

Optimización Multiobjetivo Utilizando el Principio de Selección Clonal del Sistema Inmune

Nareli Cruz Cortés y Carlos A. Coello Coello

CINVESTAV-IPN

Depto. Ing. Eléctrica, Sección de Computación

Av. Instituto Politécnico Nacional No. 2508

Col. San Pedro Zacatenco, México, D.F. 07300, MEXICO

nareli@computacion.cs.cinvestav.mx

ccoello@cs.cinvestav.mx

Resumen

Los sistemas bio-inspirados han sido utilizados con éxito en la solución de problemas de ingeniería de gran complejidad. Uno de los sistemas bio-inspirados más recientes es el Sistema Inmune Artificial, inspirado, como su nombre lo indica en nuestro sistema inmune. En este trabajo se reporta una aplicación de un sistema inmune artificial para solucionar problemas de optimización numérica con varios objetivos (con y sin restricciones). Los resultados obtenidos son competitivos con otras técnicas representativas del estado del arte del área de computación evolutiva multiobjetivo.

Palabras Clave: Sistema inmune artificial, optimización multiobjetivo, manejo de restricciones, selección clonal.

I Introducción

La mayoría de los problemas de optimización del mundo real tienen varios objetivos que deben satisfacerse de manera simultánea. Además, gran parte de ellos suelen tener restricciones difíciles de cumplir.

A pesar de la gran cantidad de algoritmos propuestos para optimización multiobjetivo [1], aún existen varios problemas abiertos en el área como la manera de incorporar restricciones y cómo obtener una buena distribución de las soluciones, entre algunos otros.

Las técnicas poblacionales (como los algoritmos evolutivos) suelen ser las más adecuadas para solucionar este tipo de problemas, debido a que con ellas se pueden explorar diferentes regiones del espacio de búsqueda de manera simultánea, así como encontrar varias soluciones a la vez.

El sistema inmune artificial es un área emergente que toma ideas del sistema inmune para solucionar problemas complejos. La principal motivación para usar este sistema como fuente de inspiración es debido a que presenta características de procesamiento de información muy interesantes, tales como: reconocimiento de patrones, memoria, aprendizaje, detección distribuida, tolerancia a fallas, comportamiento global complejo a partir de simples interacciones loca-

les, y paralelismo, entre otras.

Las emulaciones del sistema inmune toman ideas tanto del comportamiento global del sistema, como de algún mecanismo en particular.

Aunque ésta es un área relativamente nueva y el sistema inmune biológico no es totalmente comprendido (aún por los inmunólogos), existe en la literatura especializada del área una cantidad importante de aplicaciones a problemas diversos de aprendizaje de máquina e ingeniería principalmente. Entre ellos podemos mencionar [2]: robótica, reconocimiento de patrones, seguridad en sistemas de cómputo y redes, detección de anomalías, y optimización (mono-objetivo y multimodal), entre otras.

En este trabajo se presenta una emulación del sistema inmune artificial, basado en su principio de selección clonal, para solucionar problemas de optimización multiobjetivo con y sin restricciones. El algoritmo es conceptualmente simple y arroja resultados prometedores.

II Definición del problema

El problema que deseamos solucionar, es el general de optimización multiobjetivo y se define como sigue [1]:

Encontrar el vector $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ tal que satisfaga las m restricciones de desigualdad

$$(1) \quad g_i(\vec{x}) \geq 0 \quad i = 1, 2, \dots, m$$

las p restricciones de igualdad

$$(2) \quad h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p$$

y optimice el vector de funciones

$$(3) \quad \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$$

donde $\vec{x} = [x_1, x_2, \dots, x_n]^T$ es el vector de variables de decisión.

Las restricciones dadas por (1) y (2) definen la región factible Ω y cualquier punto \vec{x} en Ω define una solución factible.

Un concepto importante es el de dominancia de Pareto, el cual fue propuesto por Vilfredo Pareto [3] en el siglo XIX y constituye el origen del área de optimización multiobjetivo. Formalmente se define como sigue:

Decimos que $\vec{f}(\vec{x})$ domina a $\vec{f}(\vec{y})$ si:

$$(4) \quad \vec{f}(\vec{x}) \leq \vec{f}(\vec{y})$$

pero $\vec{f}(\vec{x}) \neq \vec{f}(\vec{y})$. Se denota como $\vec{f}(\vec{x}) \prec \vec{f}(\vec{y})$.

