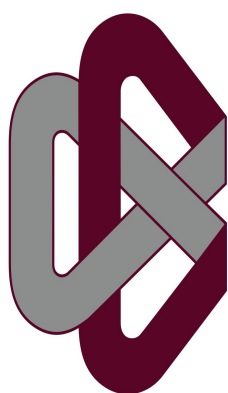


MAESTRÍA EN CIENCIAS CON
ESPECIALIDAD EN COMPUTACIÓN Y
MATEMÁTICAS INDUSTRIALES



CIMAT

INFORME
PROYECTO DE VERANO
JOEL CHACÓN CASTILLO

Introducción

En este documento se presenta el trabajo realizado en el proyecto de verano en el cual se investigan temas de optimización multi objetivo con algoritmos evolutivos y análisis de diversidad tanto en el espacio de las variables como en las funciones objetivo, los primeros temas está conformados de la teoría relacionada con conceptos de algoritmos multi objetivo, posteriormente se explica el esquema del código fuente de los algoritmos NSGAI y MOEA/D. Se exponen una recopilación de los problemas de prueba mas utilizados así como las propiedades del frente de pareto que conforma cada problema (tentativamente se agregarán otros problemas no continuos, como es el caso de 0-1 Knapshack Problem), por último se expone el módulo elaborado para realizar análisis las soluciones como resultado de algoritmo evolutivo implementado, en este punto es donde se proporcionan las herramientas para medir y analizar gráficamente los resultados (el lenguaje utilizado es R).

Temas investigados

Conceptos básicos de Algoritmos Evolutivos Multi Objetivo

- Definición de un problema de optimización Multi-Objetivo.
- Métodos tradicionales para resolver problemas Multi-Objetivo.
- Espacio de decisión de variables y espacio objetivo.
- Problemas de optimización Multi-Objetivo convexos y no convexos.
- Objetivos de algoritmos de optimización Multi-Objetivo.
- Optimalidad de pareto y dominancia.
 - Vector Objetivo Ideal.
 - Vector Objetivo Utópico.
 - Vector Objetivo Nadir.
 - Concepto de Dominación.
 - Propiedades de la relación de dominancia.
 - Optimalidad de pareto.
 - Conjunto global del pareto óptimo.
 - Dominancia fuerte y débil del pareto óptimo.

- Condiciones de optimalidad (No documentado).

Análisis de algoritmos multiobjetivos

- Representación ilustrativa de soluciones no dominadas.
 - Scatter-Plot Matrix Method.
 - Value Path Method.
 - Bar Chart Method.
 - Star Coordinate Method.
- Métricas para medir la proximidad al frente de pareto óptimo.
 - Error Ratio.
 - Set Coverage Metric.
 - Generational Distance.
- Métricas para evaluar la diversidad entre soluciones no dominadas.
 - Spacing.
 - Spread.
 - Maximum Spread.
 - Chi-Square-Like Deviation Measure.
- Métricas que evalúan cercanía al frente de pareto óptimo y diversidad de la solución no dominada.
 - Hypervolume.
 - Attainment Surface Base Statistical Metric.
 - Weighted Metric.
 - Non-Dominated Evaluation Metric.

Algoritmos para encontrar un conjunto no dominado

Se programaron algoritmos para encontrar y/o clasificar el frente de pareto no dominado:

- Algoritmos para encontrar el frante de pareto
 - Naive and Slow.
 - Continuously Updated.
- Algoritmos para clasificar el frente no dominado.
 - Non-Dominated Sorting of a Population.
 - Fast Non-Dominated Sort.

Algoritmo NSGAII

Se realizó el análisis de este algoritmo el cual es popular por su operador de selección y por el enfoque de “Fast Nondominated Sorting” los cuales ofrecen una complejidad menor que otros algoritmos como SPEA, la implementación con la que se cuenta sólo maneja la representación binaria “Binary-Encoded”.

Estructura del código fuente

La clases que se implementaron para describir al algoritmo NSGAII fueron:

- Individual.
 - Esta clase contiene la representación de cada individuo y su codificación ya sea binaria o real.
- Objectives.
 - Esta clase almacena los métodos relacionados con dominancia y se definen las variables en el espacio de decisión y la aptitud relacionada a cada función objetivo, cada función objetivo contiene su propio tipo de dualidad ya sea maximizar o minimizar, también esta clase representa la conexión entre la clase Benchmark donde se definen los problemas y se evalúa la variable de decisión.
- Measures.
 - En esta clase se implementan las métricas para analizar un determinado frente en relación al frente de pareto óptimo, para realizar la medición de la convergencia está implementado el

método de "Generational Distances", las métricas para la dispersión "Spacing" y "Spread". Adicionalmente se cuenta con la métrica "HyperVolume" y "HyperVolume Ratio".

