

**BENEMÉRITA UNIVERSIDAD
AUTÓNOMA DE PUEBLA**

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**SELECCIÓN DE PARÁMETROS DE
ESTABILIZACIÓN OPTIMAL UTILIZANDO
ESTRATEGIAS EVOLUTIVAS**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

**LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

PRESENTA

MARIA LUISA MÓNICA LÓPEZ ARCHUNDIA

DIRIGIDO POR:

**DR. HÉCTOR SIMÓN VARGAS MARTÍNEZ
DRA. DARNES VILARIÑO AYALA**

DICIEMBRE DE 2008

INDICE

Capítulo 1 INTRODUCCIÓN	1
1.1 Metodología	8
1.2 Organización de los capítulos	9
CAPÍTULO 2 CONTROL ÓPTIMO	11
2.1 Declaración general de un problema de control óptimo	11
2.2 Resultados sobre controlabilidad	16
2.3 Propiedades de extremos y máximos de control óptimo y síntesis	18
2.4 Síntesis de controladores óptimos para procesos lineales de segundo orden	21
CAPÍTULO 3 ALGORITMOS EVOLUTIVOS	24
3.1 Un vistazo histórico a la computación evolutiva	24
3.1.1 El origen de las ideas	24
3.1.2 Russell y Darwin	25
3.1.3 La programación evolutiva	26

3.1.4	Las estrategias evolutivas	26
3.1.5	Los algoritmos genéticos	27
3.2	Principales paradigmas	29
3.2.1	Introducción	29
3.2.2	Programación evolutiva	30
3.2.2.1	Algoritmo	30
3.2.2.2	Algunas aplicaciones de la programación evolutiva	30
3.2.3	Estrategias evolutivas	31
3.2.3.1	Algoritmo	31
3.2.3.2	Convergencia	33
3.2.3.3	Auto-adaptación	34
3.2.3.4	Estrategias evolutivas vs programación evolutiva	34
3.2.3.5	Aplicaciones	35
3.2.4	Algoritmos genéticos	35
3.2.4.1	Algoritmo	35
3.2.4.2	Algoritmos genéticos vs otras técnicas evolutivas	36
3.2.4.3	Aplicaciones	38
3.2.5	Diferencias de las técnicas evolutivas con respecto a las tradicionales	38
3.2.6	Ventajas de las técnicas evolutivas	39
3.3	¿Cómo saber si es posible usar Estrategias Evolutivas?	40
Capítulo 4 ALGORITMO DE SELECCIÓN AUTOMÁTICA DE PARÁMETROS DE ESTABILIZACIÓN		42
4.1	Propiedades de matrices definidas positivas	42
4.2	Definición de operadores de estrategias evolutivas	46
4.3	Estrategias evolutivas representación y evaluación de aptitud	51
4.3.1	Mutación	55
4.3.2	Recombinación	58
4.3.3	Selección	63
4.3.4	Otros componentes	66
4.3.5	Algoritmo estándar conceptual	67
4.4	Algoritmo propuesto	70
Capítulo 5 PRUEBAS		76

	ÍNDICE
5.1 Ejemplos	76
Capítulo 6 CONCLUSIONES	80
BIBLIOGRAFIA	82

1

INTRODUCCIÓN

Los métodos estocásticos de búsqueda y de optimización, basados en los principios de la evolución biológica natural, han recibido un considerable aumento de interés en la última década.

Los algoritmos evolutivos (AEs) son una clase de técnicas de búsqueda que incluye los algoritmos genéticos (GAs) (Holland, 1975), programación evolutiva (EP) (Fogel et al., 1966), estrategias evolutivas (ES) (Rechenberg, 1973) y programación genética (GP) (Koza, 1991).

Los AEs operan sobre una población de soluciones potenciales que aplican el principio de la supervivencia del más apto para producir sucesivamente mejores aproximaciones a una solución. En cada generación del AE, un nuevo conjunto de aproximaciones es creado por el proceso de seleccionar a individuos según su nivel de aptitud en el dominio del problema y de reproducción según los operadores prestados de genética natural. Este proceso conduce a la evolución de poblaciones de los individuos que satisfacen mejor a su ambiente que los individuos de los cuales fueron creados.

Comparado a los métodos tradicionales de optimización, los AE son robustos, globales y se puede aplicar generalmente sin recurso al dominio específico de la heurística [1].

Las estrategias evolutivas (ES) originalmente emplearon mutación y selección a una población de un individuo. Schwefel (Schwefel, 1981) uso representación real e introdujo recombinación para poblaciones mayores que una individual. Se selecciona a los padres aleatoriamente y la recombinación y mutación se utilizan para producir más de N descendientes. La selección entonces se utiliza para seleccionar los N mejores descendientes o individuos de los mejores padres y descendientes para mejorar las generaciones siguientes.

Algunas consideraciones que se presentan comúnmente en problemas de ingeniería de control, y la manera en la que éstos se tratan con la aplicación de AE son:

Representación. Las variables continuas de decisión se pueden manejar directamente con representaciones valuadas reales y los apropiados operadores genéticos o usando esquemas de representación binaria y los operadores genéticos estándares. En el caso de representaciones binarias, los valores reales se pueden aproximar al grado necesario con un esquema binario de punto fijo. Sin embargo en la mayoría de los problemas de control, la precisión de los parámetros es más relativa que absoluta. En tales casos, el logaritmo del parámetro se puede codificar reduciendo el número de dígitos binarios y por lo

tanto el uso de la memoria. Alternativamente, una representación directa de punto flotante se podría utilizar.

Escala. El concepto de aptitud es central a todos los acercamientos del AE. Dado que muchos problemas de optimización son caracterizados por una función objetiva valuada real, estos valores se deben convertir en un valor de aptitud no negativo para ser manejados por el AE.

Adaptación. La extensa mayoría de aplicaciones de AE se ha concentrado en su uso como una función para optimizar. Sin embargo, el AE se ha mostrado para satisfacer los sistemas que varían en tiempo (Dasgupta y McGregor, 1992), es decir uno en el cual la aptitud o aptitudes de criterio cambian con el tiempo. Tales cambios ocurren típicamente como resultado de un cambio en el ambiente externo, como un cambio en condiciones de funcionamiento, o debido a los cambios de sistema, como desgaste de componentes mecánicos.

El AE tiene la ventaja sobre muchos métodos convencionales de poder responder a tales cambios explotando la diversidad de los individuos en la población actual. Si hay insuficiente diversidad en la población, entonces nuevo material puede ser introducido fácilmente substituyendo a algunos individuos por los individuos aleatoriamente inicializados.

Software. Mientras que existen muchos paquetes del dominio del algoritmo genético, como por ejemplo GENESYS (Grefenstette, 1990) y GENITOR (Whitley, 1989), ninguno de éstos proporcionan un ambiente que sea

inmediatamente compatible con las herramientas existentes en el dominio del control.

La caja de herramientas de algoritmo genético de MATLAB (Chipperfield et al., 1994) apunta a hacer el AG accesible al ingeniero de control.

Paralelismo. Aparte de las ventajas obvias de acelerar el tiempo de ejecución, el AG paralelo tienen un grado más alto de robustez, y requiere típicamente pocas evaluaciones de la función de aptitud para alcanzar soluciones óptimas. Empleando una estructura distribuida de la población con la selección local y reproducción y una cierta forma de movilidad genética, el funcionamiento del AG se puede realzar sobre uno donde tratan a la población global. Estas ventajas se pueden observar cuando el AG se pone en ejecución en una máquina secuencial.

Una de las aplicaciones más comunes del AE es la optimización paramétrica. En diseños de sistemas de control, muchas tareas se pueden unir dentro de un marco de optimización, en particular, en diseños de sistemas de control a través del parámetro de optimización, la solución final puede ser sensible a la estimación de la solución inicial o puede haber un número de combinaciones de las configuraciones del parámetro que producen la acción deseada de control.

Los optimizadores convencionales pueden por lo tanto encontrar solamente soluciones subóptimas mientras que la búsqueda es forzada a considerar solo pequeñas regiones del espacio de búsqueda influenciado por la

estimación inicial. El AE es capaz de muestrear el espacio entero de soluciones y se puede esperar producir soluciones que son más globales en naturaleza.

Además, los acercamientos evolutivos tienen el potencial de encontrar soluciones en muchas áreas diferentes del espacio de búsqueda. Así más información con respecto a la naturaleza del diseño del problema se puede obtener durante la búsqueda, rindiendo un proceso potencialmente más informado de diseño.

El AE ha sido mostrado para ser una estrategia eficaz en el diseño de los sistemas de control a través de la optimización paramétrica. Varsek et al (1993) han mostrado cómo el AE se puede utilizar en la selección de estructuras controlables, mientras que en robótica, Gleghorn et al (1989) han demostrado cómo el AE se puede utilizar para problemas de planeación de trayectorias en ambientes inmóviles y móviles.

La optimización con objetivos múltiples extiende la teoría de la optimización permitiendo que múltiples objetivos sean optimados simultáneamente. Ha sido usada durante años en economía (Takayama, 1974) y en ciencias gerenciales (Evans, 1984) y gradualmente se ha introducido en la ingeniería (Wu, 1995). Es conocida por diversos nombres tales como *Optimización Pareto*, *Optimización Vectorial*, *Optimización Eficiente*, *Optimización Multicriterio* y otros más. Las soluciones suministradas son denominadas *Óptima Pareto*, *Vector Máximo*, *Puntos Eficientes* y *Soluciones No Dominadas*.

Casi todos los problemas reales involucran la optimización de varios objetivos que a menudo son conflictivos y competitivos entre sí. Mientras que en la optimización con un simple objetivo usualmente la solución óptima está claramente bien definida, no sucede lo mismo en problemas con objetivos múltiples. En lugar de un óptimo simple, existen un conjunto de alternativas conocidas como soluciones *óptimo Pareto*. Éstas soluciones son óptimas en al amplio sentido de que ningunas otras soluciones son superiores a éstas cuando todos los objetivos son considerados. Los problemas de optimización multiobjetiva son muy comunes en el área de la ingeniería y las técnicas para resolverlos muy diferentes a las de la optimización simple.

Los problemas de control requieren muy raramente la optimización de una sola función objeto. En cambio generalmente hay un gran número de diseño de objetivos competentes que se requieren satisfacer simultáneamente. Convencionalmente, se busca a miembros del conjunto de soluciones óptimo Pareto a través de la solución de un problema de programación no lineal apropiadamente formulada.

Además, los métodos de programación no lineales no pueden manejar en un buen espacio funciones discontinuas y multimodales así se puede esperar producir soluciones locales.

En un problema de *OM* (OPTIMIZACIÓN MULTI OBJETIVO) el objetivo es hallar el conjunto *Pareto Óptimo* o *Frontera Eficiente*. Para alcanzarlo, los métodos tradicionales agregan los objetivos en una simple y parametrizada función objetivo. Tales técnicas, por lo general, llevan a cabo

varias corridas con diferentes parámetros de tal manera de obtener un conjunto de soluciones que se aproximen al conjunto *Pareto Óptimo*. Éstas técnicas son básicamente independientes del algoritmo de optimización subyacente. Ejemplos de ésta clase de técnicas son: *método de suma ponderada de objetivos, método de prioridades o restricciones, técnica minmax y programación por metas*.

El AG Multiobjetivo (Fonseca y Fleming, 1994) desarrolla una población de las soluciones estimadas para otorgar un beneficio inmediato sobre los métodos de OM.

Muchos problemas en la ingeniería de control, el procesado de señales y maquinas de enseñanza, se pueden unir como un sistema de identificación de problemas donde la tarea es determinar un modelo apropiado de un conjunto dado de datos de entrada- salida. El modelo que resulta puede entonces ser utilizado para la predicción y control de un sistema de "caja negra".

La habilidad de los AE para resolver problemas de optimización combinatorios complejos eficientemente es una consideración importante.

Hasta ahora se ha presentado una descripción de AEs en la ingeniería de sistemas de control, describiendo muchos rasgos y las características de AEs que son importantes y relevantes a la hora de poner estos algoritmos en práctica para las aplicaciones de la ingeniería de control. Claramente, el AE se ha convertido en un procedimiento establecido de optimización en la ingeniería de control y en otras partes.

Por estas razones se considera la utilización de Algoritmos Evolutivos para la realización de esta tesis, donde el objetivo es la construcción de un AE para la selección de los mejores parámetros de control para un robot móvil autónomo. El sistema dinámico tiene la siguiente ecuación de control:

$$\dot{x} = Ax + Bu$$

y

$$J = \int (x^T Qx + u^T Ru) dt$$

J depende de Q y R (Q y R son matrices definidas positivas), entonces la mejor elección de Q y R nos dará el mejor control óptimo.

Pero no existe un método formal para la selección de Q y R , es decir la selección de Q y R se da por lo general por un especialista ingeniero en base al entorno de las necesidades que requiere para la selección de estas matrices.

Así se propone que la selección de estas matrices sea mediante la utilización de algoritmos evolutivos, particularmente el uso de Estrategias Evolutivas

1.1 Metodología

1. Estudio de control moderno hasta llegar al control óptimo.
2. Estudio de los Algoritmos Evolutivos que se han realizado hasta la fecha y, en particular los operadores que se utilizan para conseguir la convergencia y la diversidad de la población.

3. Diseño y desarrollo de un algoritmo evolutivo con dos nuevos operadores para elegir el mejor control óptimo.
4. Aplicación del en un problema real de control específicamente un robot móvil autónomo.
5. Obtención de resultados.
6. Conclusiones.

1.2 Organización de los Capítulos.

La tesis aquí esta dividida en 6 capítulos. En el capítulo 2 se hace una revisión del control moderno, una definición de que es un sistema dinámico y una descripción de los diferentes algoritmos de estabilización para así proponer una estrategia de control y llegar a la elección del mejor control óptimo, por lo cual se discutirá por qué utilizar el algoritmo evolutivo como una mejor solución para la selección de los mejores parámetros de control óptimo.

En el capítulo 3 se da un breve introducción de que es el algoritmo evolutivo, se estudian los problemas que debe resolver y se analizan los operadores para conseguirlo, así como diversas teorías acerca de la convergencia del algoritmo evolutivo.

En el capítulo 4 se plantea el algoritmo evolutivo para la selección de los parámetros de control Q y R , se definen las operaciones cruzar y mutar y se demuestra que estas dos operaciones (“cruzar” y “mutar”) conserven las propiedades de las matrices de control Q y R .

En el capítulo 5 se presenta una descripción de un robot móvil autónomo, se da a conocer el algoritmo de control para el robot móvil y se aplica el algoritmo evolutivo en dicho algoritmo de control. Se comparan los resultados.

Y por último en el capítulo 6 se presentan las conclusiones de la tesis.

2

CONTROL ÓPTIMO

2.1. Declaración general de un Problema de control óptimo

El problema mas general de control optimo se describe por cuatro tipos de datos: (1) la planta o proceso, (2) el estado inicial y el estado deseado del sistema dinámico, (3) la clase de controles admisibles y (4) el costo funcional o el índice de ejecución.

1. **La planta o proceso** relaciona el estado o respuesta $x(t)$ a la entrada o control $u(t)$ por un sistemas de vector diferencial ordinario.

(S)

$$\dot{x}^i = f^i(t, x^1, x^2, \dots, x^n, u^1, \dots, u^m) \quad i = 1, 2, \dots, n$$

A veces $x(t)$ se le llama salida, pero definiremos después la salida como alguna función de x determinada por una condición adicional de observabilidad o medida. El proceso es autónomo, o lineal, o de n-ésimo orden, según la naturaleza del sistema diferencial S.

No linearizaciones pueden ocurrir, incluso, en las teorías físicas elementales debido a los efectos de fricción no lineal, amplificación, o saturación.

Sin embargo, para los procesos lineales introduciremos retroalimentaciones no lineales, como un control de relevo, para sintetizar el control óptimo. Es más, muchos sistemas físicos tienen aspectos no lineales mayores que no pueden ignorarse o tratarse con éxito por los métodos de perturbaciones o aproximaciones lineales. En cada problema de control óptimo nuestra última meta es sintetizar el control óptimo por un diseño apropiado o lazo cerrado de retroalimentación. La ventaja de tal lazo cerrado de control, es que el proceso llega a ser auto ajustable y auto corrector. Un control de retroalimentación puede corregir a menudo las variaciones imprevisibles en el ambiente de la planta o perturbaciones repetidas, o irregularidades en el proceso.

