

ENFOQUES DE OPTIMIZACIÓN MULTI-OBJETIVO BASADOS EN PREFERENCIAS EN LA INGENIERÍA DE SOFTWARE

Martín Arrúa, Luciano Bracco, Giovanni Daián Rottoli, Esteban Schab, Adrián Tournoud, Carlos Casanova, Anabella De Battista

Grupo de Investigación sobre Inteligencia Computacional e Ingeniería de Software, Departamento Ingeniería en Sistemas de Información, Fac. Reg. Concepción del Uruguay, Universidad Tecnológica Nacional
Entre Ríos, Argentina

{arruam, braccol, rottolig, schabe, tournoura, casanovac, debattistaa}@frcu.utn.edu.ar

RESUMEN

La Ingeniería de Software Basada en Búsqueda (ISBB) estudia la aplicación de técnicas de optimización metaheurística a problemas de la Ingeniería de Software (IS). Una vez que una tarea de la IS se enmarca en un problema de búsqueda existen multitud de algoritmos que pueden aplicarse para resolver ese problema. La mayoría del trabajo existente trata a los problemas de la IS desde un punto de vista mono-objetivo. Sin embargo, muchos de estos problemas poseen múltiples objetivos en conflicto que deben ser optimizados. El número de objetivos a considerar es, en general, alto (esto es, más de tres objetivos). Si bien la comunidad científica ha propuesto varios enfoques de solución para atacar la optimización multi-objetivo, muchos de estos enfoques nos se han aplicado aún en la ISBB. Uno de estos enfoques es el llamado “basado en preferencias”, el cual permite incorporar las preferencias entre los objetivos del tomador de decisiones, restringiendo el frente Pareto-óptimo a una zona de interés específica, facilitando de esta manera la tarea de tomar una decisión.

Palabras clave: Ingeniería de Software Basada en Búsqueda, Optimización Multi-objetivo, Metaheurísticas, Hiperheurísticas, Preferencias.

CONTEXTO

El presente trabajo se desarrolla en el ámbito del Grupo de Investigación sobre Inteligencia Computacional e Ingeniería de Software (GIICIS), perteneciente al Departamento Ingeniería en Sistemas de Información de la

Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay.

1. INTRODUCCIÓN

Muchos de los problemas asociados con el manejo de la complejidad del proceso de desarrollo de software han probado ser resistentes a soluciones analíticas convencionales. El proceso de desarrollo de software y sus productos tienden a estar caracterizados por un gran número de restricciones y objetivos interrelacionados que compiten entre sí. Algunos de ellos están claramente especificados, mientras que otros son notablemente vagos y se encuentran pobremente definidos. Cambios en un parámetro con frecuencia tienen un gran impacto sobre áreas relacionadas, haciendo que el balance de intereses sea muy difícil de lograr. Estas características producen que las soluciones “perfectas” sean, si no imposibles, al menos imprácticas.

Si sólo hubiera una única solución para un conjunto típico de restricciones de un problema de la Ingeniería de Software, entonces probablemente no sería llamada “ingeniería”. De la misma manera que ocurre con otras disciplinas, la Ingeniería de Software generalmente se ocupa de soluciones cuasi-óptimas, o aquellas que caen dentro de una tolerancia aceptable. Es así que los ingenieros de software se enfrentan a problemas que consisten, no en encontrar *la solución*, sino en construir una solución aceptable, o cuasi-óptima, partiendo de una gran cantidad de alternativas. Con frecuencia no está del todo claro cómo alcanzar una solución óptima, pero puede diseñarse una forma de evaluar y

comparar candidatos. Por caso, puede ser difícil conocer cómo alcanzar un diseño con alta cohesión y bajo acoplamiento, pero es relativamente sencillo decidir cuándo un diseño está más acoplado que otro.

La siguiente es una lista ilustrativa de problemas de optimización de la ingeniería de software:

- ¿Cuál es el mínimo conjunto de casos de test que cubren todas las ramas de un programa?
- ¿Cuál es la mejor manera de estructurar la arquitectura de este sistema?
- ¿Cuál es el conjunto de requerimientos con el mejor balance entre costo de desarrollo y satisfacción de los clientes?
- ¿Cuál es la mejor asignación de recursos para este proyecto de desarrollo de software?
- ¿Cuál es la mejor secuencia de pasos de refactorización a aplicar en este sistema?

