

# OPTIMIZACIÓN DE FUNCIONES MONO-OBJETIVO CON Y SIN RESTRICCIONES, Y FUNCIONES MULTI-OBJETIVO A TRAVÉS DE HEURÍSTICAS BIO-INSPIRADAS

Victoria Aragón<sup>†</sup>, Leticia Cagnina<sup>†</sup>, Susana Esquivel<sup>†</sup>

<sup>†</sup>Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)  
Facultad de Ciencias Físicas, Matemáticas y Naturales - Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106 - (D5700HHW) - San Luis - Argentina  
Tel: (02652) 420823 / Fax: (02652) 430224 - e-mail: {vsaragon, lcagnina, esquivel}@unsl.edu.ar

## Resumen

En esta presentación se describen, en forma breve, algunas de las direcciones de investigación que en la actualidad se están desarrollando dentro de la línea "Optimización Mono y Multiobjetivo" del Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC). Uno de los objetivos de esta línea, es el estudio y desarrollo de metaheurísticas aptas para resolver problemas de optimización. En particular, el énfasis está puesto en las heurísticas de inteligencia computacional basadas en los paradigmas de inteligencia colectiva y biológicos, como ser *Particle Swarm Optimization* y Sistemas Inmunes Artificiales.

## 1. INTRODUCCIÓN

En la actualidad la línea de investigación focaliza su trabajo sobre algunas heurísticas bio-inspiradas, en particular: Sistemas Inmunes Artificiales (SIA) y Particle Swarm Optimizers (PSO), atacando los siguientes problemas: optimización numérica global (SIA), optimización de funciones con restricciones (SIA, PSO), optimización de funciones en ambientes no estacionarios (SIA) y optimización multiobjetivo (PSO).

La optimización de problemas ha sido y es, un área activa de investigación. A medida que los problemas de mundo real, ya sea en el campo de la ciencia o la ingeniería, se vuelven más complejos, es necesario contar con heurísticas capaces de resolver rápida y eficientemente estos problemas. En las últimas décadas algoritmos bio-inspirados han llamado la atención de los investigadores. Uno de ellos es la heurística *Particle Swarm Optimization*, la cual está basada en el comportamiento que muestran ciertas especies como aves, las cuales utilizan la inteligencia colectiva para sacar provecho de actividades realizadas previamente

por algún miembro de la comunidad. En la heurística todos los individuos de la población siguen la trayectoria del mejor integrante del grupo, y también la memoria de su mejor comportamiento previo. De forma análoga sucede con la heurística *Sistema Inmune Natural* y su poderosa capacidad de procesamiento de información [19]. El Sistema Inmune (SI) es un sistema muy complejo con varios mecanismos de defensa contra organismos patógenos. El principal propósito del SI es reconocer a todas las células dentro del cuerpo y categorizarlas con el objeto de inducir un mecanismo de defensa apropiado. EL SI aprende a través de la evolución para poder distinguir entre los antígenos externos peligrosos de las células propias del cuerpo. Un buen conocimiento de cómo trabaja el sistema inmune es una excelente motivación para desarrollar un Sistema Inmune Artificial (SIA) para tratar problemas de optimización, ya sean global, con restricciones, incluyendo problemas de ingeniería del mundo real o funciones en ambientes no estacionarios. Cuando este modelo computacional es comparado con otros modelos bien establecidos, de la inteligencia computacional, como los algoritmos evolutivos, las colonias de hormigas, las redes neuronales, entre otras, puede decirse que los SIAs están aún en los primeros años de su infancia. Con respecto a la aplicación de SIAs en optimización de funciones con restricciones, poco es el trabajo reportado (ver [8]).

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

### 2.1. Particle Swarm Optimization

En [2] se introdujo el algoritmo *CPSO-shake*, una versión mejorada de *Constrained Particle Swarm Optimization* para optimización de funciones numéricas con restricciones. El algoritmo utiliza un método simple pa-

ra manejar restricciones de diversos tipos: lineales, no-lineales, de igualdad y de desigualdad. Además esta versión incorpora un mecanismo de escape de óptimos locales (*shake*) a través del movimiento de las partículas a nuevas zonas del espacio de búsqueda. También se utiliza una población dual, en vez de una única población que evoluciona en el proceso de búsqueda, para evitar la convergencia prematura. Los detalles de implementación pueden consultarse en [6].