2. **El punto inicial o estado** x_0 es un vector conocido en el espacio fase, en un proceso físico real x_0 y la respuesta $x(t)$ podría tener componentes que describen la posición, velocidades, velocidades angulares, temperaturas, u otros datos que son moderados y grabados por instrumentos apropiados de diseño. También un conjunto G se prescribe para el problema del control, como el conjunto deseado.

A veces el conjunto G será continuamente cambiado con $G(t)$ en $\tau \leq t \leq \tau_1$. Es decir, para cada instante t designamos un conjunto compacto no vacío $G(t)$ en R^n . La continuidad de $G(t)$ como una función de la variable real t se define en términos del siguiente concepto de la distancia de $G(t)$ a $G(t')$:

$$dist(G(t), G(t')) = \max \left[\max_{P \in G(t)} dist(P, G(t')), \max_{P' \in G(t')} dist(P', G(t)) \right].$$

Así para el instante t y dado $\varepsilon > 0$ hay un $\delta > 0$ tal que

$$dist(G(t), G(t')) < \varepsilon \quad \text{Siempre que } |t - t'| < \delta$$

si $G(t)$ es un punto que traza una curva suave $\xi(t)$ en R^n , estudiamos a menudo el error de desviación de $x(t)$ del objetivo

$$e(t) = x(t) - \xi(t).$$

Aquí $x(t)$ es la respuesta del proceso de control

$$S \quad \dot{x} = f(t, x, u)$$

y el error $e(t)$ satisface el proceso de control

$$\dot{e} = f(t, e + \xi(t), u) - \dot{\xi}(t) = \tilde{f}(t, e, u).$$

En esta interpretación deseamos controlar $e(t)$ a cero, esto es, regular el error a cero.

3. Las **clases Δ de controladores admisibles** consiste de funciones medibles $u(t)$ sobre varios intervalos de tiempo $t_0 \leq t \leq t_1$ cada uno de los cuales dirigen x_0 $G(t)$. Esto es, la solución $x(t)$ de

$$(S) \quad \dot{x} = f(t, x, u) \quad \text{Con} \quad x(t_0) = x_0$$

Satisface $x(t_1) \in G(t_1)$.

Suponga que S es autónomo y x_0 es dirigido a x_1 por $u(t_1)$ sobre $t_0 \leq t \leq t_1$. Si $u_2(t)$ sobre $t_0 \leq t \leq t_1$ dirige x_1 a x_2 , entonces el control total

$$u(t) = \begin{cases} u_1(t) & \text{sobre } t_0 \leq t \leq t_1 \\ u_2(t - t_1 + t_0) & \text{sobre } t_1 \leq t \leq t_1 + t_2 - t_0 \end{cases}$$

Dirige x_0 a x_2 . Una observación similar muestra que para sistemas autónomos podemos siempre empezar el control en $t = 0$ sin pérdida de generalidad.

Restricciones adicionales son a menudo impuestas sobre funciones que comprenden Δ . Por ejemplo la condición $u(t) \in \Omega$ donde Ω es una restricción

fija, convexa compacta del conjunto en R^m . También el tiempo inicial o final para la duración de los controladores es en ocasiones preescrita.

4. **Costo funcional** (o índice de ejecución u objetivo funcional o esfuerzo) es un criterio aceptado cuantitativamente para la eficiencia de cada controlador $u(t)$ sobre $t_0 \leq t \leq t_1$ en la clase Δ . Si Δ consta de controladores sobre varios intervalos de tiempo que aproximen x_0 $G(t)$, a menudo se define el costo de $u(t)$ como

$$C(u) = \int_{t_0}^{t_1} f^0(t, x(t), u(t)) dt.$$

Donde $f^0(t, x, u)$ es una función continua dada. Si $f^0(t, x, u) \equiv 1$, entonces $C(u) = t_1 - t_0$ y tenemos el problema de tiempo mínimo.

Algunas veces Δ consta de controladores con una duración de tiempo fija específica, es decir $t_0 \leq t \leq T$, sin el requerimiento de alcanzar a $G(t)$, es decir la curva $\xi(t)$. A menudo se tiene el costo funcional

$$C(u) = |x(T) - \xi(T)| + \int_{t_0}^T f^0(t, x(t), u(t)) dt.$$

En particular hay cierta importancia en el costo funcional cuadrático, refiriéndose al significado de error de la respuesta $x(t)$ y la energía expandida por el controlador $u(t)$,

$$C(u) = g(x(T)) + \int_{t_0}^T [x'(t)W(t)x(t) + u'(t)U(t)u(t)] dt.$$

Aquí, $g(x)$ se especifica como son, la matrices simétricas $W(t)$ y $U(t)$; estas son positivas (semi) definidas así que $x'Wx \geq 0$ y $u'Uu > 0$ para vectores no cero x y u .

Considere un problema de control con (1) proceso S, (2) punto inicial $x(0)$ y objetivo $G(T)$, (3) controladores admisibles Δ y (4) costo funcional $C(u)$ definidas sobre todos los controladores u en el conjunto no vacío Δ .

Definición: Un controlador $u^*(t)$ en la clase admisible Δ es llamado (mínimo) óptimo, con respecto al costo funcional real $C(u)$, en caso

$$C(u^*) \leq C(u)$$

Para todo $u(t)$ en Δ .

Teorema 1: El problema de control consiste de

1. Un sistema diferencial

$$(S) \quad \dot{x}^i = g^i(t, x) + h_j^i(t, x)u^j \quad \text{Para } i=1, \dots, n \quad \text{y } j=1, \dots, m$$

Con

$$g^i(t, x), h_j^i(t, x) \quad \text{y} \quad \frac{\partial g^i}{\partial x^k}(t, x) \quad \frac{\partial h_j^i}{\partial x^k}(t, x)$$

Para $k=1, \dots, n$, todos continuos en $R^1 \times R^n$

2. Un conjunto no vacío, convexo, de restricción compacta $\Omega \subset R^n$.
3. El punto inicial $x_0 \in R^n$ y el conjunto continuo $G(t) \subset R^n$ sobre el intervalo finito $\tau_0 \leq t \leq \tau_1$.
4. El funcional de costo

$$C(u) = \int_{t_0}^{t_1} [g^0(t, x(t)) + h_j^0(t, x(t))u^j(t)] dt.$$

Donde $g^0(t, x)$ y $h_j^0(t, x)$ son continuos sobre $R^1 \times R^n$.

Sea $\Delta = \Delta(S, \Omega, x_0, G)$ la clase de controladores medibles $u(t) \subset \Omega$

Sobre los subintervalos $t_0 \leq t \leq t_1$ de $\tau_0 \leq t \leq \tau_1$ con respuestas $x(t)$ dirigiendo $x(t_0) = x_0$ a $x(t_1) \in G(t_1)$.

Asuma

- (a) Δ es no vacío.
- (b) existe un límite uniforme $B < \infty$ para todas las respuestas $x(t)$ correspondiente a los controles en Δ , esto es, $|x(t)| \leq B$.

Entonces existe un control óptimo $u^*(t)$ en Δ .

También puede demostrarse que si $\Delta(\alpha) \subset \Delta$ (controladores con $t_0 = \alpha$ fijo) o si $\Delta(\alpha, \beta) \subset \Delta(\alpha)$ (controladores con $t_0 = \alpha$ y $t_1 = \beta$ fijos) es no vacío, entonces un controlador óptimo existirá para la correspondiente clase admisible de controladores.

2.2. Resultados sobre Controlabilidad

Definición: Un proceso de control autónomo

$$\dot{x}^i = f^i(x^1, \dots, x^n, u^1, \dots, u^m) \quad i = 1, \dots, n$$

Con $f(x, u) \in C^1$ en $R^n \times R^m$, es llamado (completamente) controlable en caso de que:

Para cada par de puntos x_0 y x_1 en R^n exista un controlador medible limitado $u(t)$ sobre algún intervalo finito $0 \leq t \leq t_1$ tal que la respuesta correspondiente $x(t)$ dirijan $x(0) = x_0$ a $x(t_1) = x_1$.

Asuma

(a) $f(0,0) = 0$;

(b) la clase Δ de controladores admisibles incluyen todos los controladores medibles $u(t)$ sobre intervalos finitos de tiempo que satisfagan $|u(t)| \leq \varepsilon$ para algún $\varepsilon > 0$;

(c) el sistema diferencial lineal

$$\dot{x} = Ax + Bu.$$

Con coeficientes matrices constantes

$$A = \left(\frac{\partial f^i}{\partial x^i}(0,0) \right) \quad \text{y} \quad B = \left(\frac{\partial f^i}{\partial u^i}(0,0) \right)$$

es controlable, esto es

$$\text{rango}[B, AB, A^2B, \dots, A^{n-1}B] = n.$$

Entonces el dominio C de controlabilidad contiene una vecindad abierta del origen en R^n .

Corolario. Considere el proceso de control lineal

$$\dot{x} = Ax + Bu,$$

donde A es una matriz constante real de $n \times n$ y B es una matriz constante real de $n \times m$.

Asuma

(a) A es estable, esto es, cada eigenvalor λ de A tiene $\text{Re } \lambda < 0$;

(b) Controlabilidad, esto es, $\text{rango}[B, AB, A^2B, \dots, A^{n-1}B] = n$.

Entonces cada estado inicial x_0 puede ser dirigido a $x_1 = 0$ por algún controlador medible $u(t)$ sobre in intervalo finito $0 \leq t \leq t_1$. Además podemos requerir que $|u(t)| \leq \varepsilon$ para un $\varepsilon > 0$ arbitrariamente prescrito.

2.3. Propiedades de Extremos y Máximos de Control Óptimo y Síntesis

En cálculo diferencial, el mínimo de una función de una variable real se localiza examinando los puntos críticos, esos puntos en que la derivado es cero. Un procedimiento análogo se sigue en la teoría de control.

En el principio del máximo cada controlador óptimo $u(t)$ es un controlador máximo, esto es, $u(t)$ es crítico para el problema de control.

Consideremos el problema de control autónomo:

$$1. (S) \quad \dot{x}^i = f^i(t, x^1, x^2, \dots, x^n, u^1, \dots, u^m) \quad i=1, 2, \dots, n$$

con $f(x, u) \in C^1$ en $R^n \times \Omega$.

2. El punto inicial x_0 y el conjunto objetivo G , un conjunto compacto no vacío en R^n .

3. La clase Δ de controladores medibles $u(t)$ sobre varios intervalos finitos $0 \leq t \leq t_1$, los cuales dirijan x_0 a G , con $u(t) \subset \Omega$, el conjunto restringido compacto en R^m .

4. El funcional de costo $C(u) = \int_0^{t_1} f^0(x(t), u(t)) dt$ donde

$f^0(x, u) \in C^1$ en $R^n \times \Omega$.

Definición: Considere el proceso de control autónomo $\{S, x_0, G, \Omega, \Delta, C\}$ como el descrito anteriormente. Sea $u(t)$ sobre $0 \leq t \leq t_1$ un controlador en Δ con respuesta $x(t) = (x^i(t)), i = 1, \dots, n$. Extendemos la respuesta a un vector $(n+1)$

$$\hat{x}(t) = (x^\alpha(t)) \quad \alpha = 0, 1, \dots, n,$$

por definición

$$x^0(t) = \int_0^t f^0(x(t), u(t)) dt.$$

Un vector $(n+1)$ $\hat{\eta}(t) = (\eta_\alpha(t))$ sobre $0 \leq t \leq t_1$ se llama respuesta adjunta en caso de que $\hat{\eta}(t)$ sea una solución no dispersa del sistema diferencial Hamiltoniano

$$\dot{x}^\alpha = \frac{\partial \hat{H}}{\partial \eta_\alpha} = f^\alpha(x, u(t)) \quad \alpha = 0, 1, \dots, n$$

$$\dot{\eta}_\alpha - \frac{\partial \hat{H}}{\partial x^\alpha} = -\eta_0 \frac{\partial f^0}{\partial x^\alpha}(x, u(t)) - \dots - \eta_n \frac{\partial f^n}{\partial x^\alpha}(x, u(t))$$

de la función Hamiltoniana definida por

$$\hat{H}(\hat{\eta}, \hat{x}, u) = \eta_0 f^0(x, u) + \eta_1 f^1(x, u) + \dots + \eta_n f^n(x, u)$$

y definimos

$$\hat{M}(\hat{\eta}, \hat{x}) = \max_{u \in \Omega} \hat{H}(\hat{\eta}, \hat{x}, u)$$

Entonces, definimos el controlador $u(t)$ sobre $0 \leq t \leq t_1$ sea máximo en caso de que exista una respuesta adjunta $\hat{\eta}(t)$ así que

1. $\hat{H}(\hat{\eta}(t), \hat{x}(t), u(t)) = \hat{M}(\hat{\eta}(t), \hat{x}(t))$ casi por todas partes sobre $0 \leq t \leq t_1$,
- y
2. $\hat{M}(\hat{\eta}(t), \hat{x}(t)) = 0$ por todas partes, y $\eta_0 \leq 0$.

Teorema 4: Considere el problema de control autónomo $\{S, x_0, G, \Omega, \Delta, C\}$ como el descrito anteriormente. Sea $u(t)$ sobre $0 \leq t \leq t_1$ un controlador óptimo en Δ . Entonces $u(t)$ es un controlador máximo.

Para clarificar la naturaleza del principio del máximo consideraremos la declaración del principio para un problema de control lineal autónomo:

1. $S: \dot{x} = Ax + Bu$ para la matriz constante real de $n \times n$ A y la matriz de $n \times m$ B ;

2. Punto inicial x_0 en el dominio de controlabilidad nula y objetivo G como el origen;
3. El conjunto restringido compacto convexo $\Omega \subset R^m$
4. $C(u) = \int_0^{t_1} dt = t_1$ el tiempo de duración de control.

En este caso la función Hamiltoniana es

$$\hat{H}(\hat{\eta}, \hat{x}, u) = \eta_0 + \eta[Ax + Bu] = \eta_0 + H(\eta, x, u)$$

donde $\eta = (\eta_i)$, $i = 1, \dots, n$ es un vector n-fila, y $\eta(Ax + Bu) = H$. Entonces

$$\hat{M}(\hat{\eta}, \hat{x}) = \eta_0 + \eta Ax + \max_{u \in \Omega} \eta Bu = \eta_0 + M(\eta, x),$$

donde $M = \max_{u \in \Omega} H$. Si $u(t)$ sobre $0 \leq t \leq t_1$ es máximo, entonces la respuesta

$x(t) = (x^i(t))$, $i = 1, \dots, n$ satisfacen

$$\dot{x} = Ax + Bu(t)$$

$$\dot{\eta} = -\eta A$$

y $x^0 = t$, $\eta_0 = \text{constante}$

El principio del máximo requiere que

$$\eta_0 + \eta(t)Ax(t) + \eta(t)Bu(t) = \eta_0 + \eta(t)Ax(t) + \max_{u \in \Omega} \eta(t)Bu$$

o

$$\eta(t)Bu(t) = \max_{u \in \Omega} \eta(t)Bu.$$

Casi por todas partes sobre $0 \leq t \leq t_1$.

La segunda condición del principio del máximo afirma que

$$\eta_0 + \eta(t)Ax(t) + \max_{u \in \Omega} \eta(t)Bu = 0.$$

Por todas partes. Si $\eta(t)$ es dispersado a un punto sobre $0 \leq t \leq t_1$, entonces dispersa idénticamente desde que es una solución del sistema lineal homogéneo

$$\dot{\eta} = -\eta A.$$

Pero si $\eta(t) \equiv 0$, entonces $\eta_0 = 0$, y esto contradice la no dispersión del vector $(n+1) \hat{\eta}(t)$. Por consiguiente $\eta(t)$ en ninguna parte es cero.

2.4 Síntesis de Controladores Óptimos para procesos lineales de segundo orden.

Completaremos la síntesis para el tiempo mínimo del controlador óptimo para la regulación de (x, \dot{x}) a $(0,0)$ para el proceso lineal de segundo orden mas general

$$\ddot{x} \pm 2b\dot{x} \pm k^2 x = u,$$

donde $b \geq 0$ y $k^2 > 0$ son constantes y la restricción de control es $|u(t)| \leq 1$.

