Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

UNIVERSIDAD DE GRANADA

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

REPSOL

# La estadística como aliada en la experimentación de algoritmos evolutivos


Daniel Molina Cabrera
dmolinac@ugr.es


DECSAI
Universidad de Granada

UNIVERSIDAD DE GRANADA

# ¿Quién soy yo?

# ¿Quién soy yo?

- Titular del Dpto de CCIA, Universidad de Granada.
- Miembro del Instituto Interuniversitario de Inteligencia Artificial (DaSci).

DaSCI
Instituto Andaluz Interuniversitario en
**Data Science** and **Computational Intelligence**

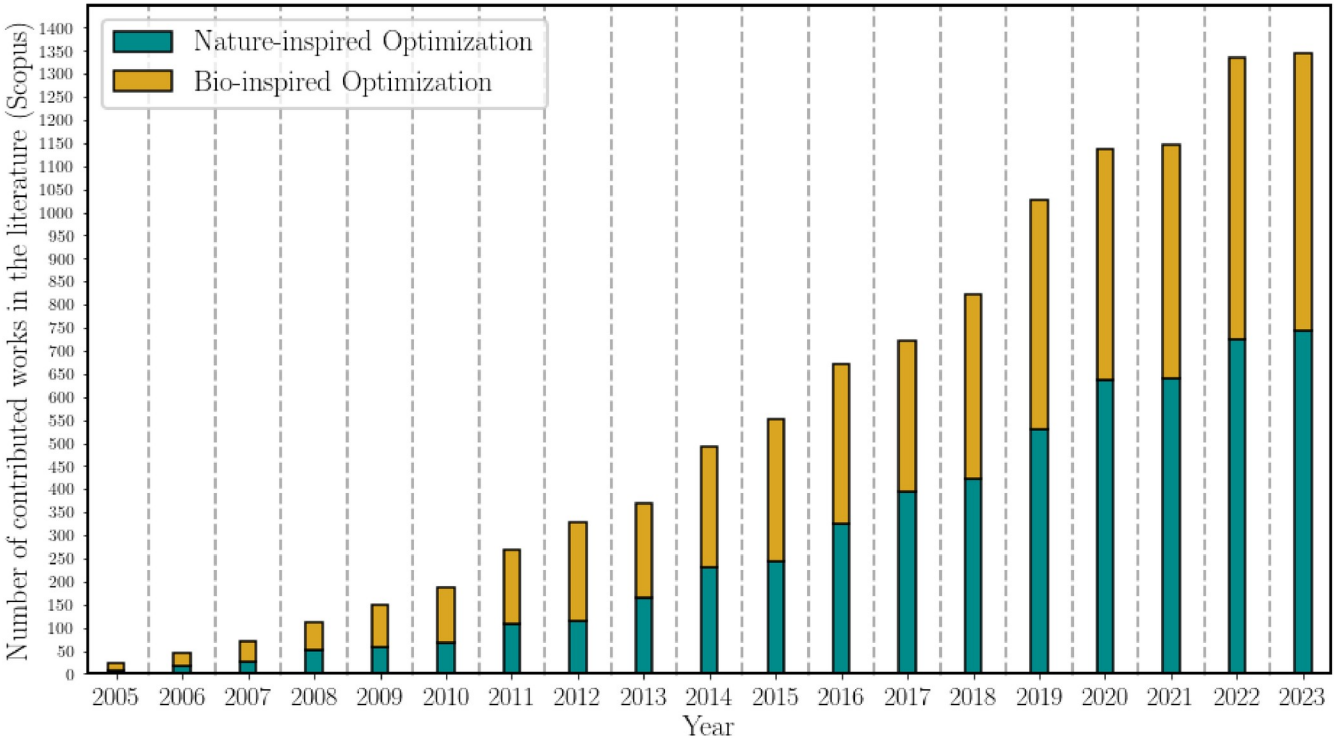
# Esperiencia

- Más de 10 años.
- Optimización en alta dimensionalidad.
- Problemas de optimización con empresas.
- *Deep Learning*, Neuroevolución.
- Dentro del 2% de científicos más influyentes según Stanford.

DaSCI
Instituto Andaluz Interuniversitario en
Data Science and Computational Intelligence

# Material

# Cada vez hay más propuestas




¿Eso es bueno, no?
¿Las mejores serán las recientes?

D. Molina, J. Poyatos, J. D. Ser, S. García, A. Hussain, y F. Herrera, «Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration versus Algorithmic Behavior, Critical Analysis and Recommendations (from 2020 to 2024)», 17 de abril de 2024, arXiv: arXiv:2002.08136. doi: 10.48550/arXiv.2002.08136. -