La performance del algoritmo *CPSO-shake* permitió la utilización del mismo en la resolución de problemas más complejos como son los de ingeniería: problema del diseño óptimo de armaduras [17] (*trusses* en inglés), y de diseño de piezas: *welded beam* [20], *pressure vessel* [23], *speed reducer* [14] y, *tension/compression spring* [4]. Estos dos tipos de problemas de ingeniería son difíciles de resolver, como todo problema del mundo real. Por ello, y para finalizar con la parte de optimización de funciones mono-objetivo, se aplicó el mismo algoritmo en la resolución del problema de despacho de cargas eléctricas. Este problema presenta costosas restricciones a satisfacer, además de la restricción de tiempo que es casi fundamental en la solución que debe arrojar el algoritmo. Esto significa que se necesita un algoritmo que sea capaz de resolver bien el problema, pero además de ello, en forma muy rápida. Al algoritmo *CPSO-shake* se le realizó una pequeña modificación: se emplea la población completa en vez de las dos poblaciones que evolucionan en paralelo. Esto es necesario ya que el algoritmo debe arrojar una respuesta en un tiempo mínimo y utilizando la menor cantidad posible de iteraciones. Si se utilizara la población dividida, ambas subpoblaciones necesitarían mayor cantidad de ciclos para obtener una respuesta óptima.

Habiendo probado la heurística PSO con un buen número de problemas mono-objetivo, y habiendo concluido en la buena performance de la misma, se decidió realizar la adaptación de la misma para el caso multiobjetivo. Si bien los problemas multiobjetivos son muy difíciles de resolver, actualmente se está trabajando en algunas aproximaciones que permitirán resolver este tipo de problemas de manera eficiente.

La próxima sección muestra los resultados obtenidos para cada caso experimental.

## 2.2. Sistema Inmune Artificial

El propósito de esta investigación es explorar las capacidades de búsqueda de un nuevo modelo propuesto de SIA, llamado TCELL, en el contexto de problemas de optimización global, problemas de optimización con restricciones, incluyendo funciones de ingeniería, y optimización de funciones en ambientes no estacionarios. TCELL está basado en las características y procesos que sufren las células T del sistema inmune de los vertebrados.

TCELL es el primer modelo computacional que emplea la filogenia de las células T. Este modelo está com-

puesto por una fase reconocedora y una fase efectora. La primera involucra a las células T vígenes mientras que en la segunda están involucradas las células efectoras y de memoria. La fase de reconocimiento debe proveer dos poblaciones de células efectoras con la diversidad suficiente para que en la próxima fase se pueda encontrar una célula que optimice el problema. Mientras que, la fase efectora es efectivamente la encargada de encontrar, a través de diferentes etapas, dicha célula. TCELL toma en consideración muchos de los procesos que afectan a las células T desde su origen hasta que se transforman en células de memoria, como ser la selección positiva y negativa de las células T vígenes y los procesos de proliferación y distintos tipos de diferenciación para las células efectoras y de memoria. Para realizar estos dos últimos procesos TCELL utiliza señales co-estimulantes, llamadas citocinas.

La premisa de TCELL es que las células T sólo reaccionan ante la presencia de un antígeno junto con señales co-estimulantes, a través del desencadenamiento de una serie de acciones, estas acciones son influenciadas por una serie de señales emitidas por las mismas células T (citocinas). Es decir, las señales determinan el nivel con el cual se desencadenan las acciones: proliferación y diferenciación de las células T.

El modelo trabaja sobre cuatro poblaciones: células vírgenes (CV), células efectoras con denominador de grupo 4 (CD4), células efectoras con denominador de grupo 8 (CD8), este denominador determina las propiedades de las células, y células de memoria (CM). Cada población está compuesta por un conjunto de células T, cuya característica está sujeta a la población que pertenece. Los detalles de implementación se citan en [3].

El pseudo-código de TCELL para problemas de optimización global se presenta en el Algoritmo 1.

## 3. RESULTADOS OBTENIDOS

### 3.1. Particle Swarm Optimization

*CPSO-shake* fue evaluado con el *benchmark* extendido presentado en [15], el cual cuenta con 20 funciones de prueba con diferentes características. Los resultados obtenidos fueron contrastados con el algoritmo *Stochastic Ranking* (SR) [22] el cual sigue representando uno de los mejores en optimización de funciones con restricciones, así como también con una aproximación de una estrategia evolutiva (AESSR) [21] que utiliza también *Stochastic Ranking*.

