Optimización multiobjetivo con funciones de alto costo computacional. Revisión del estado del arte

Multiobjective optimization with expensive functions. Survey on the state of the art

Cindy Calderón-Arce[1], Pablo Alvarado-Moya[2]

Fecha de recepción: 11 de junio del 2015
Fecha de aprobación: 17 de setiembre del 2015

Calderón-Arce, C; Alvarado-Moya, P. Optimización multiobjetivo con funciones de alto costo computacional. Revisión del estado del arte. *Tecnología en Marcha*. Edición especial. Matemática Aplicada, Mayo 2016. Pág 16-24.

1 Correo electrónico: ccalderon@itcr.ac.cr. Escuela de Matemática, Instituto Tecnológico de Costa Rica.
2 Correo electrónico: palvaradomoya@gmail.com. Escuela de Electrónica, Instituto Tecnológico de Costa Rica.

Palabras clave

Optimización multiobjetivo; costo computacional; algoritmos evolutivos; aproximaciones; modelos gaussianos; superficies de seudorespuesta.

Resumen

La optimización multiobjetivo es un proceso complejo, más aún cuando las funciones objetivo que definen los problemas no están bien condicionadas o no cumplen con los requisitos mínimos para garantizar la convergencia de algoritmos clásicos, como convexidad, continuidad y diferenciabilidad. La literatura, entonces, se enfoca en el estudio de técnicas de optimización para problemas definidos por funciones con características particulares, por ejemplo, que el costo de su evaluación sea elevado, no convexas o no diferenciables. Este artículo hace una revisión general de las técnicas predominantes en problemas este tipo de funciones.

Keywords

Multi-objective optimization; computational cost; evolutionary algorithm; aproximations; gaussian models; pseudo response surface.

Abstrac

The multi-objective optimization is a complex process, even more when the functions that define the problems are not well conditioned or do not meet the minimum set requirements to ensure the convergence of classical algorithms, such as convexity, continuity and differentiability. Hence, the technical literature focuses on optimization techniques for problems defined by functions with specific characteristics, like high evaluation cost, non-convexity or non-differentiability. This article provides an overview of some of the prevailing techniques for problems with these kind of functions.

Introducción

La complejidad de las sociedades modernas y el progreso científico mundial han dado lugar al desarrollo y evolución de herramientas que contribuyan con el quehacer diario. El diseño, la implementación y la construcción de dichos instrumentos involucra elementos que envuelven la solución de problemas, basados en sistemas o modelos complejos y multifactoriales.

Esto ha llevado a la implementación de técnicas que optimizan la producción y eficiencia de las herramientas, como lo son los algoritmos creados para la simulación de circuitos y ejecución de cadenas completas de procesamiento de señales, los cuales evidencian la optimización de funciones objetivo con un número considerable de parámetros (Pereira et al., 2007).

En una subclase de problemas de ciencia e ingeniería, el costo de evaluar las funciones que los describen es alto, lo que provoca procesos de optimización en órdenes inaceptables para el desarrollo de los proyectos que los requieren y, al ampliar las dimensiones del espacio de búsqueda en el mundo multiobjetivo, ese costo se eleva exponencialmente.

Durante décadas se han utilizado métodos de optimización analíticos pero, para los problemas de interés en este artículo, la cantidad de variables que se deben tomar en cuenta y la naturaleza matemática de los modelos obligan a realizar simplificaciones que permitan la aplicación de dichos métodos (Alexandrov & Lewis, 2000; Mosat, 2006), lo que a su vez los hace poco fiables.

En la búsqueda de estrategias que soporten la optimización simultánea de más de un criterio de costo o aptitud surge la optimización multiobjeto (Deb, 2001). Esta optimiza objetivos de manera conjunta, poniéndolos a competir entre sí, de modo que para cada valor de una función de aptitud o costo se obtiene lo mejor de los otros objetivos. Así, la solución del problema general es una combinación de las soluciones óptimas de cada objetivo, por lo que dicha solución no es única (Deb, 2001; Marler & Arora, 2004).