Decimos que un punto \vec{x}^* es **óptimo de Pareto** si no existe $\vec{x} \in \Omega$ tal que $\vec{x} \prec \vec{y}^*$.

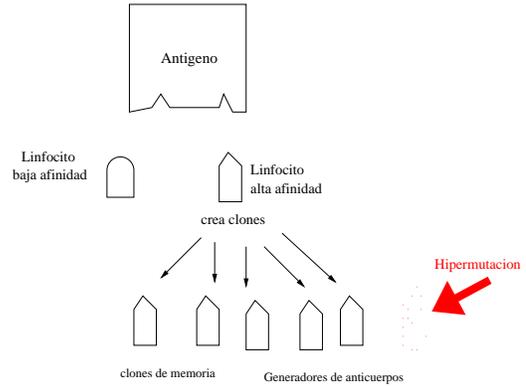


Figura 1: Principio de selección clonal

III Sistema Inmune Artificial

El principal objetivo del sistema inmune es mantener al organismo sano, de allí su gran importancia pues nuestra vida depende de su adecuado funcionamiento. El sistema inmune es capaz de detectar la invasión de agentes externos que lo puedan dañar (llamados *antígenos*), siendo capaz de distinguir éstos de las células y moléculas que le son propias, evitando así un posible autoataque. La respuesta inmune es diferente para cada tipo de antígeno por lo que se dice que ésta es específica.

El sistema inmune puede recordar antígenos que se han presentado en el pasado, de manera que es capaz de aprender y ante la aparición repetida de un mismo antígeno (o uno similar) su respuesta es más rápida y eficiente (*respuesta secundaria*)

El sistema inmune artificial es un nuevo paradigma de inteligencia computacional inspirado en el sistema inmune, que trata de emular alguno de sus mecanismos o su comportamiento global para solucionar problemas de aprendizaje máquina principalmente.

Nuestra propuesta está basada en uno de los mecanismos del sistema inmune llamado *principio de selección clonal* que a continuación se describe de manera muy general.

Principio de Selección Clonal

El principio de selección clonal fue propuesto por MacFarlane Burnett en 1978 [4], y explica el proceso por medio del cual el sistema inmune realiza el proceso de eliminación de los antígenos invasores.

Los principales actores del sistema inmune son las células llamadas *linfocitos*. Existen dos tipos principales de linfocitos: linfocitos B o células B y linfocitos T o células T. El principio de selección clonal se efectúa en ambos tipos de células. Sin embargo, por simplicidad, únicamente nos referiremos a las células B, pues éstas son grandes productoras de *anticuerpos*, cuya función es ir directamente a adherirse a los antígenos para neutralizarlos y eliminarlos.

Cuando ha sido detectada la presencia de algún antígeno, únicamente aquellas células B que poseen mayor afinidad hacia tal antígeno serán activadas. Las células que han sido activadas se reproducen por medio de clonación. Algunos de los nuevos clones se diferencian en *anticuerpos*, mientras que otros permanecerán circulando en el organismo por un largo tiempo como células de memoria. Ver figura 1.

Adicionalmente, estas células sufren un proceso de mutación de gran magnitud o hipermutación.

Después de que los anticuerpos han cumplido con la tarea de eliminación del antígeno, el sistema inmune debe regresar a sus niveles normales eliminando el excedente de células.

A todo este proceso se le conoce como *principio de selección clonal*.

Nuestra propuesta toma ideas de este principio, como se verá en el algoritmo descrito a continuación:

IV Algoritmo propuesto

Nuestro algoritmo está basado en el principio de selección clonal del sistema inmune el cual se complementa con algunos operadores de los algoritmos genéticos. Los linfocitos son las soluciones potenciales del problema que se representan mediante cadenas binarias (a la manera de los algoritmos genéticos [5]). A continuación se describe:

1. La población inicial se crea de manera que los individuos (linfocitos) estén distribuidos a lo largo del espacio de búsqueda de manera uniforme.
2. Inicializar la memoria secundaria (externa) de manera tal que esté vacía.
3. Determinar para cada individuo de la población, si es dominado (en el sentido de Pareto) o no y, para el caso de problemas con restricciones, determinar si es factible o no.