Los resultados alcanzados con la presente versión de PSO, demostraron ser muy competitivos con respecto a SR y a AESSR. Las figuras 1 y 2 muestran el error promedio de cada algoritmo con respecto al *benchmark*, para cada una de las 20 funciones. Aunque la variabilidad de resultados obtenidos por SR y AESSR fue levemente inferior a la de nuestro algoritmo, argumentamos que esto está relacionado al mecanismo de escape de óptimos (*shake*) implementado para mantener

**Algoritmo 1** Pseudo-código de TCELL para problemas de optimización global

```

1: Inicializar_CV();
2: Evaluar_CV();
3: Asignar_Proliferacion();
4: Activar_CDs_Fase_II();
5: Dividir_CDs();
6: Seleccion_Positiva();
7: Seleccion_Negativa_CD4();
8: Seleccion_Negativa_CD8();
9: while un nro determinado de evaluaciones do
10:   while un numero determinado de veces do
11:     Activar_CD4();
12:   end while
13:   Ordenar_CD4();
14:   Comunicacion_CD4_CD8();
15:   while un nro determinado de veces do
16:     Activar_CD8();
17:   end while
18:   Ordenar_CD8();
19:   Insertar_CDs_en_CM();
20:   while un nro determinado de veces do
21:     Activar_CM();
22:   end while
23:   Ordenar_CM();
24:   Estadisticas();
25: end while

```

diversidad en el algoritmo. En [6] se concluyó que *CPSO-shake* es una de las aproximaciones PSO más competitivas en cuanto a técnicas de manejo de restricciones.

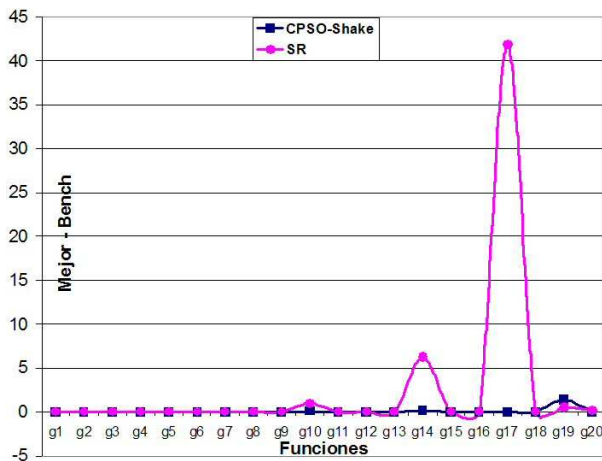


Figura 1: Performance de *CPSO-shake* versus SR

Para las funciones de armaduras se obtuvieron los resultados que se muestran en la tabla 1. Notar que los mejores conocidos hasta ese momento eran los obtenidos en [17]. Nuestro algoritmo PSO [5] obtuvo una leve mejora del óptimo para la función 10-bar, una mejora significativa para 25-bar y una mejora considerable en

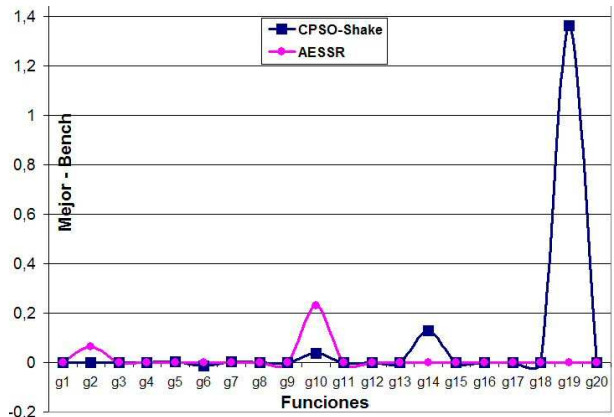


Figura 2: Performance de *CPSO-shake* versus AE

200-bar. Esto demuestra el buen poder exploratorio que ejerce la heurística que concluyó con la mejora de los óptimos conocidos.

Con respecto a los resultados obtenidos para las funciones de diseño de piezas [7], cabe destacar que se compararon los resultados de *CPSO-shake* con los de una aproximación PSO muy elaborada que incluye complejos operadores para evitar convergencia prematura, denominada *COPSO* [1]. También se utilizó en esta sintética comparación, un algoritmo evolutivo (Mezura) [18] que utiliza complejos operadores para manejar las restricciones de los problemas. Los resultados son mostrados en la tabla 2. Se puede observar que los tres algoritmos probados obtuvieron los óptimos en E01, *COPSO* y nuestro algoritmo alcanzaron los mejores conocidos en E02, E03 y E04, mientras que los valores encontrados por Mezura presentan una leve desviación con respecto al óptimo. Esto concluye que *CPSO-shake* presenta un buen funcionamiento y comparable con aproximaciones que utilizan operadores más complejos.