# Paradoja del éxito

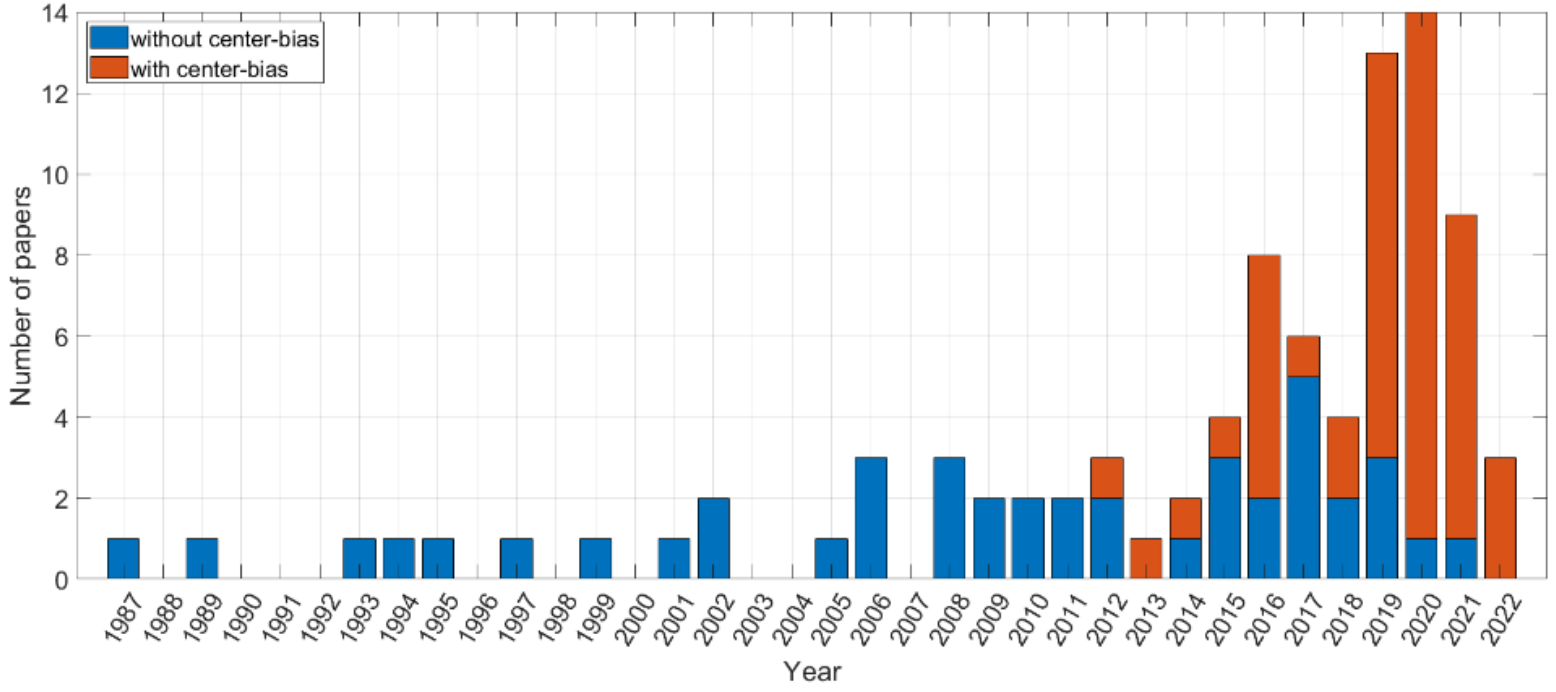
# No hay confianza en las comparaciones



¿Motivos?
Veamos algunos ejemplos

# Sobre-estimar algoritmo

*Jakub Kudela, The Evolutionary Computation Methods No One Should Use*




J. Kudela, «The Evolutionary Computation Methods No One Should Use», 5 de enero de 2023, arXiv: arXiv:2301.01984. doi: 10.48550/arXiv.2301.01984.

# Hay algoritmos muy conocidos

"butterfly" by davidyuweb is licensed under CC BY-NC 2.0.

"Origami Sailfish" by Ivan Svatko is licensed under CC BY-NC-ND 2.0

"Sky Bokeh with Dragonfly" by desertdutchman is licensed under CC BY 2.0.

IDEA: Comparar usando benchmarks que eviten estos sesgos

# Muchos más

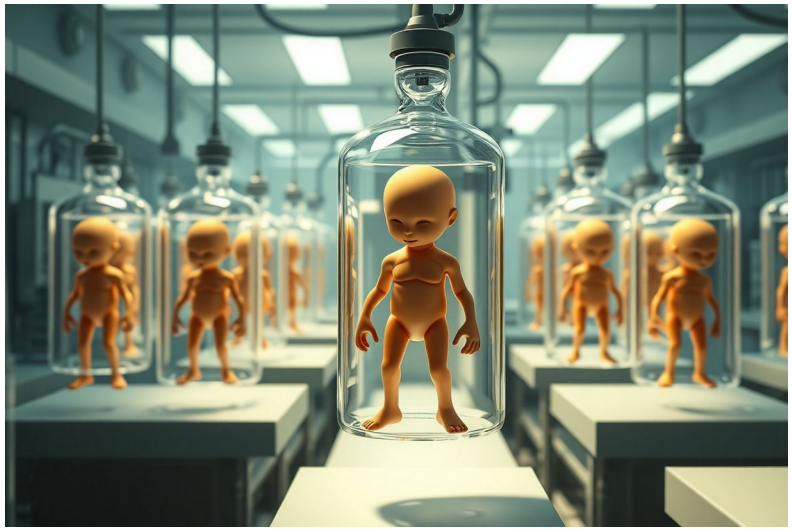
Table 3: Considered algorithms and results.