Las propiedades descriptas sobre la Ingeniería de Software son precisamente los atributos que hacen a la aplicación de técnicas basadas en búsqueda tan atractiva. Las técnicas de búsqueda metaheurística son un conjunto de algoritmos genéricos adecuados para la búsqueda de soluciones óptimas o cuasi-óptimas en problemas con un enorme espacio de búsqueda multimodal [12]. La aplicación de técnicas de búsqueda a problemas de la ingeniería de software recibe en el estado del arte el nombre de Ingeniería de Software Basada en Búsqueda (*Search-Based Software Engineering*, SBSE) [1].

Los algoritmos metaheurísticos brindan a sus usuarios estrategias de búsqueda genéricas, a la vez que requieren, para ser aplicados a un problema en específico, las siguientes definiciones (según [1]): (a) la forma de representación de las soluciones del problema; (b) una función de evaluación, definida para tal representación, que mide la calidad de las soluciones, y (c) un conjunto de operadores para manipular esas soluciones.

Es de destacar que, para resolver un problema mediante metaheurísticas, no es necesario realizar un modelado algebraico analítico de la misma manera que se realiza, por caso, en los enfoques basados en programación matemática. La función de evaluación puede estar definida, por caso, mediante un algoritmo, o utilizando varios algoritmos para su cómputo. Esta es una ventaja muy importante, mayormente en lo que se refiere a expresividad de los modelos.

Según lo expuesto hasta el momento, los problemas requieren soluciones con ciertos atributos o propiedades que pueden ser expresados como funciones que mapean posibles soluciones a valores numéricos escalares. Un enfoque para la optimización comúnmente usado es tomar esos atributos como restricciones para determinar la factibilidad de una solución, mientras que uno de ellos es elegido como función objetivo, la cual determina el orden (total) de preferencia de las soluciones factibles. Este tipo de problemas de optimización se refieren como mono-objetivo.

Un enfoque más general es la optimización multi-objetivo, donde varios atributos se emplean como funciones objetivo y se usan para definir un orden de preferencia parcial de las soluciones factibles. Por caso, en el problema nombrado más arriba sobre encontrar el subconjunto de requerimientos, se desea minimizar el costo de desarrollo y maximizar la satisfacción de los clientes. Estos objetivos, al igual que en muchos otros problemas, son de cierta forma contradictorios y compiten entre sí, definiendo sobre el espacio de soluciones un orden parcial, la relación de dominancia, donde existen pares de soluciones que no son comparables a priori. Ante esta situación pueden tomarse básicamente dos estrategias para la resolución del problema de optimización multi-objetivo:

- (a) buscar el frente de Pareto completo, compuesto por todas las soluciones no dominadas y dejar al tomador de decisiones la tarea de buscar en ese frente la solución que mejor se adecúe a su criterio (por caso, mediante otro algoritmo), o bien

- (b) utilizar algún mecanismo (por caso, una nueva relación de orden total) que reduzca el frente de Pareto a una única solución o a una zona de interés.

El enfoque descrito en el punto (a) ha sido estudiado en trabajos como [2], [3]. Una desventaja importante de este tipo de enfoques es que la performance de los algoritmos baja rápidamente cuando aumenta el número de objetivos. Se estima que para una cantidad de 4 o más objetivos este tipo de enfoques presenta algunos inconvenientes difíciles de mitigar, como frentes Pareto-óptimos muy grandes, alta dimensionalidad que dificulta encontrar buenas direcciones en la búsqueda, e incapacidad de la relación de dominancia para distinguir entre vectores objetivo [4].

Por otra parte, un mecanismo del tipo de los descritos en el punto (b) es utilizar información concerniente a las preferencias del tomador de decisiones entre los objetivos, esto es, establecer la importancia relativa de los objetivos. Los algoritmos que incorporan las preferencias, intuición o emoción en el proceso de optimización reciben el nombre de algoritmos basados en preferencias (ABP). Tanto la resolución de problemas multi-objetivo como la utilización de preferencias en el contexto de la ISBB han sido identificados en *reviews* recientes como desafíos y áreas emergentes [5]–[7].