El problema de despacho de cargas puede ser consultado en [24], en donde se presenta una algoritmo PSO que resuelve 3 instancias del problema: Caso A: despacho para 3 generadores con polución, Caso B: despacho para 3 generadores son polución y Caso C: despacho de cargas para 40 generadores sin polución. *CPSO-shake* alcanza los óptimos conocidos para cada instancia, al igual que el algoritmo de [24], pero el aporte de nuestro algoritmo tiene que ver con el tiempo de ejecución insumido para obtener dichos valores. La figura 3 muestran los tiempos de cada algoritmo. Como puede notarse, *CPSO-shake* mejora notablemente el tiempo de respuesta del PSO de referencia, convirtiéndolo en un algoritmo más adecuado para ser utilizado en este tipo de problemas.

### 3.2. Sistema Inmune Artificial

TCELL fue validado con un conjunto de 23 funciones de prueba tomado de la literatura especializada [26] y comparados con diferentes enfoques bio-inspirados

Tabla 1: Comparación de los mejores valores obtenidos con CPSO-shake, con respecto a los mejores conocidos.

Truss	Mejores Conocidos			CPSO-shake		
	Opt	Std.Dev	Worst	Opt	Std.Dev	Worst
10-bar	4,656.39	0.18	4,656.71	<b>4,656.36</b>	2.84	4,696.06
25-bar	485.33	-	534.84	<b>467.30</b>	0.35	470.87
200-bar	20,319.58	2,492.24	30,269.49	<b>16,468.82</b>	1,476.00	20,373.6719

Tabla 2: Mejores valores alcanzados con COPSO, Mezura and CPSO-shake.

Funciones	Mejores Conocidos	CPSO-skahe	COPSO	Mezura
E01	1.724852	1.724852	1.724852	1.724852
E02	6,059.714335	6,059.714335	6,059.714335	6,059.7143
E03	-	2,996.348165	2,996.372448	2,996.348094
E04	0.012665	0.012665	0.012665	0.012689

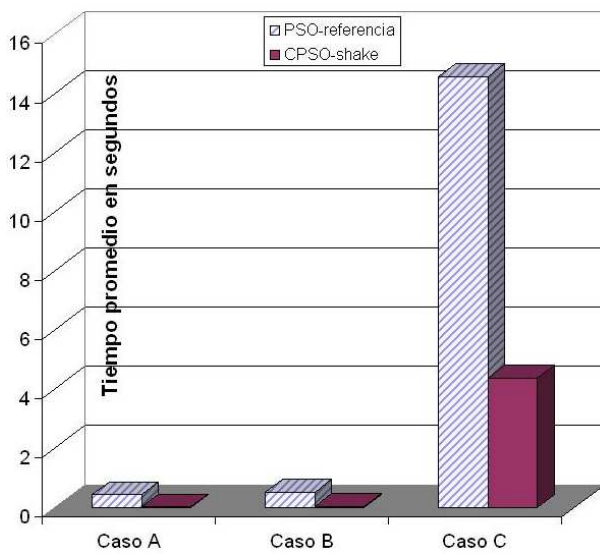


Figura 3: Tiempo insumido por *CPSO-shake* versus PSO [24] para el despacho de cargas.

[3]. Los enfoques considerados fueron: un algoritmo de Evolución Diferencial (DE) [16], un Particle Swarm Optimizer (PSO) [16], un Algoritmo Evolutivo (SEA) [16], un Algoritmo inmune para problemas de optimización global continuos (OPT-IA) [25] y una versión mejorada de este último (OPT-IMMALG) [9]. Las figuras 4, 5 y 6 muestran los valores medios obtenidos por cada enfoque para cada función.

En la actualidad se están realizando los experimentos correspondientes para los problemas de optimización restringidos y dinámicos. Versiones preliminares del modelo propuesto fueron validadas para el primer tipo de problema en [11, 10, 13, 12].

#### 4. Reconocimientos

El LIDIC reconoce el constante soporte brindado por la Universidad Nacional de San Luis y la ANPYCIT que financian sus actuales investigaciones.

