Algoritmos Genéticos Aplicados en la Optimización Multiobjetivo

Edwin Sneyder Gantiva Ramos[1]

Universidad Nacional de Colombia, Bogotá, Colombia,
I.esgantivar@unal.edu.co

Abstract. En el desarrollo de este documento se presenta una propuesta para darle solución al problema de la optimización multiobjetivo, tomando como base de trabajo la propuesta realizada en [2]. También se hace un breve acercamiento al marco teórico que envuelve dicho problema, puesto que algunos conceptos que antes se consideraban intuitivos, ahora hace falta hacer una revisión un poco mas detallada. La propuesta se trata de la implementación de un algoritmo genético convencional, con algunas modificaciones que se mencionaran con mas detalle en posteriores secciones del documento.

Keywords: Optimización, Algoritmos Geneticos, Optimización Multiobjetivo

1 Introducción

Los algoritmos genéticos han presentado buenos resultados para problemas de optimizacion en diferentes espacios de búsqueda como lo es el espacio de los números binarios y el espacio de los números reales. Estos se han potenciado acordes a diferentes trabajos que han desarrollado numerosos tipos de operadores genéticos y estrategias que han hecho que esta técnica de optimizacion tome mas relevancia en el ámbito de las ciencias de la computación.

Sin embargo estos algoritmos genéticos convencionales basan su estrategia de evolución en el mejoramiento continuo de los valores de la función de fitness o adaptación (Minimizar o Maximizar), y esta función es comúnmente una única función. El problema de la optimización multiobjetivo se puede resumir en que es una técnica que busca optimizar de manera simultanea varias funciones , por lo que el concepto del fitness se vuelve un tanto mas complicado que lo que se trabaja usualmente.

En este documento se busca dar una breve introducción a una de las estrategias que se han planteado para darle solución a este problema usando algoritmos genéticos. El documento esta divido de la siguiente manera: una sección esta dedicada a dar una pasada sobre el marco teórico del problema, la siguiente hace una presentación de la propuesta y finalizando se ofrecen los resultados obtenidos usando funciones que fueron presentadas en [1].

2 Marco Teórico

A menudo cuando nos enfrentamos a problemas de optimización mono-objetivo, es intuitiva la definición de cuando una solución es mejor o peor que otra solución, generalmente a este concepto lo asociamos al fitness ó a la función de adaptación. Sin embargo para abordar problemas que no solo tienen una función objetivo sino que pueden tener tantas como sean necesarias (Multiobjetivo), este concepto de cual solución es mejor que otra deja de ser intuitivo, para eso es necesario introducir el concepto de la dominancia, que se define a continuación.

2.1 Dominancia

La dominancia tiene la siguiente definición: X domina a Y si y solo si se cumple lo siguiente:

$$\forall_i (f_i(X) \leq f_i(Y)) \; i \in \{1, 2, 3, \ldots, n\}$$
$$\text{y}$$
$$\exists_i (f_i(X) < f_i(Y)) \; i \in \{1, 2, 3, \ldots, n\}$$

De lo anterior también podemos deducir que $X \approx Y$ si y solo si X no domina a Y y Y no domina a X.

2.2 Algoritmo Genético

Es importante recordar la definición básica de una algoritmo genético convencional, puesto que para el desarrollo de este trabajo este sera aplicado y adaptado para encontrar una aproximación al problema de la opimización mutiobjetivo.

Algorithm 2.1: Algoritmo Genetico($f, \mu, CondicionDeTerminacion$)

Inicializar la Población Inicial P_0;
Evaluar(P_0, f);
while $CondiciondeTerminacion$ **do**
  $descendientes = GENERACION(P_t, f, SELECCION)$;
  $P_t = FUNCION\_REEMPLAZO(descendientes, P_{t-1})$

2.3 The Strength Pareto Evolutionary Algorithm (SPEA)

Dada la naturaleza de la optimización multiobjetivo, hace falta aplicar algunas modificaciones al algoritmo genético convencional. Por su puesto se han estudiado diferentes alternativas, sin embargo para el desarrollo de este documento se ha tomado como referencia **SPEA** que fue presentado en [2].

De este aproximación a la solución es importante rescatar algunos conceptos que constituyen las principales modificaciones que se deben hacer frente a un algoritmo genético convencional.