| Abbreviation | Method name | Year | Geomean | Abbreviation | Method name | Year | geomean |
|---|---|---|---|---|---|---|---|
| ABC [10] | Artificial Bee Colony | 2008 | 1.29E+00 | HC [15] | Hill Climbing | 1993 | 1.13E+00 |
| ACOR [16] | Ant Colony Optimization Continuous | 2008 | 7.40E-01 | HGS [17] | Hunger Games Search | 2021 | 3.69E+06 |
| AEO [18] | Artificial Ecosystem-based Optimization | 2020 | 1.01E+07 | HGSO [19] | Henry Gas Solubility Optimization | 2019 | 8.07E+03 |
| ALO [20] | Ant Lion Optimizer | 2015 | 1.44E+00 | HHO [21] | Harris Hawks Optimization | 2019 | 1.62E+05 |
| AO [22] | Aquila Optimization | 2021 | 2.26E+05 | HS [23] | Harmony Search | 2001 | 9.97E-01 |
| AOA [24] | Arithmetic Optimization Algorithm | 2021 | 1.01E+10 | IWO [25] | Invasive Weed Optimization | 2006 | 1.88E+00 |
| ArchOA [26] | Archimedes Optimization Algorithm | 2021 | 3.75E+07 | JA [27] | Jaya Algorithm | 2016 | 1.19E+01 |
| ASO [28] | Atom Search Optimization | 2019 | 8.71E-01 | KMA [29] | Komodo Mlipir Algorithm | 2022 | 1.84E+05 |
| BA [30] | Bat-inspired Algorithm | 2010 | 1.44E+00 | LCO [31] | Life Choice-based Optimization | 2020 | 8.31E+07 |
| BBO [32] | Biogeography-Based Optimization | 2008 | 6.43E-01 | MA [33] | Memetic Algorithm | 1989 | 1.68E-03 |
| BeesA [34] | Bees Algorithm | 2006 | 1.16E+00 | MFO [35] | Moth-Flame Optimization | 2015 | 1.73E-01 |
| BES [36] | Bald Eagle Search | 2020 | 2.62E+08 | MGO [37] | Mountain Gazelle Optimizer | 2022 | 1.28E+01 |
| BFO [38] | Bacterial Foraging Optimization | 2002 | 9.66E-01 | MPA [39] | Marine Predators Algorithm | 2020 | 1.06E+02 |
| BOA [40] | Butterfly Optimization Algorithm | 2019 | 9.57E+05 | MRFO [41] | Manta Ray Foraging Optimization | 2020 | 6.40E+07 |
| BRO [42] | Battle Royale Optimization | 2021 | 2.59E+09 | MSA [43] | Moth Search Algorithm | 2018 | 8.37E+00 |
| BSA [44] | Bird Swarm Algorithm | 2016 | 1.09E+01 | MVO [45] | Multi-Verse Optimizer | 2016 | 1.75E+00 |
| BSO [46] | Brain Storm Optimization | 2011 | 7.85E+00 | NMRA [47] | Naked Mole-Rat Algorithm | 2019 | 5.65E+08 |
| CA [48] | Culture Algorithm | 2009 | 7.18E-01 | NRO [49] | Nuclear Reaction Optimization | 2019 | 2.30E+06 |
| CEM [50] | Cross-Entropy Method | 2005 | 1.33E+00 | PFA [51] | Pathfinder Algorithm | 2019 | 3.11E+08 |
| CGO [52] | Chaos Game Optimization | 2021 | 2.14E+07 | PSO [53] | Particle Swarm Optimization | 1995 | 9.70E-01 |
| ChOA [54] | Chimp optimization algorithm | 2020 | 3.89E+03 | PSS [55] | Pareto-like Sequential Sampling | 2021 | 1.77E+03 |
| COA [56] | Coyote Optimization Algorithm | 2018 | 4.00E+06 | QSA [57] | Queuing Search Algorithm | 2021 | 7.91E-01 |
| CRO [58] | Coral Reefs Optimization | 2014 | 9.69E-01 | RKO [14] | Runge Kutta Optimizer | 2021 | 7.36E+04 |
| CSA [59] | Cuckoo Search Algorithm | 2009 | 1.10E+00 | SA [60] | Simulated Annealing | 1987 | 8.95E-01 |
| CSO [61] | Cat Swarm Optimization | 2006 | 9.58E-01 | SARO [62] | Search And Rescue Optimization | 2019 | 2.27E+00 |
| DE [11] | Differential Evolution | 1997 | 9.66E-01 | SBO [13] | Satin Bowerbird Optimizer | 2017 | 3.95E+00 |
| DandO [63] | Dandelion Optimizer | 2022 | 3.59E+02 | SCA [64] | Sine Cosine Algorithm | 2016 | 1.18E+04 |
| DO [65] | Dragonfly Optimization | 2016 | 6.62E+02 | SFO [66] | SailFish Optimizer | 2019 | 2.57E+07 |
| EFO [67] | Electromagnetic Field Optimization | 2016 | 6.78E-01 | SHO [68] | Spotted Hyena Optimizer | 2017 | 1.31E+00 |
| EHO [69] | Elephant Herding Optimization | 2015 | 3.99E+03 | SLO [70] | Sea Lion Optimization Algorithm | 2019 | 2.83E+00 |
| EO [71] | Equilibrium Optimizer | 2020 | 4.65E+03 | SMA [72] | Slime Mould Algorithm | 2020 | 4.54E+06 |
| EOA [73] | Earthworm Optimisation Algorithm | 2018 | 2.55E+05 | SRSR [74] | Swarm Robotics Search And Rescue | 2017 | 2.03E+00 |
| EP [75] | Evolutionary Programming | 1999 | 1.43E+00 | SSA [76] | Sparrow Search Algorithm | 2020 | 2.61E+06 |
| ES [77] | Evolution Strategies | 2002 | 1.14E+00 | SSDO [78] | Social Ski-Driver Optimization | 2020 | 5.40E+08 |
| FA [79] | Fireworks Algorithm | 2010 | 1.34E+00 | SSO [80] | Salp Swarm Optimization | 2017 | 2.28E+01 |
| FBIO [81] | Forensic-Based Investigation Optimization | 2020 | 5.07E+06 | SSpiderA [82] | Social Spider Algorithm | 2015 | 1.12E+00 |
| FFA [83] | Firefly Algorithm | 2011 | 1.18E+00 | STOA [84] | Sooty Tern Optimization Algorithm | 2019 | 6.78E+04 |
| FOA [85] | Fruit-fly Optimization Algorithm | 2012 | 4.01E+00 | TLO [86] | Teaching Learning-based Optimization | 2012 | 3.19E+04 |
| FPA [87] | Flower Pollination Algorithm | 2012 | 9.74E-01 | TPO [88] | Tree Physiology Optimization | 2019 | 2.20E+01 |
| GA [89] | Genetic Algorithm | 1994 | 1.02E+00 | TSA [90] | Tunicate Swarm Algorithm | 2020 | 6.25E+06 |
| GBO [91] | Gradient-Based Optimizer | 2020 | 7.17E+07 | TWO [92] | Tug of War Optimization | 2017 | 9.69E-01 |
| GCO [93] | Germinal Center Optimization | 2018 | 1.02E+00 | VCS [94] | Virus Colony Search | 2016 | 2.90E+04 |
| GOA [95] | Grasshopper Optimization Algorithm | 2017 | 3.39E+00 | WDO [96] | Wind Driven Optimization | 2013 | 4.86E+01 |
| GSKA [97] | Gaining Sharing Knowledge-based Algorithm | 2020 | 4.51E-01 | WHO [98] | Wildebeest Herd Optimization | 2019 | 8.63E+02 |
| GWO [99] | Grey Wolf Optimizer | 2014 | 8.89E+05 | WOA [100] | Whale Optimization Algorithm | 2016 | 1.87E+03 |

# Poca diversidad

- Muchos algoritmos son prácticamente iguales. Aportan solo ruido.