Considere la síntesis del controlador de tiempo óptimo mínimo para la regulación a cero del proceso lineal.

$$(L) \quad \ddot{x} \pm 2b\dot{x} \pm k^2 x = u,$$

con constantes $b \geq 0$ y $k^2 > 0$ y restricción de control $|u(t)| \leq 1$. El sistema lineal correspondiente

$$(S) \quad \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \mp k^2 & \mp 2b \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

Es controlable y normal. De los teoremas 1, 2, y 3 existe un controlador óptimo $u^*(t)$ sobre $0 \leq t \leq t^*$ dirigiéndose al estado inicial dado (x_0, y_0) , el cual

miente en el dominio C de controlabilidad para (0,0). También en este caso C es un conjunto abierto conectado en el plano de fase R^2 . Por el principio de máximo el controlador óptimo $u^*(t)$ es máximo y tiene un valor extremo

$$u^*(t) = \text{sgn } \eta_2(t)$$

casi por todas partes sobre $0 \leq t \leq t^*$. De la respuesta adjunta $\eta(t) = (\eta_1(t), \eta_2(t))$ satisface

$$\begin{aligned} \dot{\eta}_1 &= \pm k^2 \eta_2 \\ \dot{\eta}_2 &= -\eta_1 \pm 2b \eta_2 \end{aligned}$$

o

$$\ddot{\eta}_2 \mp 2b \dot{\eta}_2 \pm k^2 \eta_2 = 0$$

note que $\eta_2(t)$ no es idénticamente cero, en tal caso

$$\eta_1(t) = \pm 2b \eta_2(t) - \dot{\eta}_2(t) = 0,$$

qué contradice el requisito que $\eta(t)$ no se dispersa sobre $0 \leq t \leq t^*$. Puesto que $\eta_2(t)$ es analítica, tiene solo un número finito de ceros sobre $0 \leq t \leq t^*$. Ya que L es normal, hay un controlador máximo dirigiendo (x_0, y_0) a (0,0) y uno debe ser el controlador óptimo $u^*(t)$. Construiremos el apropiado cambio de $y = W(x)$ donde la respuesta extrema cambia entre los sistemas

$$(S_-) \quad \dot{x} = y \quad \dot{y} = \mp k^2 x \mp 2by - 1$$

y

$$(S_+) \quad \dot{x} = y \quad \dot{y} = \mp k^2 x \mp 2by + 1$$

Puesto que S_- se convierte en S_+ bajo la substitución $x \rightarrow -x$, $y \rightarrow -y$, encontraremos que $W(-x) = -W(x)$ y así el cambio de lugar necesitado sólo se encuentra para $x > 0$.

Así la síntesis del controlador óptimo $u^*(t)$ es reducida a la determinación del dominio C de controlabilidad y la construcción del cambio de lugar $y = W(x)$.

3

ALGORITMOS EVOLUTIVOS

3.1 UN VISTAZO HISTÓRICO A LA COMPUTACIÓN EVOLUTIVA.

La Vida Artificial consiste en observar la vida natural e imitarla en una computadora. La *Computación Evolutiva* interpreta la naturaleza como una inmensa máquina de resolver problemas y trata de encontrar el origen de dicha potencialidad para utilizarla en nuestros programas.

En la naturaleza todos los seres vivos se enfrentan a problemas que deben resolver con éxito, como conseguir más luz del sol, o cazar una mosca. Un programador con espíritu práctico no envidia la capacidad de la naturaleza para resolver problemas: la imita.

3.1.1 El origen de las ideas

Contrario a lo que muchos creen, las ideas evolucionistas que bien hiciera en popularizar Charles Darwin en 1858 no se originaron con el, sino que estuvieron presentes en las mentes de una serie de científicos y pensadores en general que no se sentían satisfechos con la (entonces popular) idea de que había un Dios originador de todas las especies del planeta (las cuales habían

sido creadas de forma separada) y de que las especies estaban jerarquizadas por Dios de tal manera que el hombre ocupaba el rango superior, al lado del Creador.

3.1.2 Russell y Darwin

El naturalista inglés Alfred Russell Wallace era un auto-didacta que se interesó en el origen de las especies a mediados de los 1850s, publicando varios artículos al respecto que pasaron totalmente desapercibidos. En 1858, de manera súbita intuyó la teoría de la selección natural sin saber que Darwin se le había adelantado, e irónicamente le escribió a este para pedirle que le ayudara a publicar sus ideas. El resultado de esta curiosa cooperación fue la presentación de un trabajo conjunto a la *Linnean Society* de Londres, el 1 de julio de 1858, el cual fue publicado posteriormente (ese mismo año) en el *Journal of the Linnean Society*.

Tras percatarse del trabajo de Wallace, Darwin decidió interrumpir la elaboración de un libro sobre la selección natural que inició en 1856, y mejor se enfocó a escribir otro sobre la evolución. Su libro, titulado *El origen de las especies*, se publicó en 1859 [2] con gran éxito (el tiraje inicial de 1,250 ejemplares se agotó en sólo 24 horas).

Darwin se percató de que una especie que no sufriera cambios se volvería incompatible con su ambiente, ya que este tiende a cambiar con el tiempo. Asimismo, las similitudes entre hijos y padres observada en la naturaleza, le sugirieron a Darwin que ciertas características de las especies eran hereditarias, y que de generación a generación ocurrían cambios cuya principal motivación era hacer a los nuevos individuos más aptos para sobrevivir.

3.1.3 La programación evolutiva

Lawrence J. Fogel et al. [3] concibieron el uso de la evolución simulada en la solución de problemas (sobre todo de predicción). Su técnica, denominada “Programación Evolutiva” [4] consistía básicamente en hacer evolucionar autómatas de estados finitos, los cuales eran expuestos a una serie de símbolos de entrada (el ambiente), y se esperaba que, eventualmente, serían capaces de predecir las secuencias futuras de símbolos que recibirían. Fogel utilizó una función de “pago” que indicaba que tan bueno era un cierto autómata para predecir un símbolo, y usó un operador modelado en la mutación para efectuar cambios en las transiciones y en los estados de los autómatas que tenderían a hacerlos más aptos para predecir secuencias de símbolos.

Esta técnica no consideraba el uso de un operador de recombinación sexual porque, pretendía modelar el proceso evolutivo al nivel de las especies y no al nivel de los individuos.

3.1.4 Las estrategias Evolutivas

Como estudiantes de postgrado en la Universidad Técnica de Berlín, en Alemania, Peter Bienert, Ingo Rechenberg y Hans-Paul Schwefel estudiaban la mecánica de los fluidos en 1963 con un particular énfasis en la experimentación en un túnel de viento. Los problemas que les interesaban eran de carácter hidrodinámico, y consistían en la optimización de la forma de un tubo curvo, la minimización del arrastre de una placa de unión y la optimización estructural de una boquilla intermitente de dos fases. Debido a la imposibilidad de describir y resolver estos problemas de optimización analíticamente o usando métodos tradicionales como el del gradiente [13], Ingo Rechenberg decidió desarrollar

un método de ajustes discretos aleatorios inspirado en el mecanismo de mutación que ocurre en la naturaleza.

En los dos primeros casos (el tubo y la placa), Rechenberg procedió a efectuar cambios aleatorios en ciertas posiciones de las juntas y en el tercer problema procedió a intercambiar, agregar o quitar segmentos de boquilla. Sabiendo que en la naturaleza las mutaciones pequeñas ocurren con mayor frecuencia que las grandes, Rechenberg decidió efectuar estos cambios en base a una distribución binomial con una varianza prefijada. El mecanismo básico de estos primeros experimentos era crear una mutación, ajustar las juntas o los segmentos de boquilla de acuerdo a ella, llevar a cabo el análisis correspondiente y determinar que tan buena era la solución. Si esta era mejor que su predecesora, entonces pasaba a ser utilizada como base para el siguiente experimento. De tal forma, no se requería información alguna acerca de la cantidad de mejoras o deterioros que se efectuaban.

Esta técnica tan simple dio lugar a resultados inesperadamente buenos para los tres problemas en cuestión, y Peter Bienert [6] construyó un robot que podía efectuar automáticamente el proceso de optimización usando este método. Simultáneamente, Hans-Paul Schwefel se dio a la tarea de implementar esta técnica en una computadora Z23 [16].

3.1.5 Los algoritmos genéticos

John H. Holland se interesó en los 1960s en estudiar los procesos lógicos involucrados en la adaptación. Inspirado por los estudios realizados en aquella época con autómatas celulares [20] y redes neuronales [21], Holland se percató de que el uso de reglas simples podría generar comportamientos

flexibles, y visualizó la posibilidad de estudiar la evolución de comportamientos en un sistema complejo.

Holland advirtió que un estudio de la adaptación debía reconocer que [10, 9]: (a) la adaptación ocurre en un ambiente, (b) la adaptación es un proceso poblacional, (c) los comportamientos individuales pueden representarse mediante programas, (d) pueden generarse nuevos comportamientos mediante variaciones aleatorias de los programas, y (e) las salidas de dos programas normalmente están relacionadas si sus estructuras están relacionadas.

De tal forma, Holland vio el proceso de adaptación en términos de un formalismo en el que los programas de una población interactúan y mejoran en base a un cierto ambiente que determina lo apropiado de su comportamiento. El combinar variaciones aleatorias con un proceso de selección (en función de que tan apropiado fuese el comportamiento de un programa dado), debía entonces conducir a un sistema adaptativo general.

Este sistema fue desarrollado hacia mediados de los 1960s, y se dio a conocer en el libro que Holland publicase en 1975, donde lo denominó “plan reproductivo genético” [11], aunque después se popularizó bajo el nombre (más corto y conveniente) de “algoritmo genético”.

Aunque concebido originalmente en el contexto del aprendizaje de máquina, el algoritmo genético se ha utilizado mucho en optimización, siendo una técnica sumamente popular en la actualidad.

3.2 PRINCIPALES PARADIGMAS

3.2.1 Introducción

El término **computación evolutiva** o **algoritmos evolutivos**, realmente engloba una serie de técnicas inspiradas biológicamente, que basándose en la imitación de varios procesos naturales que intervienen en la evolución de las especies (selección natural, mutaciones, sobrecruzamientos) tratan de resolver complejos problemas de búsqueda, optimización, aprendizaje, predicción o clasificación. (En los principios de la teoría Neo-Darwiniana de la evolución natural).

En términos generales, para simular el proceso evolutivo en una computadora se requiere:

- ✓ Codificar las estructuras que se replicarán.
- ✓ Operaciones que afecten a los “individuos”.
- ✓ Una función de aptitud.
- ✓ Un mecanismo de selección.

Aunque hoy en día es cada vez más difícil distinguir las diferencias entre los distintos tipos de algoritmos evolutivos existentes, por razones sobre todo históricas, suele hablarse de tres paradigmas principales:

- ✓ Programación Evolutiva
- ✓ Estrategias Evolutivas
- ✓ Algoritmos Genéticos

Cada uno de estos paradigmas se originó de manera independiente y con motivaciones muy distintas.

3.2.2 Programación Evolutiva

La programación evolutiva enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos (como en el caso de los algoritmos genéticos).

3.2.2.1 Algoritmo

El algoritmo básico de la programación evolutiva es el siguiente:

- ✓ Generar aleatoriamente una población inicial.
- ✓ Se aplica mutación.
- ✓ Se calcula la aptitud de cada hijo y se usa un proceso de selección mediante torneo (normalmente estocástico) para determinar cuales serán las soluciones que se retendrán.

La programación evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación (diferentes especies no se pueden cruzar entre si). Asimismo, usa selección probabilística.

3.2.2.2 Algunas aplicaciones de la programación evolutiva:

- ✓ Predicción
- ✓ Generalización
- ✓ Juegos
- ✓ Control automático
- ✓ Problema del viajero
- ✓ Planeación de rutas
- ✓ Diseño y entrenamiento de redes neuronales

- ✓ Reconocimiento de patrones

3.2.3 Estrategias Evolutivas

Las estrategias evolutivas fueron desarrolladas en 1964 en Alemania para resolver problemas hidrodinámicos de alto grado de complejidad por un grupo de estudiantes de ingeniería encabezado por Ingo Rechenberg [5].

3.2.3.1 Algoritmo

La versión original EE-(1+1) usaba un solo padre y con él se generaba un solo hijo. Este hijo se mantenía si era mejor que el padre, o de lo contrario se eliminaba (a este tipo de selección se le llama *extintiva*, porque los peores individuos obtienen una probabilidad de ser seleccionado de cero).

En la EE-(1+1), un individuo nuevo es generado usando:

$$\bar{x}^{t+1} = \bar{x}^t + N(0, \bar{\sigma})$$

donde t se refiere a la *generación* (o iteración) en la que nos encontramos, y $N(0, \bar{\sigma})$ es un vector de números Gaussianos independientes con una media de cero y desviaciones estándar $\bar{\sigma}$.

Muy pronto la EE-(1+1) fue sustituida por la versión multimiembro llamada EE-($\mu+1$) en la cual μ padres donde ($\mu > 1$) pueden participar en la generación del único hijo de una generación, que es entonces una mezcla de atributos de los padres, es decir que nace como resultado de la recombinación de la información transportada por sus progenitores. Esta estrategia es similar a la EE-(1+1) ya que ambas son de tipo “+”, lo que significa que la selección se aplica a un conjunto cuyos elementos son los padres y el hijo generado. Por lo tanto, si la aptitud del hijo no es mejor que la de alguno de los padres, el hijo se desecha y no forma parte de la siguiente generación.

La motivación para extender el modelo EE- $(\mu + 1)$, a los modelos EE- $(\mu + \lambda)$ y EE- (μ, λ) de varios miembros (donde μ denota el número de individuos de una población mientras que λ denota el número de hijos que estos padres producen en cada generación.), fue doble: primero, aprovechar las arquitecturas en paralelo, y segundo, permitir la auto-adaptación de los parámetros estratégicos, es decir, las desviaciones estándar de las mutaciones.

En estos dos modelos ya no se usa una regla externa determinística ("éxito de 1/5") para modificar las desviaciones estándar, sino que ahora éstas forman parte de la información de todos y cada uno de los individuos de la población y por lo tanto están sujetas a las operaciones de mutación y recombinación. De esta forma la modificación de las desviaciones estándar se convierte en un proceso auto-adaptable que se desarrolla en el ambiente de evolución artificial del algoritmo. Se espera que aquellos individuos que mejor adapten sus parámetros estratégicos tengan mejor valor de aptitud y que por lo tanto tengan alta probabilidad de ser seleccionados para poblar la siguiente generación.

Eventualmente estos individuos dominarán en la población y la auto-adaptación hará que surjan mejores parámetros estratégicos. La diferencia entre los modelos EE- $(\mu + \lambda)$ y EE- (μ, λ) estriba en la forma en la que los padres de una nueva generación son seleccionados.

En la EE- $(\mu + \lambda)$, los μ padres junto con sus λ hijos son reunidos en un solo conjunto del cual se seleccionan los μ individuos más aptos para formar la siguiente generación (nótese que cuando μ y λ tienen un valor de 1, el modelo se reduce a la versión más simple del algoritmo, llamada EE- $(1 + 1)$).

En la EE- (μ, λ) , con $(\lambda > \mu \geq 1)$, los μ individuos de la nueva generación son seleccionados exclusivamente de los λ hijos producidos, sin importar si son o no mejores que sus padres.

La mutación en las EEs realiza una especie de búsqueda escalando la colina cuando se combina con la selección. En los modelos EE- $(\mu + \lambda)$ y EE- (μ, λ) ,

cada variable objetivo usa su propia desviación estándar asignada para producir mutaciones o alteraciones a sus valores nominales (es decir no correlacionados) que producen desplazamientos a lo largo de los ejes del sistema de coordenadas.

3.2.3.2 Convergencia

Rechenberg [14] formuló una regla para ajustar la desviación estándar de forma determinística durante el proceso evolutivo de tal manera que el procedimiento convergiera hacia el óptimo.

Esta regla se conoce como la “regla del éxito 1/5”, y en palabras dice:

“La razón entre mutaciones exitosas y el total de mutaciones debe ser 1/5. Si es mayor, entonces debe incrementarse la desviación estándar. Si es menor, entonces debe decrementarse”.