4. Determinar cuáles son los mejores linfocitos, de acuerdo al siguiente criterio:
Para un espacio sin restricciones: Todos los individuos no dominados son los mejores.
Para un espacio restringido:
 - Si existen individuos factibles: Se selecciona a los no dominados (con respecto al conjunto de individuos factibles)
 - Si no existe ningún individuo factible: Todos los individuos no dominados.
5. Copiar a los mejores linfocitos (resultantes del paso anterior) en la memoria secundaria.
6. Se determina para cada “mejor linfocito” la cantidad de clones que se deben crear, de manera tal que el total de clones sea igual al 60% de la población.
Además, si la memoria secundaria está saturada, se aplica el siguiente criterio:
 - Si al individuo no se le permitió ingresar a la memoria secundaria por ser repetido o pertenecer a la celda más saturada, entonces se crean cero clones.
 - Si el individuo pertenece a una celda de la memoria secundaria, tal que su población es menor al promedio (con respecto a la cantidad de individuos en cada celda de la memoria secundaria), entonces la cantidad de clones se duplica.
 - Si el individuo pertenece a una celda con población mayor al promedio (con respecto a todas las celdas), entonces la cantidad de clones es dividida entre 2.
7. Se realiza el proceso de clonación de los mejores linfocitos de acuerdo a la información creada en el paso anterior.
8. Se aplica el operador de mutación a los clones. El número de genes que se mutan de cada cadena cromosómica, es igual a la cantidad de variables existentes.
9. A aquellos linfocitos que no resultaron ser los mejores, se les aplica el operador de mutación uniforme. El porcentaje de mutación inicial es alto y se va decrementando a lo largo de las generaciones de manera lineal (desde 0.9 hasta 0.3)
10. Si la memoria secundaria está saturada, se aplica cruza a una fracción de ella (cruza de dos puntos). Los individuos resultantes que sean no dominados ingresan a la memoria.
11. El tamaño de la población debe regresarse a su valor original, de manera que se eliminan los individuos excedentes, dando preferencia de sobrevivir a los nuevos clones.

12. Se repite el proceso desde el paso 3 un número predeterminado de veces.

La selección de los parámetros de los pasos 6 y 9 del algoritmo fueron determinados de manera experimental.

El proceso de clonación permite realizar explotación, mientras que la mutación favorece la exploración del espacio de búsqueda.

Como puede verse, este algoritmo usa la idea general del principio de selección clonal del sistema inmune, pero además toma “prestados” algunos de los operadores de los algoritmos evolutivos. Por ejemplo aplica el operador de cruza de dos puntos a la manera de los algoritmos genéticos con la finalidad de alcanzar puntos intermedios entre individuos de la memoria secundaria (paso 10 del algoritmo).

Asimismo, como en los algoritmos evolutivos para optimización multiobjetivo de segunda generación, usa un archivo externo como mecanismo elitista. De manera que, podríamos considerar esta propuesta como un híbrido que toma ideas del sistema inmune y de la computación evolutiva.

A Memoria secundaria

Usamos una memoria secundaria o externa como mecanismo de elitismo con la finalidad de mantener las mejores soluciones a lo largo del proceso [6] Los individuos almacenados en esta memoria son no dominados entre ellos, y además con respecto a individuos previos que intentaron ingresar a la memoria. De manera que, esta memoria secundaria almacena el conjunto solución producido por nuestro algoritmo.

Como la memoria secundaria no puede crecer de manera indefinida, puede llegar un momento en que se sature, y existan individuos que intentan ingresar a ella. En tal situación es necesario determinar si se les permite entrar o no, de manera que únicamente se permitirá el ingreso a aquellos individuos que de alguna manera mejoren la información contenida en la memoria. Dado que todos los elementos son no dominados, entonces se debe considerar otro criterio de ingreso, que en este caso es mejorar la distribución de las soluciones a lo largo del frente de Pareto. Para ello usamos la rejilla adaptativa propuesta por Knowles y Corne [7], la cual divide el espacio de la memoria secundaria en regiones de densidad.

V Experimentos

Nuestro algoritmo multiobjetivo basado en un sistema inmune artificial (SIAM) fue validado usando un conjunto de funciones de prueba que son comúnmente adoptadas por los investigadores del área de optimización multiobjetivo; en este trabajo se reportan

los resultados obtenidos para tres de ellas debido a restricciones de espacio.