En particular, la comunidad científica que trabaja en la implementación de algoritmos para la resolución de problemas de optimización multiobjetivo se divide en dos categorías, definidas por el tipo de funciones objetivo utilizadas. El primer grupo supone funciones objetivo de bajo costo computacional, se preocupa por la eficiencia global de los algoritmos de optimización en sí. Por otro lado, el segundo grupo asume un alto costo computacional de las funciones objetivo, por lo que se interesa por un proceso de optimización que reduzca el número de evaluaciones de dicha función, pues esta última domina la duración total de la optimización.

Este artículo hace una revisión del estado del arte orientado a los problemas de interés para el segundo grupo mencionado, es decir, enfocado en los problemas de optimización multiobjetivo con funciones objetivo costosas. En la siguiente sección se presenta la definición de optimización multiobjetivo y en las secciones posteriores se describen y analizan los algoritmos en tendencia propuestos para los problemas.

Optimización multiobjetivo

En la optimización multiobjetivo, cada una de las funciones objetivo representa una aptitud o un costo a optimizar, las cuales dependen de *n* parámetros. Sin pérdida de generalidad, en el resto del artículo se mencionan problemas de optimización de aptitudes dejando claro que los principios y resultados también son validos para problemas de optimización de costos. En particular, la r-ésima aptitud está dada por una función $f_r: \mathbb{R}^n \rightarrow \mathbb{R}$ y la función evaluadora del problema a optimizar está compuesta por dichas funciones de aptitud. Suponiendo que se deben optimizar *m* aptitudes, la función a optimizar es de la forma $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ y el problema general está dado por:

$$\max_x \; F(x) = \left(f_1(x),\, f_2(x),\, f_3(x), \ldots, f_m(x)\right)$$

sujeto a P_i, con i=1,2,3,...,k

donde $x \in \mathbb{R}^n$ y la *i*-ésima restricción está dada por la proposición P_i. En la búsqueda de la solución óptima del problema general, se trabaja sobre un espacio de posibles soluciones en $\mathbb{R}^n$, donde todas satisfacen un criterio multiobjetivo de optimalidad.

En otras palabras, el objetivo es optimizar de manera simultánea todas las funciones objetivo, puesto que en la mayoría de los problemas no es posible encontrar un solo valor óptimo para todas ellas, ya que el óptimo de una función no necesariamente lo es para las demás (Zitzler et al., 2003).

Para comparar dos elementos del espacio solución se establece el concepto de dominancia de Pareto como un orden parcial sobre elementos en R^n , en el que una solución x_i domina a otra solución x_j si se cumplen las siguientes dos condiciones:

- x_i no es peor que x_j en todas las aptitudes

$$(\forall\, r)\left[f_r\,(x_i) \geq f_r\,(x_j)\right]$$

- x_i es mejor que x_j en al menos una de las aptitudes

$$(\exists\, r)\left[f_r\,(x_i) > f_r\,(x_j)\right]$$

y se denota con $x_i \succ x_j$. Si alguna de las condiciones anteriores no se cumple, se dice que x_i no domina a x_j.

Una solución es Pareto óptima si y solo si no es dominada por ninguna otra solución, en ese caso todas las soluciones Pareto óptimas forman el conjunto de soluciones óptimas de Pareto denotado por $S^* = \left\{x \,/\, (\neg\, \exists\, x_i)\left[x_i \succ x_j\right]\right\}$ y al conjunto $F^* = \left\{F\left(x^*\right) / \, x^* \in S^*\right\}$ se le llama Frente de Pareto (Parmee et al., 2000; Ngatchou, 2008).

Métodos clásicos adaptados

Estos métodos se derivan del desarrollo en series de Taylor de la función objetivo y se clasifican de acuerdo con la cantidad de términos que utilizan de dicha serie. Por ejemplo, el Steepest Descent (Gradient Descent) usa una aproximación de primer orden y el método de Newton una aproximación de segundo orden; ambos métodos realizan la búsqueda del óptimo por medio de direcciones de descenso, lo que acelera la convergencia a la solución si dicha aproximación es lo suficientemente cercana a la función real (Fliege & Svaiter, 2008).