Además de las tres definiciones que ya se han descrito como requeridas para la aplicación de técnicas basadas en búsqueda, se requiere un cuarto mecanismo para aplicar en este contexto un ABP: una forma de incorporar la información del tomador de decisiones relativa a las preferencias [7]. A grandes rasgos pueden identificarse tres tipos principales según el momento del proceso de optimización donde se incorporan las preferencias: a priori (antes de comenzar el proceso), interactivos (durante) y a posteriori (después). De ninguna manera estos enfoques son mutuamente excluyentes, pudiendo ser combinados de diversas maneras. Ejemplos de ABP utilizados en la ISBB son los algoritmos genéticos, siendo estos los más utilizados por un amplio margen, en particular el Algoritmo Genético Interactivo (IGA, de sus siglas en inglés), utilizado en varios trabajos.

También pueden encontrarse aplicaciones con ACO (*Ant Colony Optimization*), NSGA-II (*Nondominated Sorting Genetic Algorithm*), VEGA (*Vector Evaluated Genetic Algorithm*), IBEA (*Indicator Based Evolutionary Algorithm*), HCM (*Hard C-Mean*) y DE (*Differential Evolution*).

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

2.1. Estudio de ABPs en distintos problemas de la IS

La ISBB es una disciplina relativamente nueva. Si bien ha cobrado mucha relevancia en los últimos años [7], todavía escasean los estudios empíricos sobre la habilidad exhibida por distintos métodos para resolver los distintos problemas de optimización derivados de la IS. Los ABP no son la excepción a esta regla. De modo que el trabajo principal en esta línea es identificar los ABP presentes en el estado del arte y aplicarlos a aquellos problemas de la ISBB en los cuales no se tiene registro de haberse aplicado hasta el momento. Claramente, el objetivo es realizar comparaciones respecto de diversos criterios: tiempo, calidad, robustez, facilidad de uso, nivel de generalidad, etc. Una dimensión más del estudio es analizar cómo cambia el comportamiento de cada ABP según se realice la incorporación de la información relativa a las preferencias de manera a priori, interactiva o a posteriori. Ejemplos ABP en otros dominios que no han sido aplicados aún pueden encontrarse en [8]–[10].

2.2. Diseño y desarrollo de nuevos algoritmos

Los algoritmos metaheurísticos disponibles en el estado del arte pueden no ser adecuados para la resolución de ciertas clases de problemas. Por caso, la metaheurística PSO original (*Particle Swarm Optimization*) supone un espacio de búsqueda continuo, y codifica las soluciones en vectores. Esta representación no resulta adecuada para resolver problemas de optimización combinatoria, como TSP (*Traveling Salesman Problem*), donde las soluciones son permutaciones de un conjunto.

Es así que las metaheurísticas pueden reformularse para atacar distintos problemas. Para el caso de PSO se cuenta con el *template* llamado PSO Canónico descrito en [11], el cual constituye una generalización de PSO que puede especificarse para resolver distintos problemas. Es así que para problemas específicos de la ISBB se trabaja en lograr algoritmos específicos novedosos que mejoren a los disponibles en algún aspecto. Otro tipo de algoritmos específicos en los que se trabaja es en las hiperheurísticas, las cuales constituyen métodos de un nivel de abstracción superior que buscan combinar distintas estrategias heurísticas o metaheurísticas para la resolución de un problema o una clase de problemas. Este tipo de algoritmos aún no ha sido aplicado ampliamente en la ISBB.

2.3. Mecanismos de captación de preferencias

Como ya se dijo, los enfoques de optimización multi-objetivo basados en preferencias suponen que puede incluirse información del tomador de decisiones para guiar la búsqueda hacia regiones prometedoras del espacio de soluciones. Idear mecanismos adecuados para captar esta información no es una tarea trivial, y se trabaja en el estudio de los mecanismos disponibles, y el diseño e implementación de nuevos mecanismos para mejorar principalmente la usabilidad de los métodos. Los componentes analizables de estos mecanismos son principalmente tres: la cantidad de interacciones con el usuario, qué información se le solicita, y en qué momento del proceso de optimización [7].