El algoritmo (SIAM) se comparó contra dos algoritmos representativos del área que son: NSGA-II [8] y PAES [7]. Se realizaron 20 corridas de cada función para cada uno de los algoritmos. La comparación se efectuó utilizando tres métricas: dispersión, tasa de error y distancia generacional que serán descritas más adelante. Además se realizó una inspección visual de las soluciones graficando el verdadero frente de Pareto (obtenido por enumeración) contra la solución obtenida por los diferentes algoritmos (la que se encuentra en la media de acuerdo a la métrica de distancia generacional):

Dispersión (D): Esta métrica fue propuesta por Schott [9], y mide la varianza de la distancia de cada miembro del conjunto de óptimos de Pareto con respecto a su vecino más cercano:

$$(5) \quad D = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2},$$

donde $d_i = \min_j (|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$, $i, j = 1, \dots, n$, \bar{d} es el promedio de todos d_i , y n es el número de elementos del conjunto de Pareto obtenidos hasta el momento. Si $D = 0$ significa que nuestro algoritmo ha encontrado la distribución ideal de vectores no dominados.

Tasa de error (TE): Propuesta por Van Veldhuizen [10] para indicar el porcentaje de soluciones del frente Pareto actual (FP_{actual}) que no son miembros del verdadero frente de Pareto ($FP_{verdadero}$):

$$(6) \quad TE = \frac{\sum_{i=1}^n e_i}{n}$$

donde n es el número de vectores en FP_{actual} , $e_i = 0$ si el vector i es un miembro de $FP_{verdadero}$ y $e_i = 1$ de lo contrario. Un valor de $TE = 0$ indica el comportamiento ideal del algoritmo.

Distancia Generacional (DG): El concepto de distancia generacional fue introducido por Van Veldhuizen y Lamont [10] como una manera de estimar qué tan lejos están los elementos FP_{actual} de $FP_{verdadero}$ y se define como:

$$(7) \quad DG = \frac{(\sum_{i=1}^n d_i^p)^{1/p}}{n}$$

donde n es el número de vectores no dominados encontrados por el algoritmo que está siendo analizado, y d_i es la distancia Euclidiana entre cada uno de ellos y el miembro más cercano del verdadero frente de Pareto. Si $GD = 0$ indica que todos los elementos generados están en el verdadero frente de Pareto de la función.

Ejemplo 1

El primer ejemplo es una función de prueba propuesta por Kita [11]:

$$\text{Maximizar } F = (f_1(x, y), f_2(x, y))$$

donde:

$$f_1(x, y) = -x^2 + y,$$

$$f_2(x, y) = \frac{1}{2}x + y + 1$$

$$x, y \geq 0, 0 \geq \frac{1}{6}x + y - \frac{13}{2}, 0 \geq \frac{1}{2}x + y - \frac{15}{2}, 0 \geq 5x + y - 30.$$

Las figuras 2, 3 y 4 muestran el verdadero frente de Pareto graficado contra la solución encontrada por SIAM, NSGA-II y PAES, respectivamente.

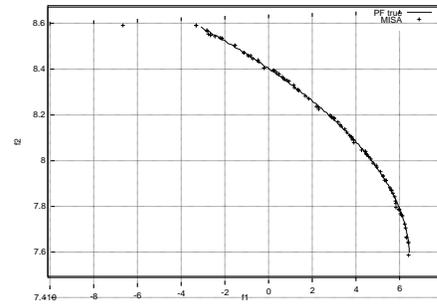


Figura 2: Frente de Pareto obtenidos por SIAM contra el verdadero frente de Pareto para el ejemplo 1

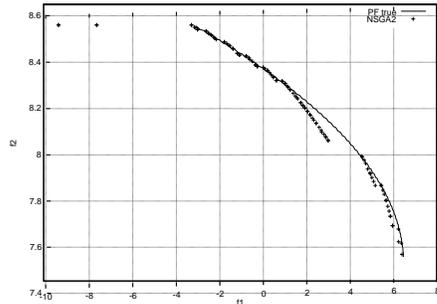


Figura 3: Frente de Pareto obtenidos por NSGA-II contra el verdadero frente de Pareto para el ejemplo 1

Los resultados obtenidos por las métricas se muestran en las tablas I, II y III para el ejemplo 1.