Usualmente, los métodos clásicos se usan para resolver problemas uniobjetivo; sin embargo, existen adaptaciones para utilizarlos en problemas multiobjetivo. Por ejemplo, Fliege et al. (2006) adaptaron el método de Newton a problemas de optimización multiobjetivo con funciones convexas, formando el conjunto de soluciones óptimas de Pareto a partir de puntos llamados puntos críticos o estacionarios (Ortega & Rheinboldt, 2000), donde $x \in \mathbb{R}^n$ es un punto crítico si $(\forall\, s \in \mathbb{R}^n)(\exists r_0)\left[\nabla f_{r_0}\,(x)^T s \geq 0\right]$.

A pesar de su exactitud, este tipo de métodos tiene la desventaja de que su comportamiento y convergencia dependen del tipo de función con la que trabajen. Por ejemplo, se requiere que la función sea derivable y convexa, condiciones no siempre garantizables en problemas de ingeniería.

Además, considerando que se trabaja con funciones cuya evaluación tiene un alto costo computacional y que la evaluación numérica de sus derivadas utiliza al menos dos llamadas de esa función, el costo total para evaluar dichas derivadas y gradientes es aún más alto; de manera que este tipo de métodos no es adecuado para esa clase de problemas.

Métodos heurísticos de optimización

Las heurísticas o métodos heurísticos son algoritmos construidos de manera intuitiva. Algunos autores denominan metaheurística a aquellos casos en los que el algoritmo de optimización utiliza a su vez subprocesos heurísticos, pero esta es una distinción no generalizada (Schweickardt, 2009).

Entre las heurísticas más populares están Colonia de hormigas (AC: Ant Colony), Recocido Simulado (SA: Simulated Annealing), Búsqueda Tabú (TS: Tabu Search), Enjambre de Partículas (PS: Particle Swarm) y los algoritmos evolutivos (EA: Evolutionary Algorithm). Los EA se inspiran en conceptos de la evolución biológica tales como la reproducción, mutación, recombinación y selección (Vasunhara et al., 2015); además, balancean la exploración y explotación del espacio de búsqueda de manera exhaustiva con pocas evaluaciones. En contraste, AC, SA y TS son los métodos más antiguos y tienden a realizar más llamadas a la función evaluadora (Eskandari & Geiger, 2008).

Los algoritmos genéticos (GA: Genetic Algorithm) son parte de los algoritmos evolutivos y utilizan técnicas aleatorias y estocásticas para simular la evolución por medio de la selección de individuos con características prometedoras (Gen & Cheng, 2000; Reeves & Rowe, 2003). Non-dominated Sorting in Genetic Algorithm II (NSGA-II) y Strength Pareto Evolutionary Algorithm 2 (SPEA2) son dos de los algoritmos más utilizados en los EA. NSGA-II es un algoritmo genético que implementa una técnica de capas o rangos para clasificar en distintos niveles de dominancia a la población completa. Además, utiliza torneos binarios de selección con reemplazo y una distancia de aglomeración para conservar la diversidad en los elementos de la población (Chen & Chiang, 2014). SPEA2 es un algoritmo evolutivo que, a partir de un valor de bondad, conserva en una población o archivo externo a los mejores individuos para tomar de allí a los padres de los descendientes. Para la selección de padres utiliza un procedimiento de truncamiento basado en un torneo binario y en las distancias entre los individuos del archivo externo (Zitzler et al., 2001; Dastfan et al., 2014).

Por otro lado, los métodos híbridos combinan las heurísticas con los métodos clásicos para acelerar el proceso de búsqueda. En la mayoría de los casos, estos métodos están adaptados a problemas particulares que hacen que la combinación trabaje de manera adecuada.