Formalmente:

$$\sigma(t) = \begin{cases} \sigma(t-n)/c & \text{si } p_s > 1/5 \\ \sigma(t-n)*c & \text{si } p_s < 1/5 \\ \sigma(t-n) & \text{si } p_s = 1/5 \end{cases}$$

donde n es el número de dimensiones, t es la generación, p_s es la frecuencia relativa de mutaciones exitosas medida sobre intervalos de (por ejemplo) $10n$ individuos, y $c = 0.817$ (este valor fue derivado teóricamente por Schwefel). $\sigma(t)$ se ajusta cada n mutaciones.

Thomas Bäck [5] derivó una regla de éxito 1/7 para las (μ, λ) -EEs.

3.2.3.3 Auto-Adaptación

En las estrategias evolutivas se evoluciona no sólo a las variables del problema, sino también a los parámetros mismos de la técnica (es decir, las desviaciones estándar). A esto se le llama “auto-adaptación”).

Los padres se mutan usando las siguientes fórmulas:

$$\begin{aligned}\sigma'(i) &= \sigma(i) \times \exp(\tau' N(0,1) + \tau N_i(0,1)) \\ x'(i) &= x(i) + N(0, \sigma'(i))\end{aligned}$$

donde τ y τ' son constantes de proporcionalidad que están en función de n .

Los operadores de recombinación de las estrategias evolutivas pueden ser:

- ✓ **Sexuales:** el operador actúa sobre 2 individuos elegidos aleatoriamente de la población de padres.
- ✓ **Panmíticos:** se elige un solo padre al azar, y se mantiene fijo mientras se elige al azar un segundo padre (de entre toda la población) para cada componente de sus vectores.

Las estrategias evolutivas simulan el proceso evolutivo al nivel de los individuos, por lo que la recombinación es posible. Asimismo, usan normalmente selección determinística.

3.2.3.4 Estrategias Evolutivas vs Programación Evolutiva

La Programación Evolutiva usa normalmente selección estocástica, mientras que las estrategias evolutivas usan selección determinística.

Ambas técnicas operan a nivel fenotípico (es decir, no requieren codificación de las variables del problema).

La programación evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de

recombinación (diferentes especies no se pueden cruzar entre si). En contraste, las estrategias evolutivas son una abstracción de la evolución al nivel de un individuo, por lo que la recombinación es posible.

3.2.3.5 Aplicaciones

Algunas aplicaciones de las estrategias evolutivas son [15]:

- ✓ Problemas de ruteo y redes
- ✓ Bioquímica
- ✓ Óptica
- ✓ Diseño en ingeniería
- ✓ Magnetismo

3.2.4 Algoritmos Genéticos

Los algoritmos genéticos (denominados originalmente “planes reproductivos genéticos”) fueron desarrollados por John H. Holland a principios de los 1960s [9, 10], motivado por resolver problemas de aprendizaje de máquina.

3.2.4.1 Algoritmo

El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre el de la mutación (operador secundario), y usa selección probabilística.

El algoritmo básico es el siguiente:

- ✓ Generar (aleatoriamente) una población inicial.
- ✓ Calcular aptitud de cada individuo.
- ✓ Seleccionar (probabilísticamente) en base a aptitud.

- ✓ Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población.
- ✓ Ciclar hasta que cierta condición se satisfaga.

Para poder aplicar el algoritmo genético se requiere de los 5 componentes básicos siguientes:

- ✓ Una representación de las soluciones potenciales del problema.
- ✓ Una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio).
- ✓ Una función de evaluación que juegue el papel del ambiente, clasificando las soluciones en términos de su “aptitud”.
- ✓ Operadores genéticos que alteren la composición de los hijos que se producirán para las siguientes generaciones.
- ✓ Valores para los diferentes parámetros que utiliza el algoritmo genético (tamaño de la población, probabilidad de cruza, probabilidad de mutación, número máximo de generaciones, etc.)

3.2.4.2 Algoritmos Genéticos vs Otras Técnicas Evolutivas

El AG usa selección probabilística al igual que la Programación Evolutiva, y en contraposición a la selección determinística de las Estrategias Evolutivas.

El AG usa representación binaria para codificar las soluciones a un problema, por lo cual se evoluciona el genotipo y no el fenotipo como en la Programación Evolutiva o las Estrategias Evolutivas.

El operador principal en el AG es la cruce, y la mutación es un operador secundario. En la Programación Evolutiva, no hay cruce y en las Estrategias Evolutivas es un operador secundario.

Ha sido demostrado [12] que el AG requiere de elitismo (o sea, retener intacto al mejor individuo de cada generación) para poder converger al óptimo. Los AGs no son, normalmente, auto-adaptativos, aunque el uso de dicho mecanismo es posible, y ha sido explorado extensivamente en la literatura especializada. Puede verse una comparación más detallada de los tres paradigmas anteriores en la tabla 3.1.

	Estrategias Evolutivas	Programación Evolutiva	Algoritmo Genético
Representación	Real	Real	Binaria
Función de Aptitud	Valor de la función objetivo	Valor de la función objetivo ajustada	Valor de la función objetivo ajustada
Auto-Adaptación	Desviaciones Estándar y ángulos de rotación	Ninguna Varianzas (PE-estándar), Coeficientes de correlación (meta-PE)	Ninguna
Mutación	Gausiana, operador principal	Gausiana, operador único	Inversión de bits, operador secundario
Recombinación	Discreta e intermedia, sexual y panmítica, importante para la auto-adaptación	Ninguna	Cruza de 2-puntos, cruce uniforme, únicamente sexual, operador principal
Selección	Determinística, extintiva o basada en la preservación	Probabilística, extintiva	Probabilística, basada en la preservación
Restricciones	Restricciones arbitrarias de desigualdad	Ninguna	Límites simples mediante el mecanismo de codificación
Teoría	Velocidad de Convergencia para casos especiales, (1+1)-ES, (1+ λ)-ES, Convergencia global para ($\mu + \lambda$)-ES	Velocidad de Convergencia para casos especiales, (1+1)-PE, Convergencia global para meta-PE	Teoría de los Esquemas, Convergencia global para la versión elitista

Tabla 3.1: Tabla comparativa de los tres paradigmas principales que conforman la computación evolutiva.

3.2.4.3 Aplicaciones

Algunas aplicaciones de los AGs son las siguientes [8]:

- ✓ Optimización (estructural, de topologías, numérica, combinatoria, etc.)
- ✓ Aprendizaje de máquina (sistemas clasificadores)
- ✓ Bases de datos (optimización de consultas)
- ✓ Reconocimiento de patrones (por ejemplo, imágenes)
- ✓ Generación de gramáticas (regulares, libres de contexto, etc.)
- ✓ Planeación de movimientos de robots
- ✓ Predicción

3.2.5 Diferencias de las técnicas evolutivas con respecto a las tradicionales

Existen varias diferencias que vale la pena destacar entre los algoritmos evolutivos y las técnicas tradicionales de búsqueda y optimización [7, 8]:

- ✓ Las técnicas evolutivas usan una población de soluciones potenciales en vez de un solo individuo, lo cual las hace menos sensibles a quedar atrapadas en mínimos y/o máximos locales.
- ✓ Las técnicas evolutivas no necesitan conocimiento específico sobre el problema que intentan resolver.
- ✓ Las técnicas evolutivas usan operadores probabilísticos, mientras las técnicas tradicionales utilizan operadores determinísticos.
- ✓ Aunque las técnicas evolutivas son estocásticas, el hecho de que usen operadores probabilísticos no significa que operen de manera análoga a una simple búsqueda aleatoria.

3.2.6 Ventajas de las técnicas evolutivas

Es importante destacar las diversas ventajas que presenta el uso de técnicas evolutivas para resolver problemas de búsqueda y optimización [23, 22]:

- ✓ Simplicidad Conceptual.
- ✓ Amplia aplicabilidad.
- ✓ Superiores a las técnicas tradicionales en muchos problemas del mundo real.
- ✓ Tienen el potencial para incorporar conocimiento sobre el dominio y para hibridizarse con otras técnicas de búsqueda/optimización.
- ✓ Pueden explotar fácilmente las arquitecturas en paralelo.
- ✓ Son robustas a los cambios dinámicos.
- ✓ Generalmente pueden auto-adaptar sus parámetros.
- ✓ Capaces de resolver problemas para los cuales no se conoce solución alguna.
- ✓ No necesitan conocimientos específicos sobre el problema que intentan resolver.

Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales. Cuando se usan para problemas de optimización -maximizar una función objetivo resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales.

Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo. Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen -tamaño de la población,

número de generaciones, etc.-. Pueden converger prematuramente debido a una serie de problemas de diversa índole.

3.3 ¿Cómo saber si es posible usar Estrategias Evolutivas?

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y confiables. Sin embargo, no todos los problemas pudieran ser apropiados para la técnica, y se recomienda en general tomar en cuenta las siguientes características del mismo antes de intentar usarla:

Su espacio de búsqueda (i.e., sus posibles soluciones) debe estar delimitado dentro de un cierto rango. Debe poderse definir una función de aptitud que nos indique qué tan buena o mala es una cierta respuesta. Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en la computadora.

El primer punto es muy importante, y lo más recomendable es intentar resolver problemas que tengan espacios de búsqueda continuos -aunque éstos sean muy grandes.

La **función de aptitud** no es más que la función objetivo de nuestro problema de optimización. El algoritmo genético únicamente maximiza, pero la minimización puede realizarse fácilmente utilizando el recíproco de la función maximizante (debe cuidarse, por supuesto, que el recíproco de la función no genere una división por cero). Una característica que debe tener esta función es que debe ser capaz de "castigar" a las malas soluciones, y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor

rapidez. La codificación más común de las respuestas es a través de números reales y letras.

4

ALGORITMO DE SELECCIÓN AUTOMÁTICA DE PARAMETROS DE ESTABILIZACIÓN

4.1 PROPIEDADES DE MATRICES DEFINIDAS POSITIVAS

Cada forma cuadrática $f = ax^2 + 2bxy + cy^2$ tiene un punto estacionario en el origen, donde $\partial f / \partial x = \partial f / \partial y = 0$; si tiene un mínimo local en $x = y = 0$, entonces ese punto es también un mínimo global.

Condiciones para a , b y c aseguren que $f = ax^2 + 2bxy + cy^2$ es positivamente definida.

Condición necesaria

i. Si f es positivamente definida, entonces necesariamente $a > 0$.

y

ii. Si f es positivamente definida, entonces necesariamente $c > 0$

Queremos un criterio preciso, que proporcione condiciones necesarias y suficientes para la definitividad positiva, la técnica más sencilla es completar el cuadrado:

$$f = ax^2 + 2bxy + cy^2 = a\left(x + \frac{b}{a}y\right)^2 + \left(c - \frac{b^2}{a}\right)y^2 \quad (1)$$

Obviamente el primer término de la derecha es positivo, ya que es un cuadrado perfecto multiplicado por el coeficiente positivo a : todavía rige la condición necesaria (i). Sin embargo este cuadrado perfecto es cero cuando $x = b$ e $y = -a$; en este punto encontramos que $f(b, -a) = a(ac - b^2)$. Por lo tanto, se requiere otra condición necesaria:

iii. Si f es positiva, entonces necesariamente $ac > b^2$

Nótese que cuando se toman juntas las condiciones (i) y (iii), automáticamente implican la condición (ii): si $a > 0$ y $ac > b^2 \geq 0$, claramente $c > 0$. Se garantiza que el lado derecho de (1) es positivo, lo que nos lleva a:

La forma cuadrática $f = ax^2 + 2bxy + cy^2$ es positivamente definida si y solo si $a > 0$ y $ac - b^2 > 0$.

Reconozcamos la manera de colocar los coeficientes de f en una matriz simétrica A . los términos ax^2 y cy^2 aparecen en la diagonal y la derivada cruzada $2bxy$ se descompone en una entrada superior y una entrada inferior, la forma f es exactamente igual al producto matricial

$$ax^2 + 2bxy + cy^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

Esta identidad se puede escribir como $f = x^T Ax$; se generaliza inmediatamente a n dimensiones. Cuando hay n variables independientes x_1, \dots, x_n en lugar de solo x e y , van en un vector columna x . Entonces **para cualquier matriz simétrica A , el producto $f = x^T Ax$ es una forma cuadrática pura.** Tiene un punto estacionario en el origen y no tiene términos de orden superior:

$$\begin{aligned}
 x^T Ax &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{bmatrix} \\
 &= a_{11}x_1^2 + a_{12}x_1x_2 + a_{21}x_2x_1 + \cdots + a_{nn}x_n^2 \\
 &= \sum_{i=1}^n \sum_{j=1}^n a_{ij}x_ix_j.
 \end{aligned} \tag{3}$$

Dada una función $f(x_1, \dots, x_n)$ que tiene un punto estacionario en el origen (todas las primeras derivadas son cero). La función tiene un mínimo cuando f es positivamente definida o, en otras palabras, cuando la matriz A es positivamente definida; esto significa que:

$$x^T Ax > 0 \text{ Para toda } x \text{ distinta de } x_1 = x_2 = \cdots = x_n = 0 \tag{4}$$

Tenemos las siguientes condiciones en una matriz:

$$\text{Si } A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \text{ necesitamos } a > 0 \text{ y } ac - b^2 > 0.$$

El objetivo es generalizar estas condiciones a una matriz de orden n . al menos en el caso de 2 por 2 las condiciones significan que ambos valores propios son positivos: su producto es el determinante $ac - b^2 > 0$, así que los valores propios son ambos positivos o ambos negativos y debe de ser lo primero por que su suma es la traza $a + c > 0$.

Es posible detectar la aparición de los pivotes, surgen cuando descomponemos f en una suma de cuadrados:

$$ax^2 + 2bxy + cy^2 = a \left(x + \frac{b}{a}y \right)^2 + \frac{ac - b^2}{a}y^2$$

Los coeficientes a y $\frac{ac-b^2}{a}$ son exactamente los pivotes de una matriz de 2 por 2. Si esta relación es válida para matrices más grandes, admitirá un criterio muy sencillo para ser positivamente definida: verificar los pivotes. Tendrá al mismo tiempo una interpretación muy natural: $x^T Ax$ es positivamente si y solo si puede escribirse como una suma de n cuadrados independientes.

No es suficiente pedir que $\det A > 0$; esta condición se satisface cuando $a = c = -1$ y $b = 0$, lo cual da $A = -I$, que es en realidad negativamente definida. El punto importante es que el criterio del determinante se aplique no solo a A misma, lo que da $ac - b^2 > 0$, sino también a la submatriz a de 1 por 1 que está en la esquina superior izquierda. La generalización natural involucrará todas las n submatrices superiores izquierdas

$$A_1 = [a_{11}], \quad A_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad A_3 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \dots, \quad A_n = A.$$

Teorema: Cada uno de los siguientes criterios es una condición necesaria y suficiente para que la matriz simétrica real A sea *positivamente definida*: [25]

- I. $x^T Ax > 0$ para todos los vectores x distintos de cero.
- II. Todos los valores propios de A satisfacen $\lambda_i > 0$.
- III. Todas las submatrices A_k tienen determinantes positivos.
- IV. Todos los pivotes (sin intercambio de filas) satisfacen $d_i > 0$

A es positivamente definida si y solo si satisface la condición:

V. Existe una matriz no singular W tal que $A = W^T W$

Si esta nueva condición es válida entonces

$$x^T A x = x^T W^T W x = \|Wx\|^2$$

4.2 DEFINICIÓN DE OPERADORES DE ESTRATEGIAS EVOLUTIVAS

Tenemos una población A de individuos los cuales son manipulados por operadores genéticos, (especialmente mutación y recombinación, pero también pueden ser incorporados otros) y expuestos a un proceso de selección basado en su aptitud, donde la aptitud de un individuo depende de su calidad con respecto a la optimización de tareas. Todo esto se resume en la siguiente definición:

Definición 4.1 (Algoritmo Evolutivo General). Un algoritmo Evolutivo (AE) es definido como una 8-tupla

$$AE = (I, \Phi, \Omega, \Psi, s, t, \mu, \lambda) \quad (5)$$

Donde $I = A_x \times A_y$ es el espacio de individuos, y A_x, A_y denotan conjuntos arbitrarios. Y $\Phi : I \rightarrow R$ denota una función de aptitud asignando valores reales a los individuos.

$$\Omega = \{\omega_{\Theta_1}, \dots, \omega_{\Theta_z} \mid \omega_{\Theta_i} : I^\lambda \rightarrow I^\lambda\} \cup \{\omega_{\Theta_0} : I^\mu \rightarrow I^\lambda\} \quad (6)$$

Es un conjunto de operadores genéticos probabilístico ω_{Θ_i} , cada uno de los cuales es controlado por parámetros específicos resumidos en los conjuntos $\Theta_i \subset R$.

$$s_{\Theta_s} : (I^\lambda \cup I^{\mu+\lambda}) \rightarrow I^\mu \quad (7)$$

Denota el operador de selección, el cual puede cambiar el número de individuos de λ o $\lambda + \mu$ a μ , donde $\mu, \lambda \in N$ y $\mu = \lambda$ es permitido.