IDEA: Añadir comparativa con algoritmo similar



*“the new population-based nature-inspired algorithms are released every month and, basically, they have nothing special and no novel features for science”, Fisher2016.*

*“Skepticism increases, since the results of each new algorithm are equivalent to those of the rest of the similar algorithms proposed by the same authors*.” Tzanetos2021

Fister I Jr, Mlakar U, Brest J, Fister I (2016) A new population-based nature-inspired algorithm every month: is the current era coming to the end. In: Proceedings of the 3rd student computer science research conference. University of Primorska Press, Berlin, pp 33–37.

Tzanetos y G. Dounias, «Nature inspired optimization algorithms or simply variations of metaheuristics?», Artif. Intell. Rev., vol. 54, n.o 3, pp. 1841-1862, mar. 2021, doi: 10.1007/s10462-020-09893-8.

# Exceso de complejidad

- *Algunas propuestas se han probado que son mejores quitando elementos.*

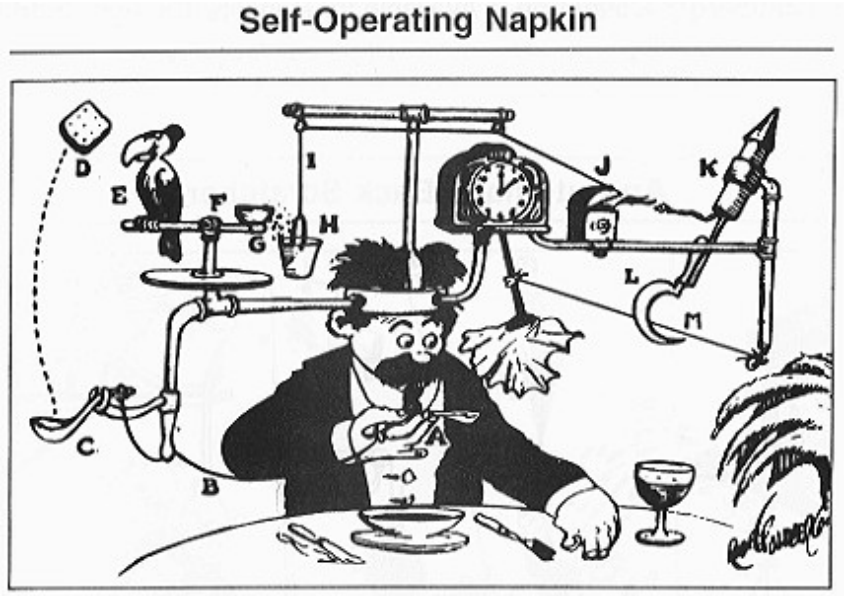
*"some metaheuristics have to be simplified because they contain operators that structurally bias their search by favouring sampling from some parts of the decision space" Piotrowski2018*

A. P. Piotrowski y J. J. Napiorkowski, «Some metaheuristics should be simplified», Information Sciences, vol. 427, pp. 32-62, feb. 2018, doi: 10.1016/j.ins.2017.10.039.

# Exceso de complejidad

*"La perfección se consigue, no cuando no haya más que añadir, sino cuando no hay nada más por quitar"*

***Antoine de Saint-Exupéry***




Source: wikipedia.org

IDEA: Test de Componentes para probar que todos ayudan

# Guías para comparar

- Benchmark: estándar para evitar sesgos, y comparar.
- Validación de resultados: Siempre estadística.
- Análisis de componentes: Todo aporte.
- Utilidad: Resultado, eficiencia, paralelismo, bien con pocas evaluaciones, …