S-Metric Selection Evolutionary Multiobjective Algorithm (SMS-EMOA) es un algoritmo que combina los GA con métodos como Newton y Gradiente Conjugado para realizar una búsqueda local (Koch et al., 2009). Por medio de una mutación polinomial y un cruzamiento binario simulado, SMS-EMOA genera una cantidad fija de descendientes, analiza probabilísticamente cada objetivo de manera independiente y, con base en los resultados, realiza búsquedas locales en algún objetivo específico por medio de alguno de los métodos clásicos.

Gutiérrez-Méndez (2011) toma en cuenta el comportamiento social para la selección de los individuos a los que se les aplicarán los operadores genéticos, de manera similar a PS. También, como parte de la búsqueda de óptimos globales, aplica búsquedas locales en ciertos sectores del espacio de búsqueda o en individuos con características particulares previamente definidas para refinar la búsqueda con candidatos prometedores. Las búsquedas locales en general se realizan por medio de procesos iterativos o algún método clásico.

El éxito de las heurísticas se debe a que encuentran soluciones satisfactorias en la mayoría de los problemas. Sin embargo, requieren de muchas evaluaciones para garantizar buenas soluciones globales como, por ejemplo, en problemas de 10 dimensiones se requieren alrededor de 1000 evaluaciones (Müller & Shoemaker, 2014), un aspecto que contribuye al aumento del costo computacional del proceso de optimización con funciones objetivo caras.

Modelos gaussianos estocásticos

Autores como Ong et al. (2003) y Zhang et al. (2009) mencionan que los modelos basados en procesos gaussianos estocásticos se consideran los más eficientes para tratar problemas caros de optimización uniobjetivo. Estos asumen que la función objetivo es un proceso gaussiano estocástico, donde los valores de la función objetivo para un punto nuevo se pueden estimar a través de los datos obtenidos en búsquedas anteriores y un factor de calidad, como la mejora esperada y la probabilidad de mejora, lo cual guía la decisión de cuáles puntos evaluar.

Efficient Global Optimization (EGO) es un conjunto de algoritmos diseñados para problemas uniobjetivo que buscan soluciones de tal manera que se maximice la mejora esperada al menor costo posible, encontrando respuestas con alto potencial para mejorar el costo mínimo con un valor de predicción bajo. Así, en la búsqueda de algoritmos evolutivos para problemas multiobjetivo que reduzcan el costo computacional, surgen los llamados ParEGO como una extensión de EGO (Zhang et al., 2009). ParEGO predice el siguiente punto a evaluar a partir

de un subconjunto del espacio de búsqueda; sin embargo, no hace distinción entre un óptimo local y uno global.

ParEGO genera un número de soluciones en un hipercubo y, a partir de un modelo gaussiano, encuentra la probabilidad de que una solución particular mejore la solución general del problema, maximizando dicha probabilidad por medio del método Down Hill Simplex (Knowles & Hughes, 2005). La principal idea es recolectar datos de los candidatos previamente evaluados que puedan ser utilizados durante el proceso de evolución para construir y refinar el modelo de aproximación y a través de ellos evitar las evaluaciones de los candidatos menos prometedores. Las funciones serán evaluadas solo en los miembros más prometedores de la población, disminuyendo considerablemente el costo computacional (Ong, 2003). ParEGO requiere de una cantidad mínima de evaluaciones de la función original (cientos), de lo contrario, no converge al frente de Pareto real.

Eskandari y Geiger (2008) implementaron un algoritmo para tratar problemas con funciones de alto costo computacional llamado Fast Pareto Genetic Algorithm (FastPGA), que usa una técnica de clasificación basada en la información acerca de la dominancia de Pareto a través de soluciones y relaciones de nichos. Su desarrollo es similar al de NSGA-II e incluye algunas de las técnicas de ParEGO. Por medio de NSGA-II, optimiza las mejoras esperadas de todos los objetivos de manera independiente y así localiza el siguiente candidato a ser evaluado, sin utilizar toda la información dada por el modelo gaussiano estocástico, lo cual reduce la eficiencia (Deb et al., 2002).