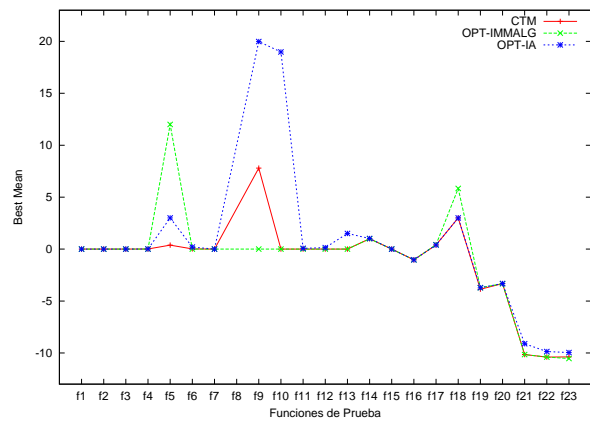


Figura 4: Resultados - Valores Medios Obtenidos por los SIA's

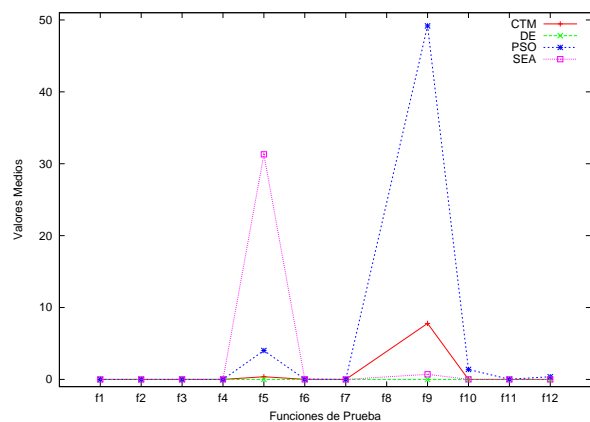


Figura 5: Resultados - Valores Medios Obtenidos por CTM, DE, PSO y SEA



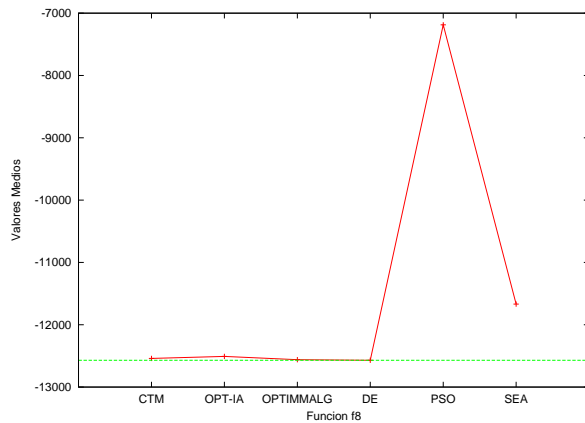


Figura 6: Resultados - Valores Medios Obtenidos por todos los enfoques para  $f_8$