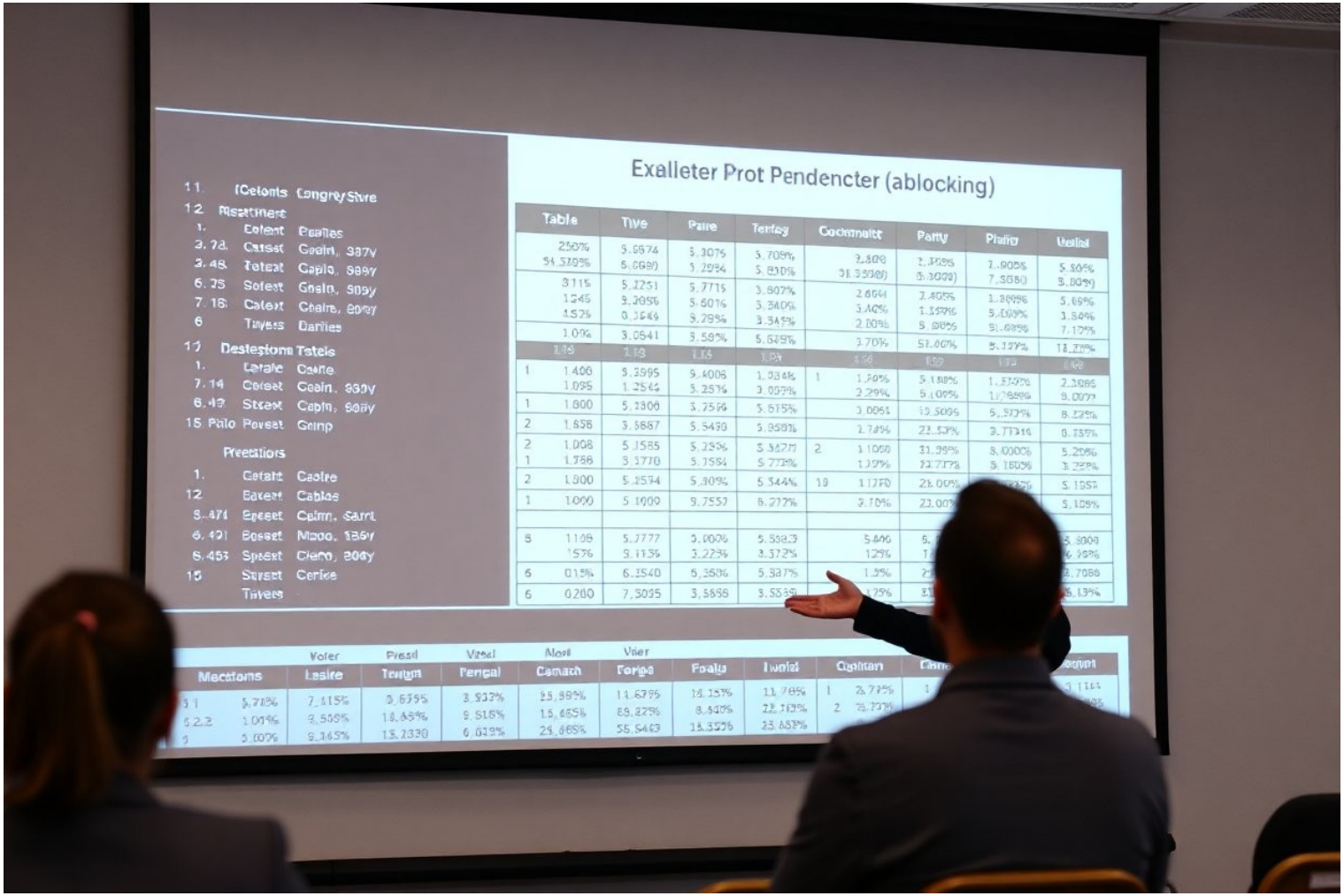
A. LaTorre, D. Molina, E. Osaba, J. Poyatos, J. Del Ser, y F. Herrera, «A prescription of methodological guidelines for comparing bio-inspired optimization algorithms», Swarm and Evolutionary Computation, vol. 67, p. 100973, dic. 2021, doi: 10.1016/j.swevo.2021.100973.

# Para problemas Reales

- Prioridad: Restricciones del problema.
- Debe ser replicable, y validable.
- Rendimiento es importante, pero también paralelizarlo.
- Resultado con distintos tiempos/evaluaciones.
- Valorar las prioridades del problema real.

E. Osaba et al., «A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems», Swarm and Evolutionary Computation, vol. 64, p. 100888, jul. 2021, doi: 10.1016/j.swevo.2021.100888.

# ¿Cómo mostrar resultados?



# Por medias

- Por medias.
- Contar cuantas veces uno es mejor.
- No valora algoritmos robustos.

| | AEO | DE | PSO | SSA |
|---|---|---|---|---|
| **F01** | 3.46E+08 | 1.52E+05 | 4.88E+09 | 3.77E+03 |
| **F02** | 1.00E+00 | 5.60E+19 | 1.00E+00 | 1.00E+00 |
| **F03** | 3.65E+04 | 4.49E+03 | 5.73E+04 | 8.25E-05 |
| **F04** | 2.42E+02 | 8.53E+01 | 1.25E+03 | 8.47E+01 |
| **F05** | 2.60E+02 | 2.04E+02 | 2.26E+02 | 2.65E+02 |
| **F06** | 6.99E+01 | 7.51E+00 | 3.81E+01 | 6.40E+01 |
| **F07** | 5.43E+02 | 2.34E+02 | 3.66E+02 | 4.66E+02 |
| **F08** | 2.01E+02 | 1.92E+02 | 1.81E+02 | 2.21E+02 |
| **F09** | 6.06E+03 | 2.31E+02 | 3.07E+03 | 6.21E+03 |
| **F10** | 6.12E+03 | 3.87E+03 | 7.07E+03 | 4.85E+03 |
| **F11** | 4.86E+02 | 8.30E+01 | 1.27E+03 | 1.55E+02 |
| **F12** | 1.10E+08 | 5.45E+05 | 4.09E+08 | 2.80E+06 |
| **F13** | 2.65E+06 | 1.92E+02 | 6.55E+07 | 1.93E+05 |
| **F14** | 1.81E+04 | 7.12E+01 | 3.27E+05 | 3.47E+03 |
| **F15** | 6.17E+04 | 6.59E+01 | 3.60E+05 | 8.89E+04 |
| **F16** | 2.04E+03 | 1.36E+03 | 1.61E+03 | 1.67E+03 |
| ... | ... | ... | ... | ... |
| **Mejor En** | 1 | 24 | 2 | 3 |

# Por Ranking

- Ordena cada algoritmo por problema/instancia.
- Calcula el ranking medio.
- Valora algoritmos robustos

| | AEO | DE | PSO | SSA |
|---|---|---|---|---|
| **F01** | 3 | 2 | 4 | 1 |
| **F02** | 2 | 4 | 2 | 2 |
| **F03** | 3 | 2 | 4 | 1 |
| **F04** | 3 | 2 | 4 | 1 |
| **F05** | 3 | 1 | 2 | 4 |
| **F06** | 4 | 1 | 2 | 3 |
| **F07** | 4 | 1 | 2 | 3 |
| **F08** | 3 | 2 | 1 | 4 |
| **F09** | 3 | 1 | 2 | 4 |
| **F10** | 3 | 1 | 4 | 2 |
| **F11** | 3 | 1 | 4 | 2 |
| **F12** | 3 | 1 | 4 | 2 |
| **F13** | 3 | 1 | 4 | 2 |
| **F14** | 3 | 1 | 4 | 2 |
| **F15** | 2 | 1 | 4 | 3 |
| **F16** | 4 | 1 | 2 | 3 |
| ... | ... | ... | ... | ... |
| **Promedio** | 3.20 | 1.27 | 2.90 | 2.63 |