Por otro lado, Multiobjective Evolutionary Algorithm/Descomposition (MOEA/D) descompone el problema en varios subproblemas uniobjetivo, obtenidos a partir de escalarizaciones del problema original definidas por una combinación lineal de las funciones que conforman la función objetivo general. Los subproblemas se relacionan entre sí por medio de las distancias entre los vectores peso de cada combinación lineal, donde dos subproblemas cercanos deben generar soluciones similares (Zhang & Li, 2007).

Así, Multiobjective Evolutionary Algorithm Descomposition with Gaussian Process Model (MOEA/D-EGO) combina MOEA/D con modelos gaussianos estocásticos. En cada iteración se construye un modelo gaussiano para cada subproblema a partir de los datos obtenidos en las búsquedas anteriores y las mejoras esperadas son optimizadas simultáneamente. Por medio de MOEA/D, este método genera nuevos candidatos y se usa un modelo basado en subconjuntos para mejorar la calidad de la predicción sin aumentar el costo computacional. En cada iteración se construye una distribución de predicción para cada objetivo de manera individual (Zhang et al., 2009).

Aproximación de funciones

Recientemente se han propuesto técnicas de optimización para funciones costosas basadas en modelos sustitutos que aproximan a la función original. En vez de usar modelos computacionales costosos durante la optimización, se utilizan métodos que trabajan con una réplica de la función reduciendo la cantidad de evaluaciones de la función original a cientos en vez de miles, lo que produce resultados con un alto grado de exactitud a un costo menor (Müller & Shoemaker, 2014).

Las primeras propuestas de aproximación construían metamodelos que actuaban sobre la mayor parte del espacio de búsqueda, trabajando con un modelo para cada objetivo, pero eso requiere de evaluaciones en todo el espacio de búsqueda, cuando en realidad lo que interesa son los puntos cerca del frente de Pareto y no los puntos en el interior (Voutchkov & Keane,

2010). Debido a esto, ahora se trabaja con modelos localizados, eficientes en regiones de interés y no en el espacio completo.

Para minimizar la cantidad de evaluaciones que se le realizan a la función objetivo original, se usan modelos de aproximación en los que se toma una muestra de puntos y su respectiva evaluación en la función original, a partir de los cuales se crea una función sustituta que se utiliza en la mayoría de las evaluaciones para guiar el proceso. La función original se evalúa únicamente para actualizar los datos de la muestra y así el modelo utilizado en cada iteración (Haanpää, 2012).

Las superficies de seudorespuesta (PRS: Pseudo Response Surface) o métodos de superficie de respuesta (RSM: Response Surface Methods) evalúan solo puntos cercanos a la región Pareto óptima realizando un mínimo de evaluaciones a las funciones objetivo originales, pero para ello se requiere una estrategia de muestreo predeterminada. RSM utiliza modelos de predicción basados en redes neuronales para aproximar la superficie y predecir cuál o cuáles puntos es mejor evaluar (Messac & Mullur, 2008).

Los modelos más utilizados en la construcción de funciones sustitutas son interpolación, regresión, funciones de base radial (RBF: Radial Basis Functions) y funciones krigeanas (KF: Kriging Function). Sobresalen, entre estos, RBF y KF, porque el error es mínimo cerca de los puntos de muestra, requieren pocas muestras y funcionan adecuadamente en problemas con dimensión mayor a 25 (Müller & Shoemaker, 2014). Sin embargo, con el aumento en la cantidad de datos y variables del problema se incrementa la cantidad de evaluaciones y, en consecuencia, también el costo computacional (Voutchkov & Keane, 2010).