Un conjunto adicional Θ_s de parámetros puede ser usado por el operador de selección. μ es el número de padres individuales, mientras λ denota el número de descendientes individuales. Finalmente, $\iota: I^\mu \rightarrow \{true, false\}$ es un criterio de terminación para los AE, y la generación de la función de transición $\Psi: I^\mu \rightarrow I^\mu$ describe el proceso completo de transformación de una población P a una subsecuente aplicando operadores genéticos y selección:

$$\begin{aligned} \Psi &= s \circ \omega_{\Theta_{i_1}} \circ \dots \circ \omega_{\Theta_{i_j}} \circ \omega_{\Theta_0} \\ \Psi(P) &= s_{\Theta_s} \left(Q \cup \omega_{\Theta_{i_1}} \left(\dots \left(\omega_{\Theta_{i_j}} \left(\omega_{\Theta_0}(P) \right) \right) \dots \right) \right) \end{aligned} \quad (8)$$

Donde $\{i_1, \dots, i_j\} \subseteq \{1, \dots, z\}$ y $Q \in \{O, P\}$.

El espacio de individuos puede ser arbitrariamente complejo, i.e. no hay restricciones sobre las estructuras de los conjuntos A_x y A_s , aunque ellos serán relativamente simples en lo siguiente. Incluso la función de aptitud Φ puede incluir algunos pasos de cálculo intermedios, uno de estos puede ser la evaluación del valor de la función objetiva, la cual proporciona la base del valor de aptitud. Siempre que $\mu \neq \lambda$, el conjunto operador Ω incluye un operador distinguido $\omega_0: I^\mu \rightarrow I^\lambda$ el cual sirve para cambiar el tamaño de la población a μ individuos. Mientras los operadores genéticos son siempre probabilísticos, la selección puede ser probabilística o completamente determinística. Tanto la selección como los operadores genéticos pueden ser controlados por algunos parámetros exógenos, el criterio de terminación ι

puede variar arbitrariamente de criterios complicados a más simples. Un paso de una generación completa, i.e. la transición de la actual población de padres a la subsecuente, consiste en la aplicación de operadores genéticos en un orden definido seguidos por la selección. Este es capturado en la generación de la función de transición Ψ , la aplicación iterada genera una secuencia de población:

Definición 4.2 (Secuencia de población). Dado un Algoritmo Evolutivo con función de transición $\Psi: I^\mu \rightarrow I^\mu$ y una población inicial $P(0) \in I^\mu$, la secuencia $P(0), P(1), P(2), \dots$ es llamada una secuencia de población o evolución de $P(0)$: \Leftrightarrow

$$\forall t \geq 0 : P(t+1) = \Psi(P(t)) \quad (9)$$

La creación de la población inicial $P(0)$ será discutida después en conjunto con la explicación de circunstancias particulares de algoritmos evolutivos. Usualmente, $P(0)$ es inicializada aleatoriamente, pero también puede ser generada de un (conocido) punto de partida. El criterio de paro t caracteriza el final de este proceso artificial de evolución, y el resultado de una corrida de un Algoritmo Evolutivo es en la mayoría de los casos el mínimo valor de la función objetivo encontrada durante la evolución completa.

Este individuo no es necesariamente idéntico al mejor individuo incluido en la población final. La siguiente definición abrevia esta descripción del tiempo de corrimiento y resultado de un Algoritmo Evolutivo.

Definición 4.3. Dada una población inicial $P(0) \in I^n$ para un Algoritmo Evolutivo con función de transición Ψ , el tiempo de ejecución τ_{AE} es dado por

$$\tau_{AE} = \min \left\{ t \in \mathbb{N} \mid \Psi^t(P(0)) = true \right\} \quad (10)$$

Un individuo

$$\vec{a} \in \bigcup_{t=0}^{\tau_{AE}} \Psi^t(P(0)) \quad (11)$$

Es llamado el resultado de un Algoritmo Evolutivo cuando es aplicado a la población inicial $P(0)$ si y solo si

$$f(\vec{a}) = \min \left\{ f(\vec{a}') \mid \vec{a}' \in \bigcup_{t=0}^{\tau_{AE}} \Psi^t(P(0)) \right\} \quad (12)$$

Los operadores genéticos son caracterizados como macro-operadores que transforman una población completa a otra población completa. Esta descripción de alto-nivel es puesta en términos más concretos especificando el mecanismo el cual guía para la creación de un nuevo individuo de uno o más ancestros. Básicamente, los operadores de alto-nivel son reducidos a una descripción por operadores de bajo-nivel como son indicados en la siguiente definición:

Definición 4.4 (Operadores genéticos sexual, asexual, panmítico)

Un operador genético $\omega_{\Theta} : I^p \rightarrow I^q$ es llamado

Sexual $:\Leftrightarrow \exists \omega'_{\Theta} : I^2 \rightarrow I :$

$$\omega_{\Theta}(\vec{a}_1, \dots, \vec{a}_p) = (\omega'_{\Theta}(\vec{a}_{i_1}, \vec{a}_{j_1}), \dots, \omega'_{\Theta}(\vec{a}_{i_q}, \vec{a}_{j_q}))$$

Donde $\forall k \in \{1, \dots, q\} i_k, j_k \in \{1, \dots, p\}$

Son escogidos aleatoriamente,

$$\text{Asexual} \quad :\Leftrightarrow \exists \omega'_{\ominus} : I \rightarrow I : \quad (13)$$

$$\omega_{\ominus}(\bar{a}_1, \dots, \bar{a}_p) = (\omega'_{\ominus}(\bar{a}_1), \dots, \omega'_{\ominus}(\bar{a}_p)) \wedge p = q$$

$$\text{Panmítico} \quad :\Leftrightarrow \exists \omega'_{\ominus} : I^p \rightarrow I :$$

$$\omega_{\ominus}(\bar{a}_1, \dots, \bar{a}_p) = \underbrace{(\omega'_{\ominus}(\bar{a}_1, \dots, \bar{a}_p), \dots, \omega'_{\ominus}(\bar{a}_1, \dots, \bar{a}_p))}_q$$

La definición caracteriza el número esencial de individuos tomados en cuenta por los operadores. La mutación es un ejemplo de un operador asexual, mientras la recombinación es típicamente sexual, pero también puede ser extendida en algunas variantes de los Algoritmos Genéticos a una forma panmítica. Los símbolos m y r serán usados para denotar la descripción de alto-nivel de mutación y recombinación, respectivamente, mientras m' y r' denotarán su forma asexual, sexual o panmítica. La descripción dada se puede traducir a un Esquema de un Algoritmo Evolutivo. En el algoritmo 1 t denota el contador de la generación, $P(t) = \{\bar{a}_1(t), \dots, \bar{a}_{\mu}(t)\}$ es la población a generar t , que consta de los individuos $\bar{a}_i \in I$ y μ denota el tamaño de población de padres. $Q \in \{O, P(t)\}$ Denota un conjunto adicional de individuos (ej. La población de padres) este puede ser tomado en cuenta para la selección.

Algoritmo 1 (Esquema de un Algoritmo Evolutivo)

$t := 0$;

inicializar $P(0) := \{\bar{a}_1(0), \dots, \bar{a}_{\mu}(0)\} \in I^{\mu}$;

```

evaluar  $P(0) : \{\Phi(\vec{a}_1(0)), \dots, \Phi(\vec{a}_\mu(0))\};$ 
while  $(t(P(t)) \neq true)$  do
    recombinar:  $P'(t) := r_{\Theta_r}(P(t));$ 
    mutar:  $P''(t) := m_{\Theta_m}(P'(t));$ 
    evaluar:  $P''(t) : \{\Phi(\vec{a}_1''(t)), \dots, \Phi(\vec{a}_\lambda''(t))\};$ 
    seleccionar:  $P'(t+1) := s_{\Theta_s}(P''(t) \cup Q);$ 
     $t := t + 1;$ 
end.
    
```

4.3 Estrategias Evolutivas Representación y Evaluación de aptitud.

Los puntos de búsqueda en las Estrategias Evolutivas son n-dimensionales con parámetro de vector objeto $\vec{x} \in R^n$. Dada la función objetivo $f : R^n \rightarrow R$, la función de aptitud Φ es en principio idéntico a f , i.e. dado un individuo $\vec{a} \in I$, tenemos

$$\Phi(\vec{a}) = f(\vec{x}) \quad (14)$$

Aquí \vec{x} el componente de la variable objeto de $\vec{a} = (\vec{x}, \vec{\sigma}, \vec{\alpha}) \in I = R^n \times A_s$, donde

$$\begin{aligned}
 A_s &= R_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha} \\
 n_\sigma &\in \{1, \dots, n\} \\
 n_\alpha &\in \{0, (2n - n_\sigma)(n_\sigma - 1)/2\}
 \end{aligned} \quad (15)$$

Además la representación del vector de la variable objeto \vec{x} , cada individuo puede adicionalmente incluir n diferentes desviaciones estándar σ_i también

como $n \cdot (n-1)/2$ ángulos de rotación $\alpha_{ij} \in [-\pi, \pi]$ ($i \in \{1, \dots, n-1\}, j \in \{i+1, \dots, n\}$)⁵ Tal que el número máximo de las cantidades de los parámetros de la estrategia para $w = n \cdot (n+1)/2$. Para el caso $1 < n_\sigma < n$ la desviación estándar $\sigma_1, \dots, \sigma_{n_\sigma-1}$ son acoplados con las variables objeto x_{n_σ}, \dots, x_n .

El número n_α de ángulos de rotación depende directamente de n_σ , el número de desviaciones estándar y n , pero este puede también ser explícitamente el conjunto cero, indicando que esta parte del parámetro de estrategia de individuos no es usada.

El conjunto de parámetro de estrategia consiste de desviaciones estándar y ángulos de rotación que proporcionan una descripción completa de la generalizada n-dimensional distribución normal con la expectativa del valor del vector $\vec{0}$ teniendo, la función de densidad de probabilidad

$$p(\vec{z}) = \frac{\exp\left(-\frac{1}{2} \vec{z}^T C^{-1} \vec{z}\right)}{\sqrt{(2\pi)^n \cdot \det C}} \quad (16)$$

Aquí $C^{-1} = (c_{ij})$ es la matriz de covarianza con elementos en la diagonal $c_{ii} = \sigma_i^2$ es decir las variaciones. Schwefel clarifica que los ejes de la mutación de elipsoides (superficies de igual probabilidad de densidad para poner una descendiente por mutación) son paralelos a los ejes de la coordenada solo si C es una matriz diagonal. En el caso más general de covarianzas que no desaparecen, la mutación de elipsoides puede tener orientación arbitraria en el

espacio de búsqueda, y las mutaciones de variables de objeto están linealmente en correlación. De esta manera, los individuos pueden adaptar cualquier dirección ventajosa de búsqueda, la cual es útil por ejemplo en el caso de valles angostos donde una orientación de mutación de elipsoides a lo largo del valle es más inapropiada.

La figura 4.1 ilustra el caso más común que ocurre normalmente para $n = 2$. Se muestran cinco individuos y sus correspondientes mutaciones hiperelipsoides se muestran en una topología de la función objetivo hipotética caracterizada por líneas de igual valor de función objeto. La parte izquierda de la figura muestra la mutación de elipsoides (las cuales en este caso son esferas) para $n_\sigma = 1$, es decir, desviaciones estándar idénticas a lo largo de ambos ejes de la coordenada. Para $n_\sigma = 2$ (en medio gráfico) dos diferentes desviaciones estándar permiten la deformación de esferas a elipsoides de tal una manera que una preferencia de dirección puede ser que se encuentre. Esto es más fácil si, como se muestra en la parte derecha de la figura, una correlación lineal se introduce $n_\sigma = 2$, $n_\alpha = 1$ tal que las elipsoides pueden girar hacia una dirección de preferencia diferente a las coordenadas de los ejes. Los tres individuos localizados cerca de la izquierda superior, y derecha inferior pueden ahora lograr los pasos más grandes hacia el óptimo que en el caso de los no correlacionados.

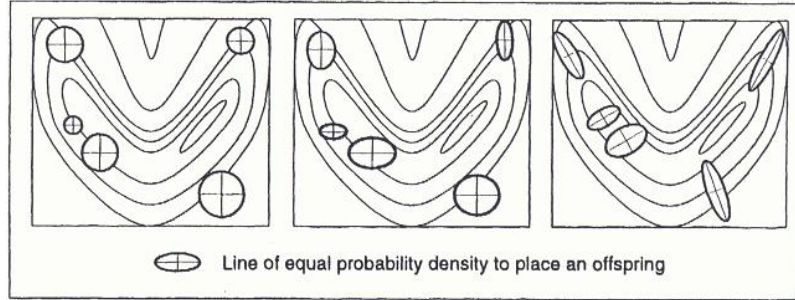


Fig. 4.1: la Ilustración de mutaciones de hiperelipsoides en el caso de mutaciones simples con $n_\sigma = 1$ (izquierda), $n_\sigma = 2$ (en medio) y mutaciones correlacionadas con $n_\sigma = 2$, $n_\alpha = 1$ (derecha). En cada grafico cinco individuos y sus elipsoides de igual de probabilidad de densidad para la posición de un descendiente se muestra sobre una topología hipotética que se representa por líneas de igual valor de la función objeto.

Sin embargo, no es aconsejable incorporar las covarianzas directamente dentro de la representación y aplicar mutación a ellas, porque es difícil garantizar que el sistema de coordenadas siga siendo ortogonal o, equivalentemente, la matriz de covarianzas permanezca positivamente definida. Esto es por qué los ángulos de rotación α_{ij} se usan por realizar los alineamientos de rotación de la mutación de elipsoides.

Estos ángulos se relacionan para covarianzas y varianzas por la ecuación:

$$\tan(2\alpha_{ij}) = \frac{2c_{ij}}{\sigma_i^2 - \sigma_j^2} \quad (17)$$

En lo siguiente, usaremos la notación $\bar{N}(\vec{0}, C)$ para denotar una realización de un vector aleatorio distribuido de acuerdo a la generalizada distribución normal

n-dimensional con expectativa $\vec{0}$ y matriz de covarianza $C^{-1} = C^{-1}(\vec{\sigma}, \vec{\alpha})$, representada por un vector $\vec{\sigma}$ de desviaciones estándar y $\vec{\alpha}$ ángulos de rotación. Algorítmicamente, la generación de una realización correlacionada $\vec{\sigma}_c$ de una no correlacionada $\sigma_u = \vec{N}(\vec{0}, \vec{\sigma})$ puede realizarse por la multiplicación de σ_u por n_α matrices de rotación $R(\alpha_{ij}) = (r_{kl})$. La forma de estas matrices se da por una matriz unidad excepto que para las entradas siguientes: $r_{ii} = r_{jj} = \cos \alpha_{ij}$, $r_{ij} = -r_{ji} = -\text{sen} \alpha_{ij}$, es decir, las expresiones trigonométricas se localiza en las columnas y renglones i y j , cada una.

La multiplicación por tal una matriz realiza una transformación de coordenada con respecto a los números de ejes i y j y ángulo α_{ij} .

Desde n_α rotaciones se necesitan para representar todas las correlaciones, la relación completa entre $\vec{\sigma}_c$ y $\vec{\sigma}_u$ resultan como

$$\vec{\sigma}_c = \left(\prod_{i=1}^{n-1} \prod_{j=i+1}^n R(\alpha_{ij}) \right) \cdot \sigma_u \quad (18)$$

Sólo unas líneas de código son suficientes para realizar estos cálculos [18], aunque la ecuación (18) actúa como un disuasivo. Las recientes investigaciones por Rudolph confirman la validez de este mecanismo, desde que él pudo mostrar que esto garantiza creación de transformaciones ortogonales y al mismo tiempo permite la creación de cualquier posible transformación ortogonal, es decir usando ángulos de la rotación no restringe la generalidad de mutaciones [24].