# ¿Es suficiente?

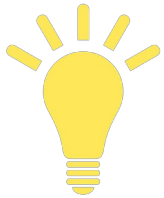
- Por media

| | AEO | DE | PSO | SSA |
|---|---|---|---|---|
| **F01** | 3.46E+08 | 1.52E+05 | 4.88E+09 | 3.77E+03 |
| **F02** | 1.00E+00 | 5.60E+19 | 1.00E+00 | 1.00E+00 |
| **F03** | 3.65E+04 | 4.49E+03 | 5.73E+04 | 8.25E-05 |
| **F04** | 2.42E+02 | 8.53E+01 | 1.25E+03 | 8.47E+01 |
| **F05** | 2.60E+02 | 2.04E+02 | 2.26E+02 | 2.65E+02 |
| **F06** | 6.99E+01 | 7.51E+00 | 3.81E+01 | 6.40E+01 |
| **F07** | 5.43E+02 | 2.34E+02 | 3.66E+02 | 4.66E+02 |
| **F08** | 2.01E+02 | 1.92E+02 | 1.81E+02 | 2.21E+02 |
| **F09** | 6.06E+03 | 2.31E+02 | 3.07E+03 | 6.21E+03 |
| **F10** | 6.12E+03 | 3.87E+03 | 7.07E+03 | 4.85E+03 |
| **F11** | 4.86E+02 | 8.30E+01 | 1.27E+03 | 1.55E+02 |
| **F12** | 1.10E+08 | 5.45E+05 | 4.09E+08 | 2.80E+06 |
| **F13** | 2.65E+06 | 1.92E+02 | 6.55E+07 | 1.93E+05 |
| **F14** | 1.81E+04 | 7.12E+01 | 3.27E+05 | 3.47E+03 |
| **F15** | 6.17E+04 | 6.59E+01 | 3.60E+05 | 8.89E+04 |
| **F16** | 2.04E+03 | 1.36E+03 | 1.61E+03 | 1.67E+03 |
| ... | ... | ... | ... | ... |
| **Mejor En** | 1 | 24 | 2 | 3 |

- Por ranking

| | AEO | DE | PSO | SSA |
|---|---|---|---|---|
| **F01** | 3 | 2 | 4 | 1 |
| **F02** | 2 | 4 | 2 | 2 |
| **F03** | 3 | 2 | 4 | 1 |
| **F04** | 3 | 2 | 4 | 1 |
| **F05** | 3 | 1 | 2 | 4 |
| **F06** | 4 | 1 | 2 | 3 |
| **F07** | 4 | 1 | 2 | 3 |
| **F08** | 3 | 2 | 1 | 4 |
| **F09** | 3 | 1 | 2 | 4 |
| **F10** | 3 | 1 | 4 | 2 |
| **F11** | 3 | 1 | 4 | 2 |
| **F12** | 3 | 1 | 4 | 2 |
| **F13** | 3 | 1 | 4 | 2 |
| **F14** | 3 | 1 | 4 | 2 |
| **F15** | 2 | 1 | 4 | 3 |
| **F16** | 4 | 1 | 2 | 3 |
| ... | ... | ... | ... | ... |
| **Promedio** | 3.20 | 1.27 | 2.90 | 2.63 |

¿Y si es por suerte/ruido?



IDEA: Test Estadísticos

# Tipos de Tests Estadísticos

¿Diferencias?

Paramétricos

No Paramétricos

# Diferencias entre ambos

- Trabajar con valores de diferencia o rangos.
- Mejor para continuo, o para ordinales.
- Más potentes, o menos.
- Requiere cumplir una distribución conocida, o no.

# Condiciones a comprobar

- Escala: Estar en misma escala, sin *outliers*.
- Independencia: Ser independientes entre sí.
- Normalidad: Distribución similar a normal.
- Homocedasticidad: Varianza similar.

# Comprobar Normalidad

- Shapiro-Wilk, de 1965.
  - Mejor con menos muestras.
  - Valora diferencias entre valores extremos, y compara con normal.
- Kolmogorov-Smirnov.
  - Considera probabilidades.
  - Menos potente.

# Comprobar Varianza

- Test de Levene.
  - Requiere normalidad.
  - Más potente.
- Bartlett.
  - No requiere normalidad.
  - Menos potente.

# Test estadísticos

| Muestra | Prueba paramétrica | Prueba no paramétrica |
|---|---|---|
| *Muestras relacionadas* | | |
| 2 muestras | t-Student | Wilcoxon |
| > 2 muestras | ANOVA | Friedman |
| *Muestras independientes* | | |
| 2 muestras | t-Student | U de Mann-Whitney |
| > 2 muestras | ANOVA | Kruskal-Wallis |

# Tests estadísticos

- Paramétricos
  - T-Student
  - ANOVA
- No-Paramétricos
  - Wilcoxon
  - Kruskal-Wallis
- Post-hoc
  - Holm
  - Bonferroni
  - Hochberg

# Test de Student

- De William Sealy Gosset, 1908.
- Student fue un seudónimo por proteger secreto industrial.
- Permite comparar dos secuencias.
- Una secuencia por algoritmo.

Fuente: wikipedia.org

# Test de ANOVA

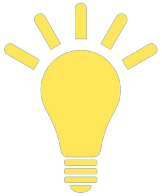
- De R.A. Fisher, 1920-1930.
- Muy usado en biología y otras áreas.
- Permite comparar más de dos secuencias.
- Permite comparar varios a la vez, de 1 vía, 2 vías, ...