La eficiencia de estos métodos depende directamente de la muestra tomada. En algunos casos, la solución podría aproximarse a mínimos locales y en otros a óptimos globales. Es así como modelos del tipo Local Optima through Metamodels (LOOM) buscan todos los óptimos locales en un número reducido de llamadas a la función objetivo real, con base en repetidas búsquedas locales por medio de un metamodelo de esta. Dado que la mayor dificultad en los problemas de optimización es el costo de evaluar el desarrollo de cada solución en un espacio de búsqueda grande, LOOM trabaja de manera esparcida respecto al número de llamadas a la función objetivo, gracias al uso del metamodelo (Rivieri et al., 2013).

Conclusiones

Los algoritmos evolutivos son los más predominantes en los métodos diseñados para optimizar problemas con funciones objetivo de alto costo computacional. Además, si las condiciones de un problema concreto lo permiten, es usual combinar dichos algoritmos con métodos clásicos e iterativos para refinar las búsquedas por medio de búsquedas locales intercaladas dentro del proceso de optimización.

La tendencia apunta al uso de modelos gaussianos y métodos de aproximación para disminuir la cantidad de llamadas a funciones costosas, utilizando aproximaciones de las mismas en la mayoría de las evaluaciones y de esa manera disminuir considerablemente el costo computacional del proceso de optimización con ese tipo de funciones.

Es claro que el éxito en el uso de superficies de seudorespuesta depende de los puntos o muestras que se utilicen para generar la aproximación, por lo que aún queda la interrogante de cómo seleccionar dichos puntos para generar una superficie suficientemente adecuada que se acerque al frente de Pareto con un mínimo error.

Por medio de PRS es posible encontrar buenas aproximaciones de óptimos locales, sin embargo, este método falla en la búsqueda de óptimos globales, por lo que es necesario

profundizar su estudio. En esa línea, se podría analizar la posibilidad de incorporar técnicas de aglomeración y criterios de separación para definir regiones de aproximación (Wikaisuksakul, 2014), en conjunto con metamodelos de aproximación dados por la combinación de varios modelos PRS (Lim et al., 2007). Así, en particular, en el uso de RBF se podría incluir un proceso para la optimización de los parámetros de dichas funciones y técnicas de aprendizaje como semiaprendizaje supervisado para el entrenamiento en la selección de los datos de entrada (Sun et al., 2014).

Bibliografía

Alexandrov, N. & Lewis, R. (April, 2000). *Analytical and Computational Aspects of Collaborative Optimization*. National Aeronautics and Space Administration, Langley Research Center, Virginia.

Chen, S. & Chiang, T. (July, 2014). *Evolutionary Many-objective Optimization by MO-NSGA-II with Enhanced Mating Selection*. IEEE Congress on Evolutionary Computation, Beijing, pp. 1397-1404.

Dastfan, A., Yassami, H. & Reza, M. (2014). Optimum Design of Passive Harmonic Filter by Using Game Theory. *Intelligence Systems in Electrical Journal*, (4), 13-22.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, England: John Wiley & Sons, Inc.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (April, 2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

Eskandari, H. & Geiger, C. (2008). A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems. *Journal Heuristics, Springer Science*, 14, 203-241.

Fliege, J. & Svaiter, B.F. (2000). Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operation Research*, *51*(3), 479-494.

Fliege, J., Graña, L.M. & Svaiter, B.F. (May, 2006). Newton's Method for Multiobjective Optimization. *SIAM Journal on Optimization*, *20*(2), 602-626.

Gen, M. & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. New York: John Wiley & Sons, Inc.

Gutiérrez-Méndez, F. (2011). *Optimización Multiobjetivo usando Algoritmos Genéticos Culturales*. Tesis de Maestro en Ciencias de la Computación. CINVESTAV, Unidad de Zacatenco, Departamento de Computación, México.

Haanpää, T. (2012). Approximation Method for Computationally Expensive Nonconvex Multiobjective Optimization Problems. *Jyväskylä Studies in Computing*,157. Jyväskylä.

Knowles, J. & Hughes, E. (2005). *Multiobjective Optimization on a budget of 250 evaluation*. Springer-Verlag Berlin Heidelberg, EMO 2005, LNCS 3410, 176-190.