Podemos ver en las gráficas que la aproximación al verdadero frente de Pareto de SIAM es mejor que NSGA-II y PAES, pues estos últimos pierden una buena fracción del mismo. NSGA-II muestra un mejor desempeño claramente según la métrica de distribución. Para la distancia generacional, el mejor valor lo obtuvo SIAM seguido por NSGA-II y finalmente PAES. Para la

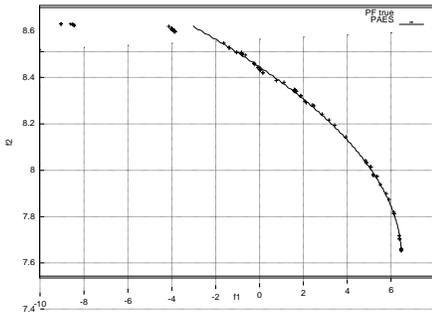


Figura 4: Frente de Pareto obtenidos por PAES contra el verdadero frente de Pareto para el ejemplo 1

Tabla I: Distribución para el ejemplo 1

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.27986 | 0.02201 | 0.12157 |
| Mejor | 0.04227 | 0.00724 | 0.04166 |
| Peor | 1.49319 | 0.03504 | 0.22073 |
| Dsv.Est. | 0.40732 | 0.00742 | 0.04998 |

de tasa de error los SIAM es un claro ganador, seguido por NSGA-II y finalmente, el peor desempeño para PAES.

Ejemplo 2

El segundo problema es una función de 2 objetivos, donde el frente de Pareto está fragmentado en 2 segmentos. Esta función fue propuesta por Schaffer [12]:

$$(8) \text{ Minimizar } f_1(x) = \begin{cases} -x & \text{if } x \leq 1 \\ -2 + x & \text{if } 1 < x \leq 3 \\ 4 - x & \text{if } 3 < x \leq 4 \\ -4 + x & \text{if } x > 4 \end{cases}$$

$$(9) \text{ Minimizar } f_2(x) = (x - 5)^2$$

y $-5 \leq x \leq 10$.

Las figuras 5, 6 y 7 muestran el verdadero frente de Pareto graficado contra la solución encontrada por SIAM, NSGA-II y PAES, respectivamente.

Los resultados obtenidos por las métricas de distribución y distancia generacional se muestran en las tablas IV y V respectivamente, y la tasa de error en

Tabla II: Distancia generacional para el ejemplo 1

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.03671 | 0.04515 | 0.09532 |
| Mejor | 0.00274 | 0.00395 | 0.00215 |
| Peor | 0.16035 | 0.47602 | 0.22446 |
| Dsv.Est. | 0.04362 | 0.11263 | 0.10471 |

Tabla III: Tasa de error para el ejemplo 1

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.58427 | 0.85409 | 0.65450 |
| Mejor | 0.45000 | 0.69091 | 0.50000 |
| Peor | 0.69000 | 1.00000 | 0.80000 |
| Dsv.Est. | 0.06997 | 0.10091 | 0.07870 |

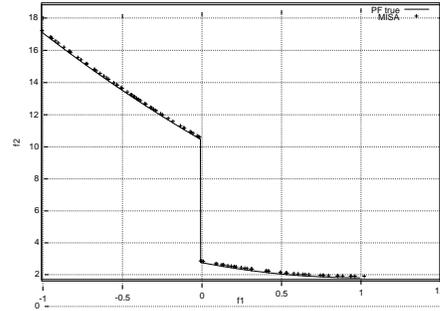


Figura 5: Frente de Pareto obtenidos por SIAM contra el verdadero frente de Pareto para el ejemplo 2

la tabla VI. El mejor valor obtenido para la métrica de distribución lo obtiene NSGA-II seguido por SIAM con valores muy cercanos y en último lugar PAES. Para el caso de la distancia generacional, la mejor aproximación la tiene NSGA-II seguida por SIAM con valores muy cercanos al ideal y finalmente PAES, por otro lado la métrica de tasa de error, muestra que SIAM tiene el mejor valor, ocupando el segundo lugar NSGA-II y en último lugar PAES.

Ejemplo 3

El tercer ejemplo es una función con 2 objetivos cuyo frente de Pareto es desconectado formando 4 segmentos.