Fuente: wikipedia.org

# ¿Cuándo usarlas?

- Difícil cumplir las condiciones de Paramétricos.
- No se cumple en benchmarks conocidos.
- No lo digo yo, lo dice:

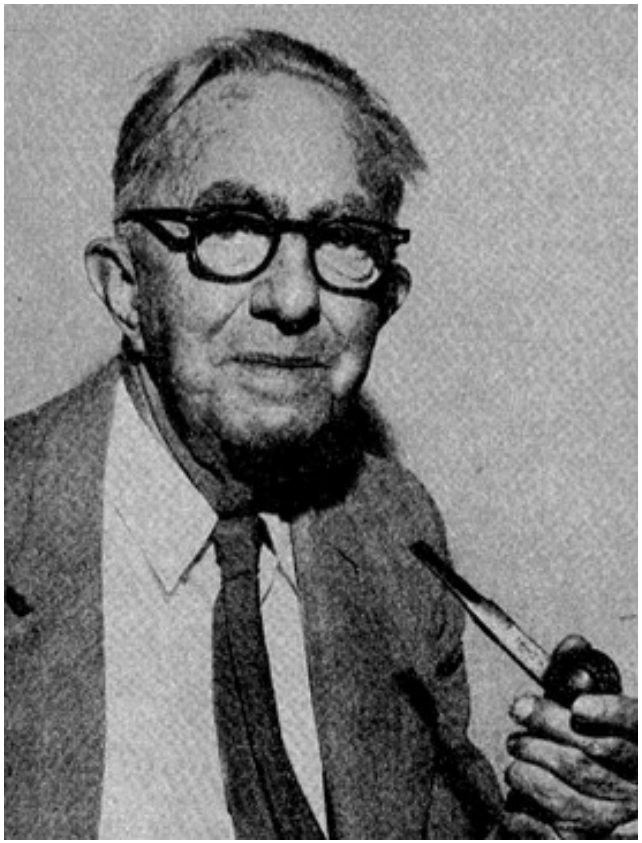
J. Derrac, S. García, D. Molina, y F. Herrera, «A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms», Swarm and Evolutionary Computation, vol. 1, n.o 1, pp. 3-18, mar. 2011, doi: 10.1016/j.swevo.2011.02.002.



Conclusión: No usar test paramétricos, centrarse en los no paramétricos

# Wilcoxon

- Hay dos tests distintos:
  - *Wilcoxon rank-sum* test, llamado *Mann–Whitney U* test.
  - *Wilcoxon signed-rank* test.
- Se aplican distinto:
  - Datos emparejados (*data paired*).



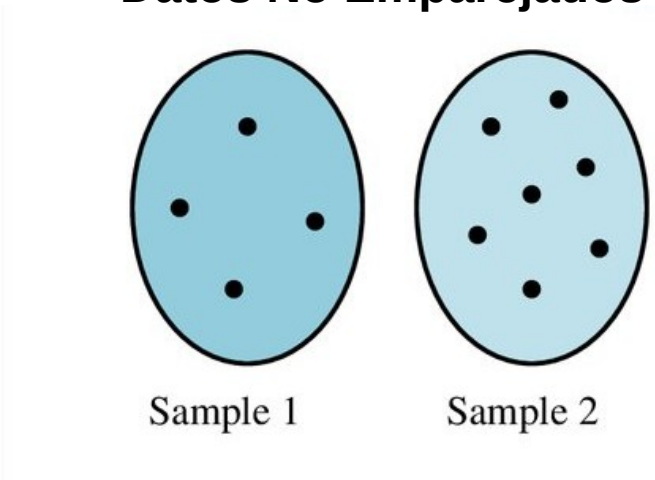
Fuente: wikipedia.org

# ¿Qué son datos emparejados?

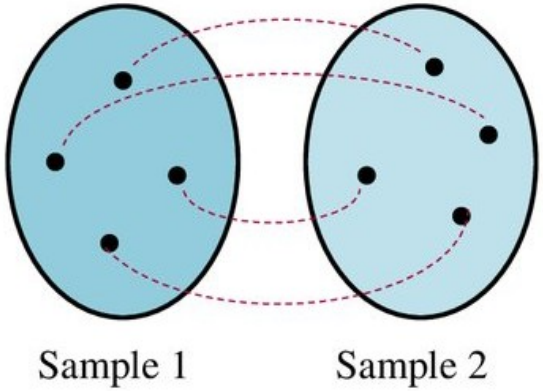
- Pueden tener mismo o distinto tamaño.
- Distintas ejecuciones, sin orden claro.

- Hay correspondencia.
- Ejemplo: Resultados por cada función/instancia.

**Datos No Emparejados**




**Datos Emparejados**



# U de Mann-Whitney(-Wilcoxon)

- Múltiples ejecuciones por problema/instancia.
- Compara las ejecuciones de un algoritmo y otro.
- Cuanta los problemas con diferencias significativas y sentido.

Ejemplo

| A1 vs Resto | A2 | A3 |
|---|---|---|
| Significant Better (+) | 9 | 7 |
| Significant Worse (-) | 4 | 1 |
| Not significant (~) | 2 | 7 |

# Cómo se calcula

- En R: wilcox.test con paired=FALSE

  > x=rnorm(30, mean=0.4);y=rnorm(30, mean=0.8)

  > wilcox.test(x, y, paired=FALSE)

  ...

  W = 511, p-value = 0.3738

# Wilcoxon signed-rank

- Múltiples instancias/problemas.
- Los datos están emparejados.
- Indica si hay diferencia significativa en el total de instancias o ejemplos, comparando por cada función.