- Almacena de manera adicional las soluciones de pareto encontradas hasta el momento.
- Usa el concepto de la dominancia para asignar un valor de fitness escalar a los individuos.
- Los individuos que se encuentran en el conjunto de pareto hacen parte del mecanismo de selección dentro del proceso de evolución de la población.

El proceso general se presenta en la figura 1. Para hacer una revisión con mas detalle de este algoritmo se recomienda dirigirse a [2].

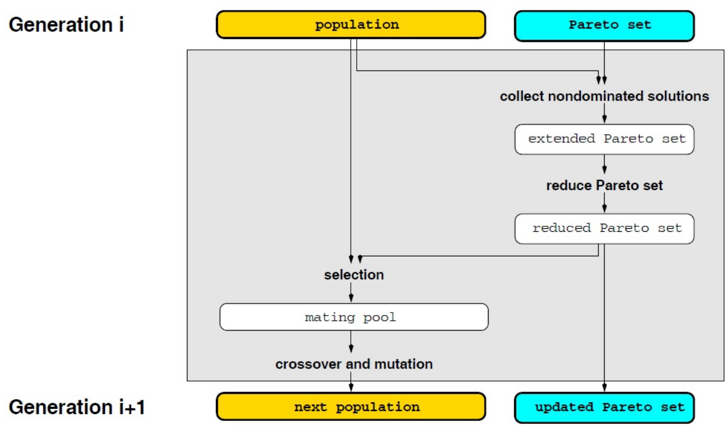

Fig. 1. Resumen del proceso implicado en la ejecución de SPEA, tomado de [2]

3 Propuesta

Basados en la teoría presentada en la anterior sección y en el trabajo presentado en [2], se ha diseñado una propuesta para la solución de este problema, primero se presenta en algoritmo planteado y posteriormente los procesos internos mas relevantes serán profundizados.

Funcionamiento General: Se inicializa la población, el conjunto de pareto es actualizado: todos los individuos no dominados en la población son copiados al conjunto de Pareto, enseguida posibles individuos dominados son removidos de este conjunto. Si el cardinal del conjunto de Pareto excede un máximo dado, se hace una reducción de este conjunto de forma aleatoria siguiendo una distribución de probabilidad uniforme. Se procede a calcular la fuerza de pareto para los individuos dentro del conjunto de pareto, basado en esta fuerza se calcula un fitness para los individuos de la población.

Una vez se hace lo anterior procediendo a un ciclo que se terminara de acuerdo a las condiciones de terminación. Dentro de este se aplicara un criterio de selección siguiendo la estrategia de ruleta basada en los fitness de los individuos. Se aplican siempre dos operadores genéticos: el primero un operador de ariedad 2 que realiza una combinación lineal de los genes dentro de los cromosomas así: $gd_i^1 = s * gp_i^1 + (1-s) * gp_i^2$ y $gd_i^2 = s * gp_i^2 + (1-s) * gp_i^1$, con s generado aleatoriamente con una distribución de probabilidad uniforme. Posteriormente se aplica un operador gaussiano de mutación, con parámetros dados por el usuario. El flujo general del algoritmo se presenta a continuación.

Algorithm 3.1: AG Multiobjetivo($f, \mu, \lambda, CondicionDeTerminacion$)

Inicializar la Población Inicial $P_0(\mu)$;
Calcular_Conjunto_Pareto $C_p(\lambda)$;
Remover_Posibles_Dominados_CPareto (C_p);
Calcular_Fuerza_Pareto(C_p);
Evaluar_Fitness(P_0);
while $CondiciondeTerminacion$ **do**
    $matingPool = P_t \cup C_{p_t}$;
    $P_{t+1} = GENERACION(matingPool, SELECCION)$;
    $C_{p_{t+1}} = C_p \cup Calcular\_Conjunto\_Pareto(P_{t+1})$;
    $Remover\_Posibles\_Dominados(C_{p_{t+1}}, \lambda)$;
    $Reducir\_Conjunto\_Pareto(C_{p_{t+1}})$;
    $Calcular\_Fuerza\_Pareto(C_{p_{t+1}})$;
    $Evaluar\_Fitness(P_{t+1})$