4.3.1 Mutación

Después de presentar tanta información sobre la representación esencial para las mutaciones, las explicaciones restantes para este operador son breves. Un mecanismo más general se discute aquí, es decir un espacio individual $I = R^n \times R^n \times R^{n(n-1)/2}$ se asume. La mutación $m_{\{\tau, \tau', \beta\}} : I^\lambda \rightarrow I^\lambda$ es un operador asexual, a partir de este momento basta presentar su forma reducida $m'_{\{\tau, \tau', \beta\}} : I \rightarrow I$ el cual proporciona un triple $(\bar{x}', \bar{\sigma}', \bar{\alpha}')$ cuando es aplicado a un individuo particular $(\bar{x}, \bar{\sigma}, \bar{\alpha})$. La anotación $N(0,1)$ se usa aquí para denotar una realización de una distribución normal de variable aleatoria uno-dimensional teniendo expectativa cero y desviación estándar uno, mientras $N_i(0,1)$ indica que la variable aleatoria se prueba nuevamente para cada posible valor del contador i . Usando esta anotación, la mutación se formaliza como sigue

$$\begin{aligned} \sigma'_i &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha'_j &= \alpha_j + \beta \cdot N_j(0,1) \\ x'_j &= \bar{x} + \bar{N}(\bar{0}, C(\bar{\sigma}', \bar{\alpha}')). \end{aligned} \tag{19}$$

Primero, las desviaciones estándar y ángulos de rotación se mutan usando multiplicativo, un proceso distribuido normalmente logarítmico en el caso de las desviaciones estándar y un aditivo, una variación distribuida normalmente en el caso de la rotación de ángulos. Finalmente, para la mutación un vector de variable objeto \bar{x} los vectores resultantes $\bar{\sigma}'$ y $\bar{\alpha}'$ se usa para crear un vector aleatorio para modificar \bar{x} . El factor global $\exp(\tau' \cdot N(0,1))$ permite un cambio global de mutabilidad y garantiza la preservación de todos los grados de

libertades, considerando que el $\exp(\tau \cdot N_i(0,1))$ permite al individuo cambios del “mean step size” σ_i . Una distribución normal logarítmica para las variaciones de desviaciones estándares σ_i se motiva como sigue:

- Un proceso de modificación de la multiplicación para σ_i que garantice valores positivos de desviaciones estándar.
- La mediana (1/2-quantil) de una modificación multiplicativa debe ser uno (esto implica la próxima condición en ser cumplida).
- Para garantizar un promedio neutral del proceso en la ausencia de presión selectiva, una multiplicación por un cierto valor debe ocurrir, con la misma probabilidad como la multiplicación por el valor recíproco.
- Las modificaciones más pequeñas deben ocurrir más a menudo que las más grandes.

Los factores τ, τ' , y β en la ecuación (19) son parámetros bastante robustos los cuales Schwefel sugiere para poner como sigue (vea [17], pág. 167-168):

$$\begin{aligned}\tau &\propto (\sqrt{2\sqrt{n}})^{-1} \\ \tau' &\propto (\sqrt{2n})^{-1} \\ \beta &\approx 0.0873\end{aligned}\tag{20}$$

Normalmente, las constantes de proporcionalidad τ y τ' tienen valor uno, y el valor sugerido para β (en radianes) es igual a 5° . τ y τ' pueden interpretarse en el sentido de “aprender proporciones” como en una red neural artificial, y experimentos preliminares con diferentes factores de proporcionalidad indican que

los procesos de búsqueda pueden ser afinados para particulares funciones objeto modificando estos factores.

Sin embargo, todavía es posible para desviaciones estándar llegar a ser prácticamente cero por el proceso multiplicativo para los ángulos de rotación salir del rango $[-\pi, \pi]$ de valores factibles. Para prevenir ambos eventos, el algoritmo en el primer caso fuerza todas las desviaciones estándar para permanecer más grande que el valor mínimo ε_σ y en el segundos caso los ángulos son circularmente trazados al rango factible, es decir siempre que un ángulo se volviera una cantidad c_α más grande (o más pequeño) que $\pi(-\pi)$, es trazada a $-\pi + c_\alpha(\pi - c_\alpha)$:

$$\begin{aligned} |\alpha'_j| > \pi &\Rightarrow \alpha'_j := \alpha'_j - 2\pi \cdot \text{sign}(\alpha'_j) \\ \sigma'_i < \varepsilon_\sigma &\Rightarrow \sigma'_i := \varepsilon_\sigma \end{aligned} \quad (21)$$

En conjunto, este mecanismo de mutación especial permite a una Estrategia Evolutiva para desenvolver sus propios parámetros de estrategia, es decir desviaciones estándar y covarianzas (representadas por ángulos de rotación) durante la búsqueda, aprovechándose de un eslabón implícito entre el modelo interior apropiado y los buenos valores de aptitud. La evolución resultante y adaptación de parámetros de estrategia de acuerdo a los requisitos topológicos han sido estipulados colectivos *misma-adaptación* por Schwefel [19].

4.3.2 Recombinación

Una variedad de diferentes mecanismos de recombinación se usa actualmente en las Estrategias de Evolutivas, y los operadores son sexuales como panmicticos. En la forma sexual, el operador de recombinación actúa en dos individuos escogidos aleatorios de la población de padres, dónde escogiendo el mismo individuo para la creación de un individuo descendente no es suprimido (pero este podría ser introducido fácilmente). Recíprocamente, para las variantes panmicticas de recombinación un padre es escogido al azar y retenido mientras para cada componente de sus vectores el segundo padre es al azar nuevamente escogido de la población completa. En otros términos, la creación de un solo individuo descendiente puede involucrar a todos los individuos padre.

La recombinación siempre se usa en las Estrategias Evolutivas para la creación de todos los descendientes cuando $\mu > 1$. Además, no sólo las variables objeto también los parámetros de estrategia están sujetos a precombinación, y el operador de recombinación puede ser diferente para las variables objeto, desviaciones estándar y ángulos rotación. Esto implica que la recombinación de estos grupos proporcione beneficios de información independientemente de cada uno, es decir no hay ninguna restricción para los parámetros de estrategia para originar del mismo padre las variables objeto. La utilización de recombinación independiente en las variables objeto y parámetros de estrategia (las desviaciones estándar y los ángulos de rotación) está justificado por las observaciones experimentales acerca de la actuación de las variantes resultantes de Estrategias Evolutivas. Las investigaciones teóricas de este tema son aun un campo abierto de investigación.

Los diferentes operadores de recombinación tradicionales de Estrategias Evolutivas son llamados *recombinación discreta* y *recombinación intermedia* ambos existiendo en forma sexual y panmictica. En el caso de recombinación discreta para cada componente de los vectores se decide al azar de los cuales ambos padres (en el caso sexual) el componente se copia al descendiente. La variante panmictica escoge un nuevo padre de la población de padres entera. La recombinación intermedia indica que los componentes del descendiente se obtienen calculando la media aritmética de los componentes correspondientes de ambos padres. Para la variante panmictica un padre se escoge al azar y se retiene mientras para cada componente el segundo compañero de la unión es nuevamente escogido de una nueva población.

Recientemente, Schwefel propuso para generalizar la recombinación intermedia permitiendo factores de peso arbitrarios del intervalo $[0,1]$ además del valor 0.5 y además escogiendo el factor de peso nuevamente para cada componente en caso de la recombinación intermedia panmictica. Aunque esta generalización que será llamada como *recombinación intermedia generalizada* en lo siguiente, promete ser útil, no se ha probado todavía extensivamente.

En orden para formalizar lo que se ha explicado hasta ahora, las formas sexual y panmictica se resumen aquí para la creación de la recombinación en un individuo $\vec{a}' = (\vec{x}', \vec{\sigma}', \vec{a}')$ de una población $P(t) \in I^\mu$, es decir $r' : I^\mu \rightarrow I$. Asumimos implícitamente la forma sexual $r' : I^2 \rightarrow I$ para actuar en dos padres al azar sin distinguirlo formalmente, y presentar representativamente las reglas para las variables objeto ($\forall i \in \{1, \dots, n\}$):

$$x'_i = \begin{cases} x_{S,i} & \text{no recombinación} & r'_- \\ x_{S,i} \text{ o } x_{T,i} & \text{discreta} & r'_d \\ x_{S,i} \text{ o } x_{T_i,i} & \text{panmítica discreta} & r'_D \\ x_{S,i} + (x_{T,i} - x_{S,i})/2 & \text{intermedia} & r'_i \\ x_{S,i} + (x_{T_i,i} - x_{S,i})/2 & \text{panmítica intermedia} & r'_I \\ x_{S,i} + X \cdot (x_{T,i} - x_{S,i}) & \text{generalizada intermedia} & r'_g \\ x_{S,i} + X_i \cdot (x_{T_i,i} - x_{S,i}) & \text{panmítica generalizada intermedia} & r'_G \end{cases}$$

Los índices S y T denotan dos padres seleccionados al azar de la población (el índice i en T_i indica T para ser probado nuevamente para cada valor de i), $X \in [0,1]$ es una variable aleatoria uniforme, probada nuevamente para cada posible valor del contador i cuando se usó en la forma X_i y se probó solo una vez por la creación de un nuevo descendiente sin índice y “o” denota una decisión justa hecha por un volado.

Fijando $X = 1/2$ y $X_i = 1/2$ ($\forall i \in \{1, \dots, n\}$), respectivamente, la recombinación intermedia generalizada se reduce a recombinación intermedia. Desde que el operador de recombinación completo el resultado de la combinación de operadores de componentes-sabios (para variables objeto, desviaciones estándar y ángulos de rotación), un operador de recombinación está en lo siguiente denotado por $r_{r_x r_\sigma r_\alpha}$ donde $r_x, r_\sigma, r_\alpha \in \{-, d, D, i, I, g, G\}$ indica el mecanismos de recombinación usado en los tres diferentes componentes de individuo. Esto resulta en $7^3 = 343$ operadores diferentes. Para el caso bidimensional los posibles resultados de la recombinación se muestran esquemáticamente en la figura 4.2.

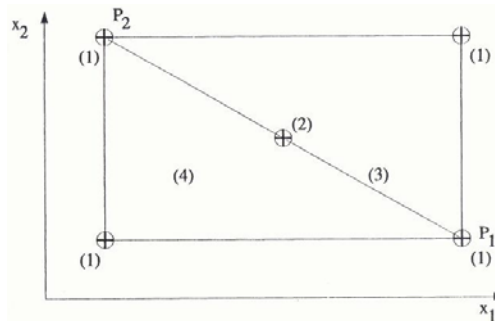


Fig. 4.2: Esquema 2-dimensiones de diferentes mecanismos de recombinación.

Aquí P_1 y P_2 indican los puntos de la variable objeto representados por dos padres. Sólo las esquinas (indicadas (1) en la figura) del rectángulo definido por P_1 y P_2 puede alcanzarse por medio de la recombinación discreta. El rendimiento de la recombinación intermedia el centro (2) de las líneas diagonales del rectángulo. La recombinación intermedia generalizada permite un resultado localizado en cualquier parte en la línea diagonal (3) conectando P_1 y P_2 , y finalmente recombinación intermedia generalizada panmictica permite la creación de un punto arbitrario localizado en alguna parte dentro del rectángulo (4), dónde todavía se restringen los padres a P_1 y P_2 . Sin embargo, es importante notar que por medio de la recombinación el hiper cuerpo n-dimensional formado por la población de padres nunca puede dejar de ser un descendiente, es decir la recombinación causa más o menos una reducción de volumen.

Schwefel informa una aceleración notable del proceso de búsqueda por introducir la recombinación ([17], pág. 171) y la recombinación identifica parámetros estratégicos como un prerrequisito necesario para facilitar la misma adaptación de parámetros estratégicos [19]. Siguiendo sus sugerencias la recombinación discreta sobre variables objeto y (panmictica) y recombinación

intermedia de parámetros estratégicos podrían ser preferidos, es decir r_{III} . Para explicar para explicar el efecto de la aceleración de la recombinación de variables objeto, Schwefel contó el número de posibles resultados de recombinación en el caso de recombinación discreta y panmictica el cual se presenta en el teorema siguiente, completado por el número de resultados de recombinación de recombinación intermedia (panmictica).

Teorema 1 (Número de posibles resultados de recombinación)

Dada una población de padres de μ individuos, cada uno con diferentes variables objeto n-dimensional, el número posible de combinaciones de los diferentes resultados de la recombinación de las variables objeto es

$$\begin{aligned}
 &\mu + \binom{\mu}{2} (2^n - 2) && \text{para recombinacion discreta} \\
 &\mu^n && \text{para recombinacion discreta panmictica} && (22) \\
 &\mu \cdot (\mu + 1) / 2 && \text{para recombinacion intermedia} \\
 &(\mu \cdot (\mu + 1) / 2)^n && \text{para recombinacion intermedia panmictica}
 \end{aligned}$$

Teorema 2 (Recombinación Discreta Generalizada)

Dada un población de padres de μ individuos que tienen cada uno diferentes variables objeto n-dimensional con componentes sabios, y se asume que $1 \leq \rho \leq \mu$ individuos pueden escogerse para producir un descendiente por medio de recombinación discreta, el número $R(\mu, \rho, n)$ de posibles combinaciones de resultados de la recombinación se dan por

$$R(\mu, \rho, n) = \sum_{i=1}^{\rho} \binom{\mu}{i} i! S(n, i), \tag{23}$$

Donde

$$S(n, i) = \frac{1}{i!} \sum_{j=0}^i (-1)^j \binom{i}{j} (i-j)^n \quad (24)$$

Denota la segunda clase de números Stirling

4.3.3 Selección

Los operadores de selección usados en las Estrategias Evolutivas son completamente deterministas. Schwefel introduce una anotación elegante para los mecanismos de selección en las Estrategias Evolutivas, caracterizando el método Básico y los números de padres y descendientes, respectivamente. El distingue entre una $(\mu + \lambda)$ -selección $s_{(\mu+\lambda)} : I^{\mu+\lambda} \rightarrow I^\mu$ y una (μ, λ) -selección $s_{(\mu,\lambda)} : I^\lambda \rightarrow I^\mu$. La primera función selecciona los mejores μ individuos de la unión de padres y descendientes para formar la próxima generación de padres, mientras que la segunda función selecciona los mejores μ individuos de los descendientes solamente. Más formalmente, esto puede ser expresado como sigue:

$s_{(\mu+\lambda)}(\mathbf{P}) = \mathbf{P}'$ tal que

$$\exists \vec{a}_k \in \mathbf{P} - \mathbf{P}' : (\exists \vec{a}_i \in \mathbf{P}' : f(\vec{x}_k) < f(\vec{x}_i)) \quad (25)$$

(el caso $s_{(\mu,\lambda)}$ es idéntico, excepto que \mathbf{P} consiste solamente de la población de descendientes).

A primera vista, el $(\mu + \lambda)$ -selección con su supervivencia garantizada de los mejores individuos parece ser más eficaz, porque un curso monótono de la evolución se logra de esta manera. Sin embargo, este mecanismo de selección

tiene varias desventajas cuando lo comparamos con la (μ, λ) -selección, la cual restringe vidas de individuos a una generación:

- En caso de los ambientes cambiantes la $(\mu + \lambda)$ -selección conserva (anticuado) las soluciones y no puede ser capaz para seguir el movimiento óptimo. (Claro, una posibilidad de engañar este problema es reevaluar la población de padres también, pero ésta no es parte del algoritmo original de Estrategias Evolutivas y podría introducir una sobrecarga de μ evaluaciones por generación en contraste con la (μ, λ) -estrategia.)
- La capacidad de una (μ, λ) -selección para olvidarse de soluciones buenas en el principio permite dejar un óptimo local (pequeño) y por consiguiente la desventaja en caso de las topologías multimodales.
- la $(\mu + \lambda)$ -selección impide el mecanismo de mismo-adaptación con respecto a los parámetros de la estrategia para trabajar efectivamente, porque parámetros de la estrategia desadaptados pueden sobrevivir por un número relativamente grande de generaciones cuando ellos causan una mejora de aptitud por el cambio.

Resumiendo las condiciones para una misma-adaptación exitosa de parámetros de la estrategia, la lista siguiente se obtiene:

- Una (μ, λ) -estrategia es requerida para facilitar la extinción de individuos mal adaptados.
- La presión selectiva no puede llegar a ser demasiado fuerte, es decir, u es requerida para ser claramente más grande que uno.

