algoritmos, se confirmó formalmente la intratabilidad de los problemas considerados conforme a su definición final (fijémonos en que, en sentido estricto, son variantes de los problemas originalmente considerados en nuestra propuesta original, a pesar de ser muy similares). Cabe destacar que comprobar dicha intratabilidad es necesario para que esté justificado resolverlos de manera aproximada con algoritmos subóptimos tales como e.g. algoritmos evolutivos o de enjambre. Tras dicha comprobación, y debido a las características concretas de los problemas a resolver, se decidió que el método subóptimo utilizado para resolverlos serían los algoritmos genéticos. Una vez fijados de esta forma los objetivos de la herramienta, se llevó a cabo su implementación completa, así como el posterior ajuste de los parámetros de los algoritmos genéticos utilizados para la mejor resolución de las instancias de problema consideradas. Esto podrá ayudar en el futuro a resolver mejor otras instancias construidas también a partir de ejercicios reales, ya que es previsible que dichas futuras instancias tengan, hasta cierto punto, formas similares.

5. Desarrollo de las actividades

Las actividades desarrolladas han sido las siguientes:

- Definición formal de los problemas a resolver.
- Selección de criterios para clasificar los ejercicios de Matemática Discreta y Lógica Matemática.
- Clasificación de los problemas de dicha asignatura conforme a dichos criterios.
- Comprobación formal de la intratabilidad de los problemas considerados.
- Selección de la metaheurística a utilizar para resolver dichos problemas.
- Implementación de la herramienta conforme a todos los criterios fijados.
- Ajuste de parámetros del algoritmo genético, para facilitar la resolución de otras futuras instancias con forma similar a las instancias consideradas.
- Evaluación de la idoneidad de los conjuntos de problemas y exámenes diseñados por la herramienta para las instancias consideradas.

Cada una de las actividades anteriores ha sido descrita en mayor detalle a lo largo de las dos secciones anteriores.

Finalmente, hemos comenzado a trabajar en un artículo de investigación que presentará la herramienta desarrollada, así como unos experimentos más amplios y sistemáticos donde se comprobará la idoneidad de los conjuntos de ejercicios y exámenes obtenidos.