

Proyecto Tecnológico

Vázquez Ballesteros Gabriel Tadeo

BENCHMARK 2006

En el 2006 se utilizaron 24 funciones, en las cuales había de tipo lineal, cuadráticas, cubicas y no lineales. Para cada función se tenía la información de las restricciones de igualdad y desigualdad lineales y no lineales.

Para realizar la comparación entre que algoritmo es mejor que otro se tomaba cada función se corría 25 veces con un máximo de 500 mil evaluaciones. El objetivo de cada función es Minimizar

$$f(x), x = (x_1, x_2, \dots, x_n)$$

Sujeto a

$$g_j(x) \leq 0, j = 1, \dots, q$$

$$h_j(x) = 0, j = q + 1, \dots, m$$

Se utilizó un $\epsilon = 0,0001$ como margen de error para que las restricciones de igualdad se transforman en desigualdades de la forma

$$|h_j(x)| - \epsilon \leq 0$$

Si el algoritmo cumplía con tales restricciones se consideraba como solución factible. La clasificación toma en cuenta que tal algoritmo cumpla con las restricciones y que no sobrepase el número de evaluaciones, la forma en que se realiza es a través del Success Performance(SP)

$$SP = \frac{\text{media}(\text{Num ECE}) * (\text{Num corridas})}{\text{Num corridas exitosas}}$$

ECE : evaluaciones para cada corrida exitosa.

BENCHMARK 2017-2018

En el 2017-2018 se utilizaron 28 funciones, en las cuales había de tipo separable, no separable y rotadas. Para cada función se tenía la información de las restricciones de igualdad y desigualdad lineales y no lineales.

Para este año no se mostró cual fue el máximo número de evaluaciones y el número total de corridas para cada función. Al igual que en el 2006 el objetivo de cada función es Minimizar

$$f(x), x = (x_1, x_2, \dots, x_n)$$

Sujeto a

$$g_j(x) \leq 0, j = 1, \dots, q$$

$$h_j(x) = 0, j = q + 1, \dots, m$$

Se utilizó un $\epsilon = 0,0001$ como margen de error para que las restricciones de igualdad se transforman en desigualdades de la forma

$$|h_j(x)| - \epsilon \leq 0$$

Para cada problema la clasificación del algoritmo es determinada en términos de la media de los valores y la mediana de

las soluciones con respecto al número máximo de evaluaciones permitidas. Para la media de los valores se utiliza la clasificación de los algoritmos basado en la tasa de factibilidad, la media la cantidad de violaciones y en términos de la media la función objetivo. Y para la mediana de las soluciones se utiliza la tasa de la factibilidad basada en su función objetivo, clasificación de la soluciones infactibles de acuerdo la suma de las restricciones violadas y que una solución factible es mucho mejor que una solución infactible.

Sumando estos términos, aquel que sea menor que otros, es mejor.

#1-2006 eDE

Este algoritmo utiliza como base un algoritmo evolutivo el cual es Evolución Diferencial. eDE es estable porque utiliza un mecanismo simple y estable de selección y reemplazo que excluye las operaciones estocásticas.

Este algoritmo introduce una mutación basada en gradiente que encuentra un punto factible desde un punto no factible utilizando el gradiente de restricciones en el punto no factible, así mismo introduce una restricción de violación $\phi(x)$ para indicar por cuánto un punto de búsqueda x viola las restricciones. Esta restricción de violación con respecto a un nivel ϵ ayuda para al principio relajar el sistema y aceptar soluciones no factibles, pero conforme pasa el tiempo esta restricción permite encontrar soluciones factibles.

La variante del método de cruce implementado es $DE/rand/1/exp$. La mutación basada en gradiente es una operación similar al método de mejoramiento basado en gradiente.

Si al realizar la operación de mutación el nuevo vector x está dentro de una región no factible con respecto $\phi(x)$ se utilizará el gradiente de restricciones hasta que dicho vector sea factible con respecto a nivel ϵ .

#2-2006 DMS-PSO

Este algoritmo utiliza como base un algoritmo evolutivo el cual es Particle Swarm Optimization. DMS-PSO originalmente está diseñado para manejar problemas de optimización sin restricciones, por lo tanto, para manejar problemas restringidos, se requiere un mecanismo de manejo de restricciones para guiar al enjambre en la búsqueda de una región viable. Basándose en su propiedad de varios enjambres, es natural dejar que los sub-enjambres se encarguen de diferentes tareas, es decir cada subpoblación intenta evolucionar a lo largo de una sola restricción o función objetivo y cada individuo en una subpoblación puede aparearse con cualquier otro en cualquier subpoblación. De esta manera, se espera que tenga una población de individuos factibles con altos valores de aptitud física.

En cada generación a cada individuo se le asigna una restricción al azar en orden de prioridad de manera que en cada sub-enjambre se tengan un tamaño de población similar, así al realizar las actualizaciones de cada partícula se realizará conforme a su grupo(aquel que tenga la asignación de la misma restricción). Por último cada cierto número de generaciones se seleccionan al azar n partículas a las cuales se les aplicará una búsqueda local utilizando Programación Secuencial Cuadrática(SQP) .