- La Recombinación sobre los parámetros de la estrategia es necesaria (usualmente, la recombinación intermedia da mejores resultados).

Finalmente, anticipándose a una discusión más íntima de la importancia de la selección en Algoritmos Evolutivos, mencionamos el hecho que la razón u/k proporciona el instrumento de parametrización básica para controlar el carácter de la búsqueda. Decrementando u enfatiza una búsqueda orientada-a-la-ruta y la velocidad de convergencia, mientras que incrementando u nos lleva a una búsqueda orientada-a-volumen.

4.3.4 Otros Componentes

En la implementación original Schwefel condiciona la posibilidad de crear una completa población inicial $P(0) = \{\vec{a}_1(0), \dots, \vec{a}_\mu(0)\}$ desde un único punto de partida inicial por medio de mutación. Esto fue hecho para lograr comparabilidad con los métodos tradicionales de optimización. El punto de inicio se puede definir por el usuario del algoritmo, o se puede escoger de una región factible.

Para la inicialización de la desviación estándar $\sigma_i(0)$, Schwefel recomendó la elección $\sigma_i(0) \approx \Delta x_i / \sqrt{n}$, donde Δx_i sería la distancia estimada entre el punto de inicio y el óptimo. La experiencia práctica sugiere que la elección de la desviación estándar inicial tiene que ser $\sigma_i(0) = 3$. Ya que dependiendo de la topología de la función objeto, una combinación de $\sigma_i(0)$ y un valor muy grande de μ puede causar que el algoritmo diverja.

Finalmente es probado comparando el peor y mejor valor de aptitud de la población de padres. Cualquiera absoluta o relativa.

$$t\{\bar{a}_1(t), \dots, \bar{a}_\mu(t)\} = \begin{cases} \text{true, } \max\{\Phi(\bar{a}_i(t))\} - \min\{\Phi(\bar{a}_i(t))\} \leq c(P(t)) \\ \text{falso, de lo contrario} \end{cases} \quad (26)$$

Aquí,

$$c(P(t)) = \varepsilon_{\Phi,1} > 0 \quad (27)$$

En caso de la media absoluta, y

$$c(P(t)) = \frac{\varepsilon_{\Phi,2}}{\mu} \cdot \left| \sum_{i=1}^{\mu} \Phi(\bar{a}_i(t)) \right| \quad (28)$$

En caso de la media relativa, donde $\varepsilon_{\Phi,1} > 0$ y $\varepsilon_{\Phi,2} > 0$ denota constante.

4.3.5 Algoritmo Estándar Conceptual

El siguiente algoritmo conceptual de $(\mu + \lambda)$ -EE y (μ, λ) -EE es:

Algoritmo 2. (Contorno de una Estrategia Evolutiva)

$t := 0;$

inicializa $P(0) := \{\bar{a}_1(0), \dots, \bar{a}_\mu(0)\} \in I^\mu$

donde $I = R^n \times R_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha}$

y $\bar{a}_k = (x_i, \sigma_j, \alpha_{j'}), i \in \{1, \dots, n\}, j \in \{1, \dots, n_\sigma\},$

$j' \in \{1, \dots, n_\alpha\}$

o $\bar{a}_k = (x_i, \sigma_j), i \in \{1, \dots, n\}, j \in \{1, \dots, n_\sigma\}$

evaluar $P(0): \{\Phi(\bar{a}_1(0)), \dots, \Phi(\bar{a}_\mu(0))\}$

donde $\Phi(\bar{a}_k(0)) = f(\bar{x}_k(0))$

While ($\iota(P(t)) \neq true$) do
 recombinar: $\bar{a}_k(t) := r'(P(t)) \quad \forall k \in \{1, \dots, \lambda\}$;
 mutar: $\bar{a}_k''(t) := m'_{\{r, r', \beta\}}(\bar{a}_k'(t)) \quad \forall k \in \{1, \dots, \lambda\}$;
 evaluar: $P''(t) := \{\bar{a}_1''(t), \dots, \bar{a}_\lambda''(t)\}$;
 $\{\Phi(\bar{a}_1''(t)), \dots, \Phi(\bar{a}_\lambda''(t))\}$
 Donde $\Phi(\bar{a}_k''(t)) = f(\bar{x}_k''(t))$;
 seleccionar: $P(t+1) := \text{if}(\mu, \lambda)$ -selección
 then $s_{(\mu, \lambda)}(P''(t))$;
 else $s_{(\mu + \lambda)}(P(t) \cup P''(t))$;
 $t := t + 1$;
 end

Definición 4.5 (Estrategia Evolutiva) Un Algoritmo Evolutivo

$$AE = (I, \Phi, \Omega, \Psi, s, t, \mu, \lambda)$$

Es llamado Estrategia Evolutiva: \Leftrightarrow

- (1) $I = R^n \times R_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha}$, donde $n_\sigma \in \{1, \dots, n\}$ y $n_\alpha \in \{0, (2n - n_\sigma)(n_\sigma - 1)/2\}$
- (2) $\Phi = f$
- (3) $\Omega = \{m_{\{r, r', \beta\}} : I^\lambda \rightarrow I^\lambda\} \cup \{r_{r_x r_\sigma r_\alpha} : m_{\{r, r', \beta\}} : I^\mu \rightarrow I^\lambda\}$ donde los operadores se definen como en (19) y (22)
- (4) $\Psi(P) = s(Q \cup m_{\{r, r', \beta\}}(r_{r_x r_\sigma r_\alpha}(P)))$ donde $Q = 0$ en caso de (μ, λ) -selección y $Q = P$ en caso que $(\mu + \lambda)$ -selección,

- (5) $s \in \{s_{(\mu+\lambda)}, s_{(\mu,\lambda)}\}$, el operador de la selección, se define como en (25),
- (6) el criterio de terminación ι es el dado por ejemplo por (26), y
- (7) si $s = s_{(\mu,\lambda)}$ entonces $1 \leq \mu \leq \lambda$ es requerido (por otra parte, no hay restricciones especiales en la relación entre μ y λ).

Esta definición no incorpora variantes tempranas de estrategia tales como (1+1)-ES y $(\mu+1)$ -ES, pero resume todas las estrategias misma-adaptación al menos un parámetro de la estrategia. Las estrategias más simples usan a un operador de mutación completamente diferente y por consiguiente no se discuten aquí.

Siempre que ninguna parametrización diferente se mencione explícitamente, una Estrategia Evolutiva Estándar según esta descripción es asumida en lo siguiente. Si una descripción exacta de una Estrategia Evolutiva que se desvía de la norma se requiere, se escribirá en el formato $ES(n_\sigma, n_\alpha, r_{r,\sigma,r_\alpha}, s \in \{s_{(\mu,\lambda)}, s_{(\mu+\lambda)}\})$ por ejemplo, la anotación $ES(n, 0, r_{dl}, s_{(15,100)})$ denota la Estrategia Evolutiva estándar. Desde el punto de vista de un practicante que es confrontado con un problema de optimización de características topológicas completamente desconocidas, es aconsejable realizar experimentos que siguen un tipo de "jerarquía default" desde simples hacia algoritmos complejos en aumento. Tal jerarquía de Estrategias Evolutivas se da por

- (1) $ES(1, 0, r_{dl}, s_{(15,100)})$
- (2) $ES(n, 0, r_{dl}, s_{(15,100)})$
- (3) $ES(2, 1, r_{dl}, s_{(15,100)})$

$$(4) ES\left(n, n \cdot (n-1)/2, r_{all}, s_{(15,100)}\right)$$

Y proporciona una posibilidad de mejorar los resultados probando las estrategias más complejas a costa del tiempo de computación. La sugerencia del caso (3) requiere alguna explicación adicional. Su utilización está justificada por la idea que dos desviaciones estándar permiten el ajuste de una dirección preferida de búsqueda donde la desviación estándar es relativamente grande comparado con el n-1 direcciones restantes de desviaciones estándar constantes. El ángulo de rotación entonces permite una orientación arbitraria de esta dirección preferida. Tal estrategia podría ser muy útil para una topología donde la búsqueda puede proceder a lo largo de un valle con un fondo relativamente llano y paredes empinadas.

4.4 Algoritmo propuesto

El algoritmo propuesto utiliza entonces el planteamiento de Estrategias Evolutivas (EE) para la selección de parámetros para los algoritmos de estabilización. Las EE tiene las siguientes características para la optimización de parámetros de sistemas dinámicos: (a) son Buenos para la optimización de parámetros de valores reales [29]; (b) usualmente son usados con auto-adaptación de parámetros de mutación [29], esto es deseado porque la superficie del “fitness” no es conocida y puede cambiar (de acuerdo con variaciones de pruebas de datos característicos); (c) tiene bajo tiempo de término [29], esta característica permite una rápida adaptación a el momento característico.

Las EE utilizan perturbación gaussiana en la mutación con auto adaptación de la desviación estándar y una desviación estándar por cada

variable en el sistema dinámico a ser optimizado. Esta clase de mutación es adecuada para optimización de parámetros numéricos ordinales.

A continuación se dan las definiciones de las operaciones de recombinación y mutación, y se demostrará que estas operaciones preservan las propiedades de las matrices definidas positivas. La sección concluye con el algoritmo evolutivo [30].

Definición 4.6 (Representación y Evaluación del *Fitness*):

Son matrices simétricas definidas positivas, esto es:

$$\Xi = \left\{ G \mid G \subseteq \mathfrak{R}^n \times \mathfrak{R}^n, x^T G x > 0 \quad \forall x \neq 0 \quad x \in \mathfrak{R}, G = G^T \right\} \quad (29)$$

Dada la función objeto $\mathfrak{F}: \Xi \rightarrow \mathfrak{R}$, la función de aptitud Φ es en principio idéntica a \mathfrak{F} , i.e. dado un individuo $\vec{a} \in I$, tenemos

$$\Phi(\vec{a}) = \mathfrak{F}(G) \quad (30)$$

Aquí G es la variable objeto componente de $\vec{a} = (G, \vec{\sigma}, \vec{\alpha}) \in I = \Xi \times A_s$, donde

$$\begin{aligned} A_s &= \mathfrak{R}_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha} \\ n_\sigma &\in \{1, \dots, n\} \\ n_\alpha &\in \{0, (2n - n_\sigma)(n_\sigma - 1)/2\} \end{aligned} \quad (31)$$

Además de representar a la variable objeto G , cada individuo pueda adicionalmente incluir un más que n diferentes desviaciones estándar σ_i así como $n \cdot (n-1)/2$ ángulos de rotación $\alpha_{ij} \in [-\pi, \pi]$ ($i \in \{1, \dots, n-1\}, j \in \{i+1, \dots, n\}$), tal que el número máximo de parámetros de estrategias $\omega = n \cdot (n+1)/2$. Para el caso

$1 < n_\sigma < n$, la desviación estándar $\sigma_1, \dots, \sigma_{n_\sigma-1}$ se acopla con las variables objeto $G_{1,1}, \dots, G_{n_\sigma-1, n_\sigma-1}$ y σ_{n_σ} es usada para la variable que queda $G_{n_\sigma, n_\sigma}, \dots, G_{n,n}$.

Definición 4.7 (Mutación): El espacio de los individuos $I = \Xi \times \mathfrak{R}^n \times \mathfrak{R}^{n \cdot (n-1)/2}$ es asumido. Mutación $m_{\{\tau, \tau', \beta\}}: I^\lambda \rightarrow I^\lambda$ es un operador asexual, es una tres-tupla $\vec{a}' = (G', \vec{\sigma}', \vec{\alpha}')$ cuando es aplicado a un individuo particular $\vec{a} = (G, \vec{\sigma}, \vec{\alpha})$. Entonces la mutación se formaliza como sigue:

$$\begin{aligned}\sigma'_i &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha'_j &= \alpha_j + \beta \cdot N_j(0,1) \\ G &= G + \vec{N}(\vec{0}, C(\vec{\sigma}', \vec{\alpha}'))\end{aligned}\quad (32)$$

Los factores τ , τ' , y β en la ecuación (32) son parámetros bastante robustos, que Schwefel sugiere presentar como sigue [17]:

$$\begin{aligned}\tau &\propto \left(\sqrt{2\sqrt{n}}\right)^{-1} \\ \tau' &\propto \left(\sqrt{2n}\right)^{-1} \\ \beta &\approx 0.0873\end{aligned}\quad (33)$$

Usualmente, la constante de proporcionalidad para τ y τ' tienen valor uno, y el valor sugerido para β (en radianes) igual 5° . La $C(\vec{\sigma}', \vec{\alpha}')$ es la matriz de covarianza con elementos diagonal $c_{ii} = \sigma_i^2$, i.e. las varianzas.

Finalmente, para la mutación del objeto variable G el vector resultante $\vec{\sigma}'$ y $\vec{\alpha}'$ son usados para crear el objeto aleatorio para modificación G , i.e. f para las propiedades de G , esta se descompone en $G = U\Lambda U^T$, así la operación de mutación: $\forall \lambda_i \in \Lambda$ (eigenvalores) con $i = 1, 2, \dots, n$, nos lleva a $\vec{\lambda}' = \vec{\lambda} + \vec{N}(\vec{0}, C(\vec{\sigma}', \vec{\alpha}'))$

tal que $\vec{\lambda}' > 0$, de esta manera tenemos $G' = U\Lambda'U^T$ donde Λ' los contiene λ_i para $i = 1, 2, \dots, n$, y la nueva matriz $G' = G + \vec{N}(\vec{0}, C(\vec{\sigma}', \vec{\alpha}'))$ es el resultado de la mutación que sufren sus eigenvalores de la matriz original G .

Definición 4.8 (Recombinación) Se asume un espacio de individuos $I = \Xi \times \mathfrak{R}^n \times \mathfrak{R}^{n \cdot (n-1)/2}$. La Recombinación $r_{r_x, r_\sigma, r_\alpha} : I^\mu \rightarrow I^\lambda$ es un operador sexual. Entonces la recombinación tiene forma sexual, actúa sobre dos individuos escogidos aleatoriamente de la población de padres, donde se escoge al mismo individuo dos veces para la creación de un individuo descendiente que no es suprimido. Entonces, sabemos que G se descompone en $G = U\Lambda U^T$ y $G' = U'\Lambda'U'^T$ es otra variable objeto, entonces podemos expresar el operador de recombinación como: $r_{r_x, r_\sigma, r_\alpha}(G_1, G_2) = U_j \Lambda_i U_j^T$ donde Λ_i representa un cambio de un subconjunto de n elementos seleccionados de un grupo de $2n$ elementos estos elementos son $|\Lambda_1 \cup \Lambda_2| = 2n$, de esta manera $i = 1, 2, \dots, \frac{(2n)!}{(2n-n)!}$ y j es alternado por cada cambio ocurrido en el grupo $\{1, 2\}$.

Ahora, cuando se obtuvo una nueva población de sus predecesores, se debe tener mucho cuidado si las operaciones de recombinación y mutación no alteraron la genética de los, una opción es verificar en cada paso del algoritmo, que genera una nueva población, si hubo alteraciones, otra opción, la cual se describe en este trabajo, es demostrar que estas operaciones no generan una nueva población, evitando verificar en todos los pasos del algoritmo y

mejorando el campo del mismo, esto es, mejoramos la complejidad computacional.

Proposición 1: La operación de recombinación preserva las propiedades de las matrices definidas positivas.

Prueba: Sea $X \subset \Xi$, para toda $A \in X$, se puede $A = U\Lambda U^T$, y entonces la rotación $y = U^T x$ produce la suma de los cuadrados

$$\begin{aligned} x^T A x &= x^T U \Lambda U^T x = y^T \Lambda y \\ &= \lambda_1 y_1^2 + \lambda_2 y_2^2 + \dots + \lambda_n y_n^2 \end{aligned} \quad (34)$$

Ahora, de la ecuación (8) los λ_i se sustituyen por $\lambda'_i > 0 \quad \forall i = 1, 2, \dots, n$

$$\begin{aligned} \lambda'_1 y_1^2 + \lambda'_2 y_2^2 + \dots + \lambda'_n y_n^2 &= y^T \Lambda' y = \\ &= x^T U \Lambda' U^T x = x^T A' x \end{aligned} \quad (35)$$

Donde $\lambda'_i > 0$ son reales, entonces A' es definida positiva. Sin pérdida de generalidad, uno puede pensar que los nuevos valores $\lambda'_i > 0$ son cambios ocurridos en la operación de cruce para generar la nueva matriz C_j por $j = 1, 2, \dots, \frac{(2n)!}{(2n-n)!}$. Razón porque, concluimos que la operación de recombinación

preserva las propiedades de definida positiva simétrica

Proposición 2: La operación mutación conserva las propiedades de las matrices definidas positivas.