Koch, P., Kramer, O., Rudolph, G. & Beume, N. (July, 2009). *On the Hybridization of SMS-EMOA and local search for Continuous Multiobjective Optimization*. GECCO'09. Montréal, Québec, Canada.

Lim, D., Ong, Y. & Jin, Y. (July, 2007). *A Study on Metamodeling Techniques, Ensembles and Multi-Surrogates in Surrogates-Assisted Memetic Algorithms.* Genetic and Evolutionary Conference, London, pp. 1288-1295.

Marler, R.T. & Arora, J.S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26, 369-395. DOI: 10.1007/s00158-003-0368-6.

Messac, A. & Mullur, A. (Spring, 2008). A computationally efficient metamodeling approach for expensive multiobjective optimization. *Optimization and Engineering*, 9(1).

Mosat, A. (2006). *Deterministic and stochastic batch design optimization techniques*. Zürich: ETH. DOI:10.3929/ethz-a-005344678.

Müller, J. & Shoemaker, C. (October, 2014). Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2), 123-144.

Ngatchou, P., Zarei, A., Fox, W. & El-Sharkawi, M. (2008). *Modern Heuristic Optimization Techniques. Chapter 10: Pareto Multiobjective Optimization*. Wiley-IEEE Press.

Ong, Y., Nair, P. & Keane, A. (2003). Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. *AIAA Journal*, 41(4), 687-696. DOI: 10.2514/2.1999

Ortega, J.M. & Rheinboldt, W.C. (2000). Iterative solutions of nonlinear equations in several variables. *SIAM Classics in Applied Mathematics*.

Parmee, I., Watson, A., Cvetkovic, D. & Bonham, C. (2000). Multiobjective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation, Massachusetts Institute of Technology, 8*(2), 197-222.

Pereira, R., Alvarado, P. & Krawtschneider, H. (May, 2007). *Desing of a MCML Gate Library Applying Multiobjective Optimization*. IEEE Computer Society Symposium on VLSI 2007, Brasil.

Reeves, C. & Rowe, J.E. (2003). *Genetic Algorithm: Principles and Perspectives. A guide to GA Theory*. Dordrecht: Kluwer Academic Publishers.

Rivieri, J., Le Riche, R. & Picard, G. (September, 2013). LOOM, an algorithm for finding local optima of expensive functions. En *New and Smart Information Communication Science and Technology to support Sustainable Development*. France: Clermont Ferrand.

Schweickardt, G. (diciembre, 2009). Metaheurística FPSO-*X* multiobjetivo. Una aplicación para la planificación de la expansión de mediano/largo plazo de un sistema de distribución eléctrica. *Energética*, (42), 73-88.

Sun, C., Jin, Y., Zeng, J. & Yu, Y. (April, 2014). A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Computing*. DOI: 10.1007/s00500-014-1283-z.

Vansundhara, R., Siva, S. & Selvaraj, M. (February, 2015). Evolutionary algorithms for de novo drug design. A survey. *Applied Soft Computing*, 27, 543-552.

Voutchkov, I. & Keane, A. (2010). Multiobjective Optimization Using Surrogates. En *Computational Intelligence in Optimization*. Springer-Verlag Berlin Heidelberg, ALO 7, 155-175.

Wikaisuksakul, S. (November, 2014). A multi-objective generic algorithm with fuzzy c-means for automatic data clusterig. *Applied Soft Computing*, 24, 679-691.

Zhang, Q. & Li, H. (December, 2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712-731.

Zhang, Q., Liu, W., Tsang, E. & Virginias, B. (February, 2009). *Expensive Mutiobjective Optimization by MOEA/D with Gaussian Process Model. Technical Report CES-489*. School of Computer Science & Electronic Engineering, University of Essex.

Zitzler, E., Laummans, M. & Thiele, L. (May, 2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. En *Computer Engineering and Networks Laboratory (TIK) Report*.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. & Grunert da Fonseca, V. (April, 2003). Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation, 7*(2).