Minimizar: $F = (f_1(x, y), f_2(x, y))$, donde

$$f_1(x, y) = x,$$

$$f_2(x, y) = (1 + 10y) * [1 - (\frac{x}{1+10y})^\alpha - \frac{x}{1+10y} \sin(2\pi qx)]$$

$$\text{y } 0 \leq x, y \leq 1, q = 4, \alpha = 2.$$

Las figuras 8, 9 y 10 muestran el verdadero frente de Pareto graficado contra la solución encontrada por SIAM, NSGA-II y PAES, respectivamente.

Los resultados obtenidos por las métricas se muestran en las tablas VII, VIII y IX para la función 3. De acuerdo con la métrica de distribución, el mejor va-

Tabla IV: Distribución para el ejemplo 2

| | SIAM | NSGA-II | PAES |
|---------|---------|---------|---------|
| Media | 0.06452 | 0.04422 | 0.24369 |
| Mejor | 0.04977 | 0.03906 | 0.01003 |
| Peor | 0.07638 | 0.05035 | 1.59981 |
| Dsv.Est | 0.00771 | 0.00292 | 0.34644 |

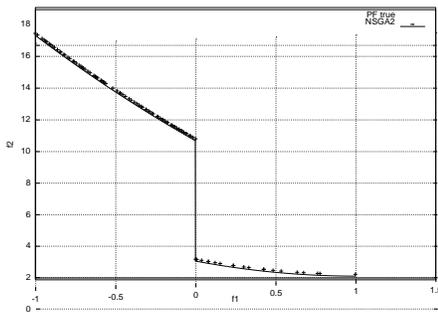


Figura 6: Frente de Pareto obtenidos por NSGA-II contra el verdadero frente de Pareto para el ejemplo 2

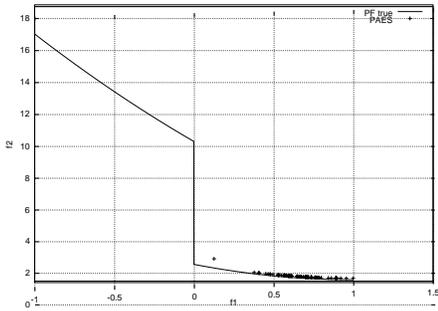


Figura 7: Frente de Pareto obtenidos por PAES contra el verdadero frente de Pareto para el ejemplo 2

El promedio lo obtuvo NSGA-II, y el segundo lugar lo ocupa SIAM, cuyos valores son muy aproximados. El tercer lugar lo ocupa PAES.

El mejor valor para la métrica de distancia generacional lo obtiene SIAM, seguido por NSGA-II ocupando el último lugar PAES. Para la tasa de error el mejor desempeño lo tienen PAES y NSGA-II, quedando SIAM en tercer lugar. Gráficamente, SIAM y NSGA-II muestran una mejor aproximación del verdadero frente de Pareto.

VI Conclusiones y perspectivas

Hemos presentado un algoritmo híbrido de un sistema inmune artificial con un algoritmo genético para problemas de optimización multiobjetivo con y sin restricciones.

Este algoritmo ha sido comparado contra dos técnicas que son representativas del estado del arte en optimización evolutiva multiobjetivo valiéndose pa-

Tabla V: Distancia Generacional para el ejemplo 2

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.00037 | 0.00029 | 0.00238 |
| Mejor | 0.00020 | 0.00025 | 0.00005 |
| Peor | 0.00171 | 0.00034 | 0.03494 |
| Dsv.Est. | 0.00039 | 0.00002 | 0.00778 |

Tabla VI: Tasa de error para el ejemplo 2

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.00700 | 0.01050 | 0.01550 |
| Mejor | 0.00000 | 0.00000 | 0.00000 |
| Peor | 0.02000 | 0.04000 | 0.10000 |
| Dsv.Est. | 0.00571 | 0.01099 | 0.02645 |

Tabla VII: Distribución para el ejemplo 3

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.00830 | 0.00760 | 0.03229 |
| Mejor | 0.00712 | 0.00657 | 0.01246 |
| Peor | 0.00957 | 0.00920 | 0.25179 |
| Dsv.Est. | 0.00069 | 0.00065 | 0.05240 |

Tabla VIII: Distancia generacional para el ejemplo 3

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.00024 | 0.00044 | 0.00304 |
| Mejor | 0.00020 | 0.00021 | 0.00012 |
| Peor | 0.00032 | 0.00443 | 0.02520 |
| Dsv.Est. | 0.00003 | 0.00094 | 0.00758 |