- Benchmark.
 - En la implementación de esta clase se tienen contemplados una serie de problemas clasificados como ZDT, SCH donde cada test contiene información como los límites en cada variables de decisión en cada función objetivo.

Algoritmo MOEA/D

Se realizó el análisis de este algoritmo el cual consiste en descomponer un problema de optimización multi objetivo en un número de subproblemas sobre optimización escalar los cuales se optimizan simultaneamente. Teóricamente este algoritmo es de menor complejidad computacional que los algoritmos NSGAI y MOGLS. Existen varios enfoques para convertir el problema en un conjunto de subproblemas de optimización escalar, donde los tres (y una variación) enfoques más populares de descomposición son:

- Suma de pesos (Weighted Sum Approach), (Para frentes no convexos debe ser modificado).
- Tchebycheff.
- Intersección de límites (Boundary Intersection Approach).
- Intersección de límites basado en penalizaciones (Penalty-based Boundary Intersection).

Para obtener un mejor rendimiento de este algoritmo se implementan estrategias de descomposición complejas, esto permite obtener aproximaciones al frente de pareto óptimo en relación al problema establecido. Una ventaja es que se pueden implementar mecanismos de optimización multio objetivo clásica.

Estructura del código fuente

Para implementar este algoritmo se desarrolló un framework el cual sirve para realizar la programación de otros algoritmos evolutivos, este consiste en los ficheros:

- main
- EvolutiveAlgorithm
- Individual
- Objectives

- Benchmark
- Measures
- EAOperators

Para implementar una algoritmo con las propiedades de un EA, se genera una nueva clase la cual debe ser una clase hija de la clase `EvolutionaryAlgorithm`, posteriormente se pueden redefinir los métodos que se deseen de la clase base. En la clase nueva se maneja cada individuo como un miembro de la clase `Individual`, también se pueden definir los operadores que se deseen y agregar al fichero `EAOperators`.

Benchmark

En base a una recopilación de artículos y tesis se genera un módulo el cual contiene problemas los cuales se deben analizar, el principal artículo que se toma en cuenta es:

“A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit”, Simon Huband, Philip Hingston and Luigi Barone.

Donde se muestra amplio análisis de las funciones de prueba, el objetivo de este módulo es implementar las funciones del benchmark en el lenguaje de C++.

Módulo para analizar el frente de pareto

Este módulo ofrece herramienta para analizar el frente de pareto obtenido mediante algún algoritmo en contraste al frente de pareto ideal, además ofrece métricas para la convergencia y realizar la comparación de dominancia entre dos algoritmos distintos. La visualización del resultado obtenido se implementa por medio de R con un Script de análisis el cual procesa la información devuelta por un determinado algoritmo.

Referencias

Artículos:

- “(MOEA/D) A Multiobjective Evolutionary Algorithm Based on Decomposition”, Qingfu Zhang, and Hui Li.
- “A Fast and Elitist Multiobjective Genetic Algorithm”, NSGA-II Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan.
- “Running Performance Metrics for Evolutionary Multi-Objective Optimization”, Kalyanmoy Deb and Sachin Jain.
- “Performance metrics in multi-objective optimization”, Nery Riquelme, Christian Von Lüken and Benjamín Barán.
- “A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit”, Simon Huband, Philip Hingston and Luigi Barone.
-

Libros consultados:

- “Multi-Objective Optimization using Evolutionary Algorithms”, Kalyanmoy Deb.
- “Evolutionary Algorithms for Solving Multi-Objective Problems”, Carlos A. Coello Coello.
- Multiobjective Evolutionary Algorithms and Applications”, K. C. Tan, E. F. Khor, T. H. Lee.

Direcciones web:

- <http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/?page=testProblem.php#source>

Tesis:

- “Un Algoritmo Evolutivo Multiobjetivo basado en hipervolumen en GPUs”, Edgar Manóatls López, Carlos Coello Coello, CINVESTAV.
- “Optimización Multiobjetivo Usando algoritmos Genéticos Culturales”, Fernando Gutiérrez Méndez, Carlos Coello Coello, CINVESTAV.
- “Optimización Evolutiva Multiobjetivo basada en el Algoritmo de Kuhn-Munkres, José Antonio Molinet Berenguer, Carlos Coello Coello, CINVESTAV.
- “Optimización de problemas con más de tres objetivos mediante algoritmos evolutivos”, Mario Garza Fabre, Gregorio Toscano Pulido, CINVESTAV.