Algorithm 3.2: Calculo Fuerza Pareto(C_p, P_t)

foreach $paretoInd\ in\ C_p$ **do**
    $count := 0$;
    foreach $popInd\ in\ P_t$ **do**
        if $paretoInd\ isDominating\ to\ popInd$ **then**
            $count := count + 1$;
    $strength := count/(|P_t + 1|)\ setStrength(paretoInd, strength)$

Algorithm 3.3: Evaluar Fitness(P_t, C_p)

foreach $popInd\ in\ P_t$ **do**
    $sum := 0$;
    foreach $paretoInd\ in\ C_p$ **do**
        if $paretoInd\ is\ Dominating\ to\ popInd$ **then**
            $sum := sum + getFitness(paretoInd)$
    $setFitness(popInd, sum + 1)$

Nota : La propuesta anteriormente presentada se encuentra disponible para su libre uso y consulta en Repositorio en GitHub.

4 Resultados

Los resultados presentados en esta sección son el fruto de la experimentación sobre las funciones presentadas en [1], Que se definen de la siguiente manera:

$$f_m = (g(x), h(x))$$

donde

$$g(x) = x^2$$

y

$$h(x) = (x - 2)^2$$

.

El objetivo sera optimizar simultáneamente g y h, las funciones son graficadas de manera conjunta en la figura 2. Las gráficas de la distribución del conjunto de pareto se deben interpretar como una curva paramétrica de la siguiente manera, en el eje y se interpreta como la representación del valor de $h(x)$ y el eje x como la representación de $g(x)$.

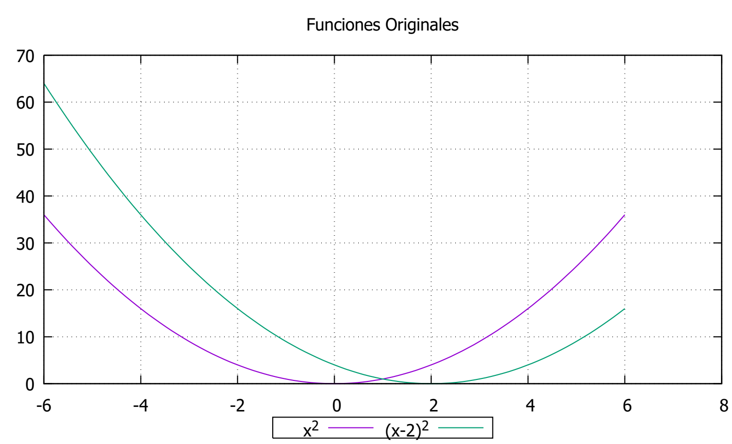

Fig. 2. Se presentan graficadas las funciones Schaffer's, entre el intervalo $[-6, 6]$

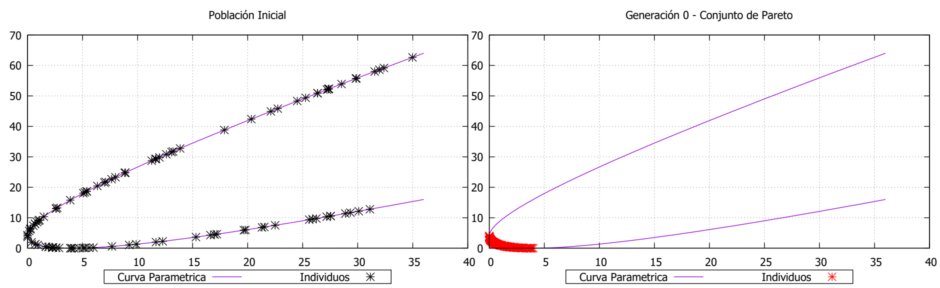

Fig. 3. Del lado izquierdo la distribución de la población inicial, Del lado derecho la distribución del conjunto de Pareto en la Generación 1