Tabla IX: Tasa de error para el ejemplo 3

| | SIAM | NSGA-II | PAES |
|----------|---------|---------|---------|
| Media | 0.32500 | 0.14250 | 0.13300 |
| Mejor | 0.16000 | 0.08000 | 0.06000 |
| Peor | 0.49000 | 0.29000 | 0.26000 |
| Dsv.Est. | 0.08544 | 0.05035 | 0.05121 |

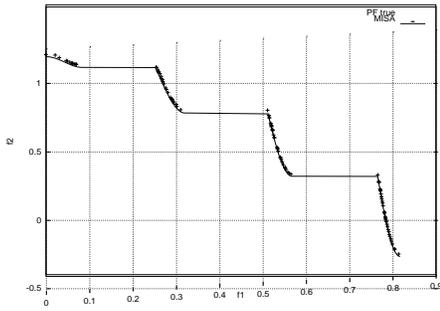


Figura 8: Frente de Pareto obtenidos por SIAM contra el verdadero frente de Pareto para el ejemplo 3

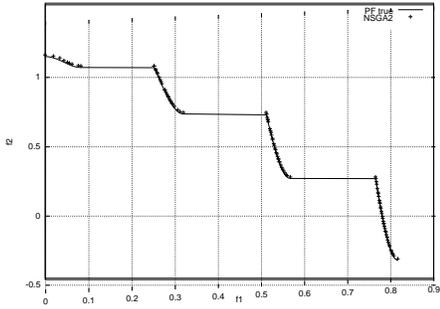


Figura 9: Frente de Pareto obtenidos por NSGA-II contra el verdadero frente de Pareto para el ejemplo 3

ra ello de algunas métricas que evalúan diferentes aspectos del desempeño de los algoritmos en cuestión.

Esta propuesta es el primer intento por usar un sistema inmune artificial para solucionar problemas de este tipo, siendo adicionalmente un algoritmo conceptualmente sencillo.

Podemos concluir que esta emulación del sistema inmune es una alternativa viable y competitiva para solucionar problemas de optimización multiobjetivo con y sin restricciones.

Como trabajo futuro, es necesario realizar ajustes al algoritmo con la finalidad de mejorar la distribución de las soluciones a lo largo del frente de Pareto, así como realizar una caracterización del tipo de problemas en los que puede obtener mejor desempeño.

Agradecimientos

La primera autora agradece a CONACyT por la beca otorgada para realizar estudios de posgrado en la Sección de Computación del Departamento de Ingeniería Eléctrica del CINVESTAV-IPN. El segundo autor agradece a CONACyT por el apoyo recibido a través del proyecto 34201-A.

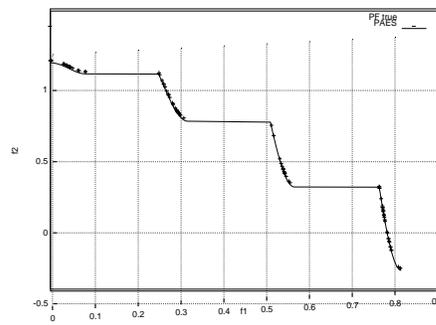


Figura 10: Frente de Pareto obtenidos por PAES contra el verdadero frente de Pareto para el ejemplo 3

Referencias

- [1] Coello Coello, Carlos A. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, August 1999.
- [2] Leandro Nunes de Castro y Jonathan Timmis, *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm* Springer-Verlag, 2002
- [3] Pareto, Vilfredo. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.
- [4] Burnet, F. M. *Clonal Selection and After*, In G. I. Bell and A. S. Perelson and G. H. Pimgley Jr., editors, *Theoretical Immunology*, pp.63–85, 1978.
- [5] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.
- [6] Coello Coello Carlos A., Van Veldhuizen David A. y Lamont Gary *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York ISBN 0-3064-6762-3 2002.
- [7] Knowles, Joshua D. and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [8] Deb, Kalyanmoy, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

- [9] Schott, Jason R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.
- [10] Van Veldhuizen, David A. and Gary B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.
- [11] Kita, Hajime, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 504–512, Berlin, Germany, September 1996. Springer-Verlag.
- [12] Schaffer, J. David, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, Vanderbilt University, 1984