#3-2006 AG

Este algoritmo utiliza como base un Algoritmo Genético en el que se realizan una pequeñas modificaciones para adaptarse al problema de optimización con restricciones. La evaluación de los individuos cambiarán conforme a las generaciones, es decir, la función de evaluación con respecto a las restricciones tendrá una personalización baja y conforme al tiempo esta irá incrementando para así lograr tener soluciones factibles. El operador de cruza implementado es PCX(recombinación centrada en los padres).

En cada generación al crear λ descendientes, el método de reemplazamiento propuesto es seleccionar γ individuos de la población actual de forma aleatoria y ordenar conforme a la función de evaluación los $\lambda + \gamma$ individuos y los mejores $gamma$ individuos reemplazarán a los $gamma$ individuos de la población seleccionados.

#1-2018 UDE

Este algoritmo utiliza como base un algoritmo evolutivo el cual es Evolución Diferencial. UDE (Evolución Diferencial Unificado) está inspirado en las siguientes variantes de DE de: CoDE, JADE, SaDE, DE con operador de mutación basado en clasificación, SHADE y C2oDE. UDE utiliza tres estrategias de generación de vectores de prueba:

$DE/rand/1$, $DE/current-to-rand/1$, y $DE/current-to-pbest/1$. UDE se basa en un enfoque de doble población y en cada generación, UDE divide la población actual en dos subpoblaciones. En la subpoblación superior, UDE emplea las tres estrategias de generación de vectores de prueba en cada vector objetivo, al igual que en CoDE. Para la subpoblación inferior, UDE emplea la adaptación de la estrategia, en la cual las estrategias de generación de vectores de prueba se autoadaptan periódicamente al aprender de sus experiencias en la generación de soluciones prometedoras en la subpoblación superior. La operación de mutación en UDE se basa en la mutación basada en la clasificación. Además, UDE emplea el principio de adaptación de parámetros de SHADE. El principio de manejo de restricciones en UDE se basa en una combinación de la regla de viabilidad y la técnica de manejo de restricciones ϵ como se propone en C^2oDE es decir la misma que se propone para ϵDE .

#2-2018 ϵ MAG-ES

Este algoritmo utiliza como base el algoritmo MA-ES el cual para crear a los descendientes en cada generación selecciona a los mejores λ individuos con respecto a la función de evaluación la cual es adaptativa, al principio se relajan las

restricciones pero conforme pasa el tiempo éstas tienen mayor peso, a partir de estos λ individuos se crea un nuevo vector de desplazamiento de la población en el cual para crearlo es necesario utilizar la matriz de covarianza y descenso de gradiente de restricciones. Para mayor aclaración sobre como funciona creación de los descendientes es necesario contar con el paper : H.-G Beyer and B. Sendhoff, "Simplify-your covariance matrix adaptation evolution strategy"

#3-2018 LSHADE IEPSILO

Este método se basa en el algoritmo LSHADE (el cual se basa en DE) en el se proponen algunas variaciones, una de ellas es utilizar 4 operadores de mutación previo a mencionar cuales son $DE/current-to-pbest/1$ Bin, $DE/current-to-pbest/1$ Exp, $DE/rand/1*$ /1 Bin, $DE/rand/1*$ /1 Exp al principio ambas tienen la misma probabilidad de ser seleccionadas pero conforme vayan siendo durante cada generación se ira modificando la probabilidad de seleccionar alguna de tal manera que se procurara utilizar el mejor operador de mutación que se haya acoplado a tal función; para mejorar el rendimiento de DE se propone reducir el tamaño de la población dinamicamente mientras más evaluaciones de la función se realizan; la comparación de nivel ϵ es la misma que se propuso para ϵDE con una pequeña modificación por último los parámetros CR y F del individuo i se generara utilizando dos memorias S_F y S_{CR} las cuales guardan los parámetros exitosos de tal manera que F se creará a través una distribución Cauchy utilizando la memoria S_F y para CR se creará utilizando una distribución Normal utilizando la memoria S_{CR} .

DUDAS

Se encontraron las reglas y las funciones para el competición en el año 2010 y 2017 pero la información encontrada no mostraba ningún resultado.

En los Benchmark no se encontró sobre el tiempo máximo que podía ejecutarse un algoritmo, se plantea la hipótesis que los mejores algoritmos realizaban demasiados cálculos los cuales los benefician (como utilizar descenso de gradiente estocástico).

Es posible descargar artículos de la iEEE para poder complementar el algoritmo $\epsilon MAG - ES$.

Se encontró el siguiente enlace el cual contiene la información de las competiciones realizadas en cada año así como los códigos (para estos se encontraron algunos los cuales estaban implementados en Matlab) <http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared20Documents/Forms/AllItems.aspx>