Prueba: sea $X \subset \Xi$, para toda $A \in X$, entonces todos los valores característicos de la matriz A son positivos, esto es $\lambda_i > 0$, para $\forall i = 1, 2, \dots, n$, supongamos que para cada $\lambda_i > 0$ tiene el correspondiente vector propio unitario,

$$A x_i = \lambda_i x_i \text{ tal que}$$

$$x_i^T A x_i = x_i^T \lambda_i x_i = \lambda_i \quad (36)$$

Desde, $x_i^T x_i = 1$ y $x^T A x > 0 \quad \forall x \neq 0$, esto será válido en particular para el vector propio x_i y la cantidad $x_i^T A x_i = \lambda_i$. Ahora, de la ecuación (10) añadimos aquellos λ_i , $\beta_i \in N(0,1)$ esto es $\gamma_i = \lambda_i + \beta_i$, tal que $\gamma_i > 0$, entonces se tiene lo siguiente

$$0 < \lambda_i + \beta_i = x_i^T (\lambda_i + \beta_i) x_i = x_i^T A' x_i \quad (37)$$

De esta manera, la suma de un número aleatorio $\beta_i \in N(0,1)$, no afecta la propiedad de definida positiva de la matriz original A original, o que la nueva matriz es definida positiva.

Una (μ, λ) -estrategia se requiere para facilitar la extinción de individuos inadaptados. La presión selectiva puede no ser muy fuerte, i.e., μ . La recombinación sobre los parámetros de la estrategia son necesarios (usualmente, la recombinación intermedia da mejores resultados).

Algoritmo

$t := 0$

Inicializar $P(0) := \{\bar{a}_1(0), \dots, \bar{a}_\mu(0)\} \in \Xi \times \mathfrak{R}^n \times \mathfrak{R}^{n \cdot (n-1)/2}$

Evaluar $P(0) : \{\Phi(\bar{a}_1(0)), \dots, \Phi(\bar{a}_\mu(0))\}$ donde $\Phi(\bar{a}) = \mathfrak{F}(G)$

Mientras $(t(P(t)) \neq true)$ **hacer**

Recombinar: $\bar{a}'_k(t) := r'(P(t)) \quad \forall k \in \{1, \dots, \lambda\}$

Mutar: $\bar{a}''_k(t) := m'_{\{\tau, \epsilon, \beta\}}(\bar{a}'_k(t)) \quad \forall k \in \{1, \dots, \lambda\}$

Evaluar: $P''(t) : \{\Phi(\bar{a}'_1(t)), \dots, \Phi(\bar{a}''_\lambda(t))\}$

Seleccionar: $s_{(\mu, \lambda)}(P''(t))$

$t := t + 1$

FIN

5

PRUEBAS

5.1 Ejemplo

Consideramos el siguiente modelo:

$$\ddot{x} = u_1 \quad u_1 = -k_1 x_1 - k_2 \dot{x}_1 \quad (15)$$

Deseamos resolver el problema

$$\max_{|x(t_0)| \leq \nu} \int_{t_0}^{\infty} (x_1^2 + \dot{x}_1^2 + \ddot{x}_1^2) dt \rightarrow \min_{u_1 \in U} \quad (16)$$

Cuyo sentido físico es el: dada la peor condición inicial, queremos minimizar la desviación del sistema, así como la velocidad y aceleración de estas desviaciones.

En este caso

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad K^T = (k_1, k_2) \quad x^T = (x_1, \dot{x}_1) \quad (17)$$

El especialista experto escoge la siguiente matriz definida positiva

$$G^T = G = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix} \quad N = 1 \quad (18)$$

En este ejemplo, el punto silla existente, satisface lo siguiente

$$\min_{K \in Q} \max_{|x(t_0)| \leq \nu} \mathfrak{J} \quad \max_{|x(t_0)| \leq \nu} \min_{K \in Q} \mathfrak{J} \quad \text{donde}$$

$$\mathfrak{J} = \int_{t_0}^{\infty} (\dot{x}_1^2 + \ddot{x}_1^2) dt \quad (19)$$

Después verificando el sistema (9) es completamente controlable, procedemos al cálculo del problema interno, el cual es completamente controlable, procedemos al cálculo del problema interno [2], esto es:

$$\min_{K \in \mathcal{Q}} \mathfrak{J} = \min_{K \in \mathcal{Q}} \int_{t_0}^{\infty} (\dot{x}_1^2 + \ddot{x}_1^2) dt = x^{oT}(t_0) L_0 x^o(t_0) \quad (20)$$

Por lo tanto, el problema externo $\max_{|x(t_0)| \leq \nu} x^{oT}(t_0) L_0 x^o(t_0) = \mu_{\max}$ es resuelto. Así un valor optimo para el funcional es $1 + \sqrt{3}$ el cual se obtuvo de la peor condición inicial. La grafica se muestra abajo.

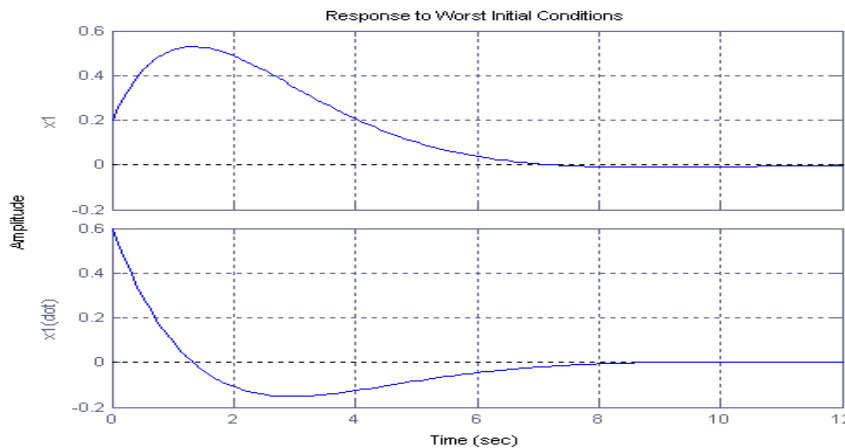


Fig. 2. Comportamiento asintótico de la solución cuando la matriz definida G ha sido fijada, elegida por un especialista humano

Ahora presentamos el resultado de aplicar el algoritmo basado en Estrategias Evolutivas, para elegir automáticamente la mejor matriz definida positiva G sin la intervención del experto.

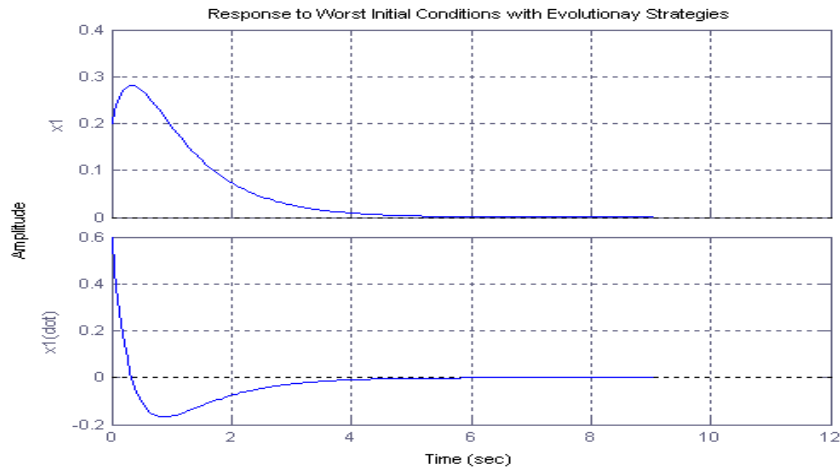


Fig 3: Resultado del comportamiento de las coordenadas del sistema, para la mejor opción de la matriz ponderada, dada en la peor condición inicial

Cuando el funcional no cambia en la siguiente generación, significa que tenemos al mejor individuo, es decir, la mejor matriz definida positiva G , la cual cuando se aplicó el funcional, redujo nuestro tiempo de estabilización, como se muestra en el siguiente grafico.

Las dos propuestas demuestran que las operaciones de recombinación y mutación preservan las estructuras de las matrices definidas positivas. Significa que dentro de cada iteración del algoritmo no hay necesidad de verificar si la nueva generación pertenece a su especie o no. En otras palabras, el nuevo individuo generado por una mutación o una recombinación tiene las características de ser una matriz definida positiva. Esto evita incrementar la complejidad en tiempo del algoritmo basado en Estrategia Evolutivas.

El algoritmo basado en Estrategias Evolutivas mejora la elección hecha por el especialista, así obtenemos la autonomía completa de los algoritmos de prueba en la computadora, aún cuando ha sido probado por medio de un ejemplo, podemos extender su uso para incluir sistemas mas robustos por medio del concepto del concepto y construcción de dicho algoritmo. Comparando las graficas 2 y 3, podemos observar que la estabilidad del sistema dinámico mejora.

6

CONCLUSIONES

A continuación se dan los resultados más importantes de la tesis:

1. Se realizó la investigación de los algoritmos evolutivos, se estudiaron las diferencias las cuales ayudaron a justificar la utilización de las estrategias evolutivas como la mejor opción.
2. La teoría de control optimal presenta en su metodología que se debe de minimizar un funcional, para encontrar la mejor estrategia de control o el control óptimo, el cual depende de dos matrices definidas positivas como parámetros importantes del funcional. La elección de las matrices lo hace el especialista ingeniero, el cual no tiene ninguna metodología para su elección. El planteamiento es la elección de las matrices definidas positivas de manera autónoma. Las estrategias evolutivas nos dan un procedimiento para hacer esta elección sin la intervención del especialista ingeniero.
3. Se definieron las operaciones de cruza, mutación y la función de aptitud.
4. Se demostró que la operación de cruza conserva la propiedad de la matriz definida positiva, esto es, dada dos matrices definidas positivas o

dos individuos, su cruce resultará un nuevo individuo de su propia especie o una nueva matriz definida positiva.

5. Se demostró que la operación de mutación conserva la propiedad de la matriz definida positiva, esto es, dada una matriz definida positiva o un individuo, la mutación que sufre para la nueva generación resulta ser un nuevo individuo de su propia especie o una nueva matriz definida positiva.
6. Con las dos proposiciones anteriores se obtiene una mejora en la complejidad del algoritmo, ya que no se necesita comprobar si los individuos de la nueva generación siguen perteneciendo a su especie, y esto implica una reducción de código en la programación y por ende una reducción en la complejidad del tiempo.
7. Se comprobó la eficiencia del algoritmo evolutivo con un problema de maxmin en la teoría del control óptimo, esta clase de problemas sugiere encontrar el mejor control de un sistema dinámico para las peores perturbaciones, por lo que representa un planteamiento completo para demostrar que la solución es satisfactoria.
8. Para el ejemplo el algoritmo evolutivo hizo 100 iteraciones para encontrar los mejores individuos o las mejores matrices definidas positivas, las cuales encontraron la mejor estrategia de control que el especialista ingeniero. Es importante resaltar que las estrategias evolutivas evitan que el ingeniero intente repetidas veces la elección de los parámetros en busca del mejor control, dependiendo del problema estos intentos pueden ser bastantes y aún así las estrategias evolutivas

pueden mejorar dicha elección, en el entendido que la elección pertenecen a un óptimo local.

BIBLIOGRAFIA

- [1] Adam Marczyk. Algoritmos genéticos y computación evolutiva, <http://the-geek.org/docs/algen>, 2004.
- [2] Charles Robert Darwin. On the origin of species by means of natural selection or the preservation of Favoured races in the Struggle for life. Cambridge University Press, Cambridge, UK, sixth edition, 1964. Originally published in 1859.
- [3] Lawrence J. Fogel, A. J. Owens, and M. J. Walsh. Artificial Intelligence through a Simulation of Evolution. In M. Maxfield, A. Callahan, and L.J. Fogel, editors, Biophysics and Cybernetic Systems: Proceedings of the Second Cybernetic Sciences Symposium, Pages 131-155. Spartan Books, Washington, D.C., 1965.
- [4] Lawrence J. Fogel, Artificial Intelligence through a Simulation of Evolution. John Wiley, New York, 1966.
- [5] Thomas Bäck. Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.
- [6] Peter Bienert. Aufbau einer optimierungsautomatik für drei parameter. Dipl.-Ing. thesis, 1967. (in German).

- [7] Bill P. Buckles and Frederick E. Petry, editors. Genetic Algorithms. Technology series. IEEE Computer Society Press, 1992. [15] Gilbert Strang. Algebra Lineal y sus Aplicaciones, Addison-Wesley Iberoamericana, 1988.
- [8] G. Gibson. Application of Genetic Algorithms to Visual Interactive Simulation Optimization. PhD thesis, University of South Australia, 1995.
- [9] John H. Holland. Concerning efficient adaptive systems. In M. C. Yovits G. T. Jacobi, and G. D. Goldstein, editors, Self-Organizing Systems-1962, pages 215-230. Spartan Books, Washington, D. C. 1962.
- [10] John H. Holland. Outline for a logical theory of adaptive systems. Journal of the Association for Computing Machinery, 9:297-314, 1962
- [11] John H. Holland. Adaptation in natural an Artificial Systems. University of Michigan Press, Ann Arbor, Michigan, 1975.
- [12] C.C. Palmer. An aprrtoach to a problem in network designusing genetic algorithms. PhD thesis, polytechnic University, Troy, New York, 1994.
- [13] Singiresu S. Rao. Engineering Optimization. Theory and Practice. John Wiley & Sons, Inc., third edition, 1996.
- [14] Ingo Rechenberg. Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart, Alemania, 1973.
- [15] Hans Paul Schwefel. Numerical Optimization of Compute Models. Wiley, Chichester, UK, 1981.
- [16] Hans Paul Schwefel. Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. Dipl.Ing.

thesis, 1965 (in German).

- [17] Hans Paul Schwefel. Numeric Optimierung von Computer-Modellen mittels der Evolutionsstrategie, volume 26 of Interdisciplinary systems Research. Birkhäuser, Basel, 1977.
- [18] Hans Paul Schwefel. Subroutines EVOL, GRUP, KORR, listings and user's guide. Internet Bericht KFA-STE-IB-2/80, Kernforschungsanlage Jülich, Programm-gruppe Systemforschung und Technologische Entwicklung, April 1980.
- [19] Hans Paul Schwefel. Collective phenomena in evolutionary systems. In Preprints of the 31st Annual meeting of the International Society for General System Research, Budapest, volume 2, pages 1025-1033, June 1987.
- [20] A. W. Burks. Computation, behavior and structure in fixed and growing automata. In M. C. Yovits and S. Cameron, editors, Self-Organizing Systems, pages 282-309. Pergamon Press, New York, 1960.
- [21] O.G. Selfridge. Pandemonium: a paradigm for learning. In Proceedings of the Symposium on Mechanization of Thought Processes, pages 511-529, Teddington,, England, 1958.
- [22] David B. Fogel. Evolutionary Computation. Toward a new Philosophy of Machine Intelligence. The Institute of Electrical and Electronic Engineers, New York, 1995.
- [23] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.
- [24] G. Rudolph. On correlated mutations in evolution strategies. In Männer and Manderick
- [25] Gilbert Strang. Algebra Lineal y sus Aplicaciones, Addison-

Wesley Iberoamericana, 1988.

- [26] Vadim F. Krotov, *Global Methods in Optimal Control Theory*, Marcel Dekker, Inc. New York (1996)
- [27] Knowles Greg, *An Introduction to Applied Optimal Control*, Academic Press, New York (1981)
- [28] Bullo Francesco and Lewis D. Andrew, *Geometric Control of Mechanical Systems: Modeling, Analysis and Design for Simple Mechanical Control Systems*, Springer.
- [29] Eiben, A.E., Smith, J. E. *Introduction to Evolution Computing*. Berlin: Springer, 2003.
- [30] Héctor Vargas, Vittorio Zanella, Vladimir Alexandrov, Mónica López, Lorna Rosas, “Estimation of symmetric positive definite of matrices using evolution strategy for differential games”, *Journal of Computer* ISSN:1870–4069 , pp. 189-198, México 2006.

.