## Referencias

- [1] E. Villa Diharce A. Hernandez Aguirre, A. Muñoz Zavala and S. Botello Rionda. Copso: Constrained optimization via pso algorithm. *Center for Research in Mathematics (CIMAT)*.
- [2] Victoria Aragón, Leticia Cagnina, Claudia Gatica, and Susana Esquivel. Metaheurísticas basadas en inteligencia computacional aplicadas a la resolución de problemas de optimización numérica con y sin restricciones y optimización combinatoria. In *IX Workshop de Investigadores en Ciencias de la Computación (WICC 2007)*, pages 124–129, Trelew, Chubut, Argentina, 2007.
- [3] Victoria Aragon, Susana Esquivel, and Carlos Coello Coello. Artificial immune system for solving global optimization problems. In *Congreso Argentino en Ciencias de la Computacion (CACIC08)*, 2008.
- [4] A. Belegundu. A study of mathematical programming methods for structural optimization. *PhD thesis, Department of Civil Environmental Engineering, University of Iowa, Iowa*, 1(1).
- [5] Leticia C. Cagnina, Susana C. Esquivel, and Carlos A. Coello Coello. Solving constrained optimization problems with a particle swarm optimization algorithm. *Artículo enviado (actualmente en proceso de revisión) a: Computer Methods in Applied Mechanics and Engineering (CMAME)*, X(X).
- [6] Leticia C. Cagnina, Susana C. Esquivel, and Carlos A. Coello Coello. A bi-population pso with a shake-mechanism for solving constrained numerical optimization. In *Congress on Evolutionary Computation - CEC 2007*, pages 670–676, Singapur, 2007.
- [7] Leticia C. Cagnina, Susana C. Esquivel, and Carlos A. Coello Coello. Solving engineering optimization problems with the simple constrained pso. *Informatica, International Journal of Computing and Informatics - ISSN 0350-5596*, 32(3):319–326, 2008.
- [8] Nareli Cruz Cortés, Daniel Trejo-Pérez, and Carlos A. Coello Coello. Handling constrained in global optimization using artificial immune system. In Christian Jacob, Marcin L. Pilat, Peter J. Bentley, and Jonathan Timmis, editors, *Artificial Immune Systems. 4th International Conference, ICARIS 2005*, pages 234–247. Springer. Lecture Notes in Computer Science Vol. 3627, Banff, Canada, August 2005.
- [9] Vincenzo Cutello, Giuseppe Nicosia, and Mario Pavone. Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 950–954, New York, NY, USA, 2006. ACM.
- [10] S. Esquivel, C. Coello Coello, and V. Aragón. Artificial immune system for solving constrained optimization problems. *Revista Iberoamericana de Inteligencia Artificial (AEPIA)*, ISSN: 1137-3601. Vol. 11, Nro. 35, pp. 55-66, 2007.
- [11] S. Esquivel, C. Coello Coello, and V. Aragón. A novel model of artificial immune system for solving constrained optimization problems with dynamic tolerance factor. *VI Mexican International Conference on Artificial Intelligence (MICA107)*, Aguascalientes, México. LNAI 4827, pp. 19-29, year=2007.
- [12] S. Esquivel, C. Coello Coello, and V. Aragón. Optimizing constrained problems through a t-cell artificial immune system, 2008. *Journal of Computer Science & Technology (JCS&T)*, ISSN 1666-6038 Vol. 8, Nro. 3, pp. 158-165.
- [13] S. Esquivel, C. Coello Coello, and V. Aragón. Solving constrained optimization using a t-cell artificial immune system, 2008. *Revista Iberoamericana de Inteligencia Artificial (AEPIA)*, ISSN: 1137-3601. Vol. 12, Nro. 40, pp. 7-22.
- [14] J. Golinski. An adaptive optimization system applied to machine synthesis. *Mechanism and Machine Synthesis*, 8.
- [15] [http://www.ntu.edu.sg/home/EPNSugan/index\\_files/CEC-06/CEC06.html](http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-06/CEC06.html).
- [16] Versterstrom J. and Thomsen R. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems. *Congress on Evol. Comp., CEC04*, vol. 1, pp. 1980-1987, 2004.
- [17] Ricardo Landa Becerra and Carlos A. Coello Coello. Cultured differential evolution for constrained optimization. *Computer Methods in Applied Mechanics and Engineering*, 195(33–36):4303–4322, July 1 2006.
- [18] E. Mezura and C. Coello. Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *Proceedings of the 4th Mexican International Conference on Artificial Intelligence, MICA1 2005*, pages 652–662, 2005.
- [19] Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, New York, 2002.
- [20] K. Ragsdell and D. Phillips. Optimal design of a class of welded structures using geometric programming optimization. *ASME Journal of Engineering for Industries*, 98(3):1021–1025, 1976.
- [21] T. P. Runarsson. Approximate evolution strategy using stochastic ranking. In *2006 IEEE World Congress on Computational Intelligence*, volume 3, pages 2760–2767, British Columbia, Canada, 2006.
- [22] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. In *IEEE Transactions on Evolutionary Computation*, volume 3, pages 284–294, 2000.
- [23] E. Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *ASME Journal of Mechanical Design*, 112.
- [24] P. Sriyanyong, Y. H. Song, and P. J. Turner. Particle swarm optimisation for operational planning: Unit commitment and economic dispatch. Chapter in *Evolutionary Scheduling*. Keshav P. Dahal, Kay Chen Tan and Peter I. Cowling (Eds.). ISSN: 1860-949X. Studies in Computational Intelligence, Vol. 49, Springer. pp. 313-347, 2007.
- [25] Cutello V, Narzisi G, Nicosia G, and Pavone M. An immunological algorithm for global numerical optimization. *7th International Conference on Artificial Evolution, EA'05*, October 26-28 2005, University of Lille, France. Springer-Verlag, Lecture Notes in Computer Science, vol. 3871, pp. 284-295.
- [26] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3:82–102, 1999.