3. RESULTADOS OBTENIDOS/ESPERADOS

Se espera obtener una plataforma de pruebas homogénea para realizar comparaciones entre distintos ABP, utilizados con distintos esquemas de incorporación de preferencias. Para ello no se descarta la codificación o recodificación de algunos de los algoritmos en un lenguaje de programación de alta performance, como C++.

También se espera proponer mejoras en distintos ABP existentes, así como la adecuación de aquellos que no se encuentran adaptados para los problemas de la ISBB basada en preferencias.

Finalmente, se espera lograr una herramienta amigable y flexible que permita captar la información del tomador de decisiones de diversas maneras.

4. FORMACIÓN DE RECURSOS HUMANOS

Las líneas de investigación presentadas constituyen las líneas fundacionales de un nuevo grupo de investigación dentro de la UTN-FRCU, el GIICIS. Un investigador se encuentra realizando su tesis de doctorado. Un investigador está desarrollando su trabajo de especialización. Además participan en el proyecto tres becarios alumnos de la carrera Ingeniería en Sistemas de Información que inician su formación en la investigación. Finalmente, uno de los investigadores finalizó su doctorado en el mes de febrero de 2018 en la temática de métodos hiperheurísticos.

5. BIBLIOGRAFÍA

- [1] M. Harman y B. F. Jones, «Search-based software engineering», *Inf. Softw. Technol.*, vol. 43, n.º 14, pp. 833-839, 2001.
- [2] A. S. Sayyad, T. Menzies, y H. Ammar, «On the value of user preferences in search-based software engineering: A case study in software product lines», *2013 35th Int. Conf. Softw. Eng.*, pp. 492-501, 2013.
- [3] S. Z. Qasim y M. A. Ismail, «Research problems in Search-Based Software Engineering for many-objective optimization», en *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, 2017, pp. 1-6.

- [4] M. W. Mkaouer, M. Kessentini, S. Bechikh, y D. R. Tauritz, «Preference-based multi-objective software modelling», en *2013 1st International Workshop on Combining Modelling and Search-Based Software Engineering, CMSBSE 2013 - Proceedings*, 2013, pp. 61-66.
- [5] M. Harman, S. A. Mansouri, y Y. Zhang, «Search Based Software Engineering: Trends, Techniques and Applications», *ACM Comput. Surv.*, vol. 45, n.º 1, p. 11, 2012.
- [6] M. Harman, S. Mansouri, y Y. Zhang, «Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications», p. 78, 2009.
- [7] T. N. Ferreira, S. R. Vergilio, y J. T. de Souza, «Incorporating user preferences in search-based software engineering: A systematic mapping study», *Inf. Softw. Technol.*, vol. 90, pp. 55-69, 2017.
- [8] R. Malek, «An agent-based hyper-heuristic approach to combinatorial optimization problems», *Proc. - 2010 IEEE Int. Conf. Intell. Comput. Intell. Syst. ICIS 2010*, vol. 3, pp. 428-434, 2010.
- [9] G. Schweickardt, J. M. G. Alvarez, y C. Casanova, «Metaheuristics approaches to solve combinatorial optimization problems in distribution power systems. An application to Phase Balancing in low voltage three-phase networks», *Int. J. Electr. Power Energy Syst.*, vol. 76, pp. 1-10, 2016.
- [10] C. Casanova y G. Schweickardt, «Análisis de las habilidades de Metaheurísticas X-PSO multi-objetivo mediante Indicadores de Inteligencia de Grupo: Aplicación en el Balance de Carga en Redes Eléctricas de Baja Tensión», *Rev. la Esc. Perfec. en Investig. Oper.*, n.º 40, pp. 23-42, 2016.
- [11] M. Clerc, *Particle Swarm Optimization*. 2006.
- [12] F. Glover y G.A. Kochenberger. *Handbook of Metaheuristics*. 2003.