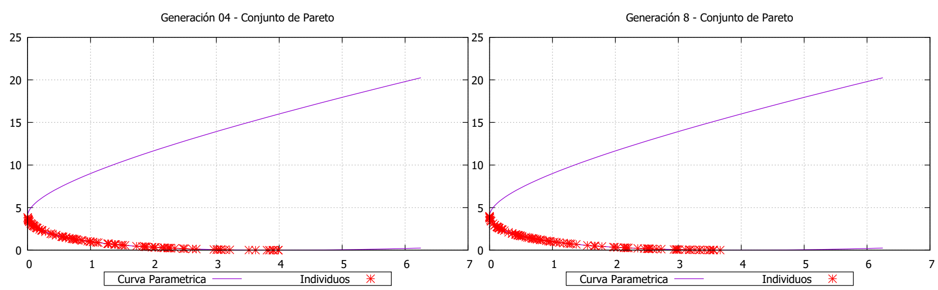

Fig. 4. Del lado izquierdo se presenta la distribución del conjunto de pareto sobre la curva paramétrica en la generación 4 y el lado derecho se presenta la distribución del conjunto de pareto en la generación 8

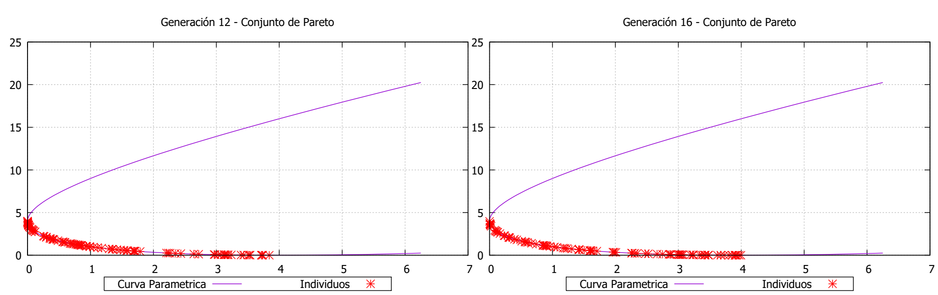

Fig. 5. Del lado izquierdo se presenta la distribución del conjunto de pareto sobre la curva paramétrica en la generación 12 y el lado derecho se presenta la distribución del conjunto de pareto en la generación 16

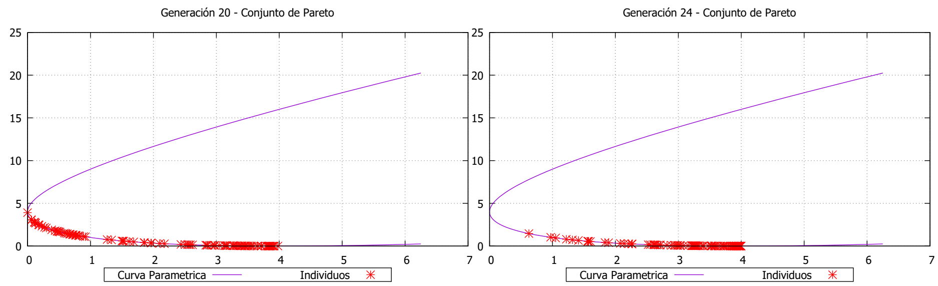

Fig. 6. Del lado izquierdo se presenta la distribución del conjunto de pareto sobre la curva paramétrica en la generación 20 y el lado derecho se presenta la distribución del conjunto de pareto en la generación 24

5 Conclusiones

- Se encuentra que el algoritmo propuesto converge rápidamente a la frontera de pareto, dado que la función multiobjetivo de prueba no posee gran complejidad.
- Se identifica que si se permiten grandes números de iteraciones, dado que no hay implementada una estrategia de control de diversidad la frontera de pareto tiende a converger a un único individuo.

Trabajo Futuro

De acuerdo a los procesos de experimentación realizados se encontró que para permitir grandes cantidades de iteraciones se hace necesario aplicar un método de control de diversidad, para evitar que los individuos de pareto converjan a uno único individuo. En [2] se plantea usando una técnica de clustering al momento de aplicar una reducción del conjunto de pareto. Sin embargo se sugiere aplicar un mecanismo de selección Deterministic Crowding con política de restricción a nivel de fenotipo, para lograr contrarrestar este problema de diversidad.

References

1. J David Schaffer, *Multiple objective optimization with vector evaluated genetic algorithms.*, Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, USA, July 1985, 1985, pp. 93–100.
2. Eckart Zitzler, Lothar Thiele, Eckart Zitzler, Eckart Zitzler, Lothar Thiele, and Lothar Thiele, *An evolutionary algorithm for multiobjective optimization: The strength pareto approach*, vol. 43, Citeseer, 1998.