Ejemplo

| | A2 | A3 |
|---|---|---|
| A1 | ~ | + |

# Wilcoxon de pares con R

- Usando R:

  x=c(59, 23, 91, 20, 50, 35, 17, 50, 14, 64, 48);

  y=c(61, 28, 77, 20, 48, 41, 11, 50, 10, 63, 44)

  wilcox.test(x, y, paired=TRUE)

  Wilcoxon signed rank test with continuity correction data: x and y

  V = 29, p-value = 0.476

El valor obtenido es mucho mayor que 0.05, por lo que no se detecta diferencia.

# Wilcoxon single rank, cálculo

- Tenemos dos secuencias: $X_i$ y $Y_i$.
- Se calcula la diferencia

$$z_i = x_i - y_i$$

- Se calcula los ranking, se promedian los iguales:
- Se separan los calores positivos y negativos:

$$R_i = ranking|z_i|$$

- Se calcula W.

$$W^+ = \sum_{z_i>0} R_i \qquad W^- = \sum_{z_i<0} R_i \qquad W = min(W^+, W^-)$$

- Se compara W con el valor de referencia, para obtener el p-value:

  https://real-statistics.com/statistics-tables/wilcoxon-signed-ranks-table/

- Si p-value es menor que 0.05 se suele considerar mejora significativa.

# Ejemplo

- Valores:

  x=[59, 23, 91, 20, 50, 35, 17, 50, 14, 64, 48]

  y=[61, 28, 77, 20, 48, 41, 11, 50, 10, 63, 44]

  z=[-2, 5, 14, 0, 2, -6, 6, 0, 4, 1, 4],

  |z|=[2, 5, 14, 0, 2, 6, 6, 0, 4, 1, 4],

  sign=[-, -, +, 0, +, -, +, 0, +, +, +]

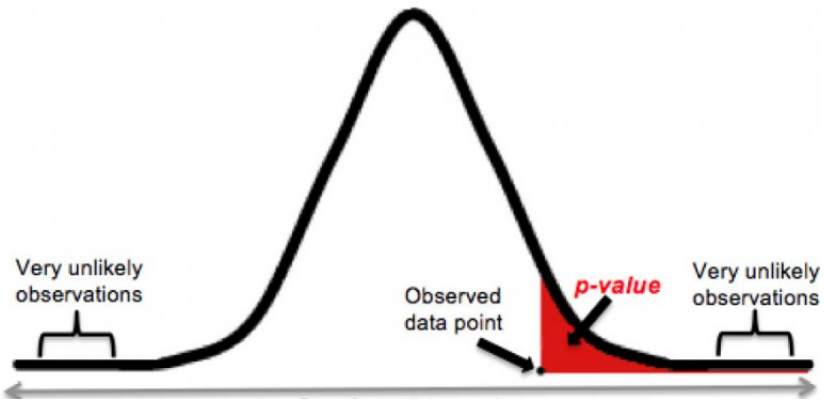
  Ranking = [2.5, 6, 9, -, 2.5, 7.5, 7.5, -, 4.5, 1, 4.5],

  W+=29, W-=16, W=min(29, 16)=16

  Referencia con N=11, alpha=0.05, da 10, por tanto no detecta diferencias significativas.

# Cómo interpretar p-value

- *Es la probabilidad de obtener resultados tan extremos como los observados, o más, si la hipótesis nula es verdadera*.
- Valor alto no indica que no haya diferencia.
- p-value se puede ver como la probabilidad de que no sea correcto afirmar que haya diferencias significativas.



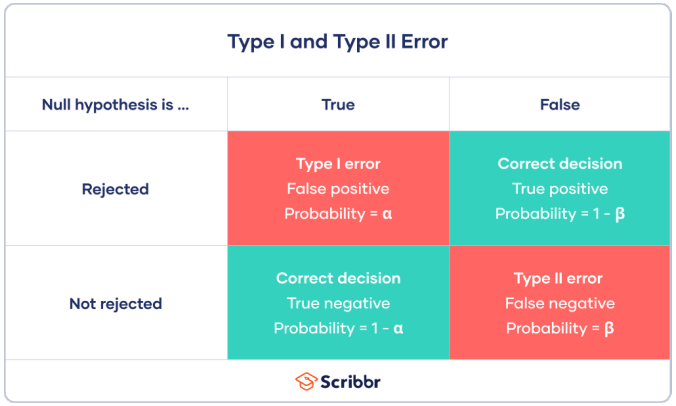
# Conflicto sobre p-value

- Es una probabilidad, no da certeza.
- Valorar el valor, no solo como umbral de 5%.
- Más *fácil* detectar cambios con más muestras.
  - Mejora en 2 de 10 (20%) ⇒ p-value=0.26 > 0.05
  - Mejora en 10 de 50 (20%) ⇒ p-value=0.02 < 0.05

O. Y. Chén et al., «The roles, challenges, and merits of the p value», Patterns (N Y), vol. 4, n.o 12, p. 100878, dic. 2023, doi: 10.1016/j.patter.2023.100878.

# Potencia Estadística o sensibilidad

- Tipo I: Identificar diferencias que no hay.
- Tipo II: No detectar diferencias.




- Buena potencia reduce el riesgo de Tipo II.
- Depende de:
  - Nivel de significancia.
  - Tamaño muestra.

# Error acumulado

¿Es correcto comparar 5 Algoritmos entre sí con Wilcoxon por pares (20 veces)?

IDEA: Aplicar Tests Post-hoc

# Post-hoc

- Permite adjustar los p-values para mantener controlado el error.
- Varios: Bonferroni, Holm, ..., Hochberg (muy similares)
- En R es muy fácil, basta poner p.adjust en los parámetros.

# Post-hoc

- Holm
  - Empieza de p bajo a alto.
  - Hace:

$$\alpha'(i) = \frac{\alpha}{m - i - 1}$$

  - Si p >= ά, acepto el resto de hipótesis nulas.

- Hochberg
  - Empieza de p alto a bajo.
  - Hace:

$$\alpha'(i) = \frac{\alpha}{m - i - 1}$$

  - Si p < ά(i) rechaza el resto de hipótesis nulas.

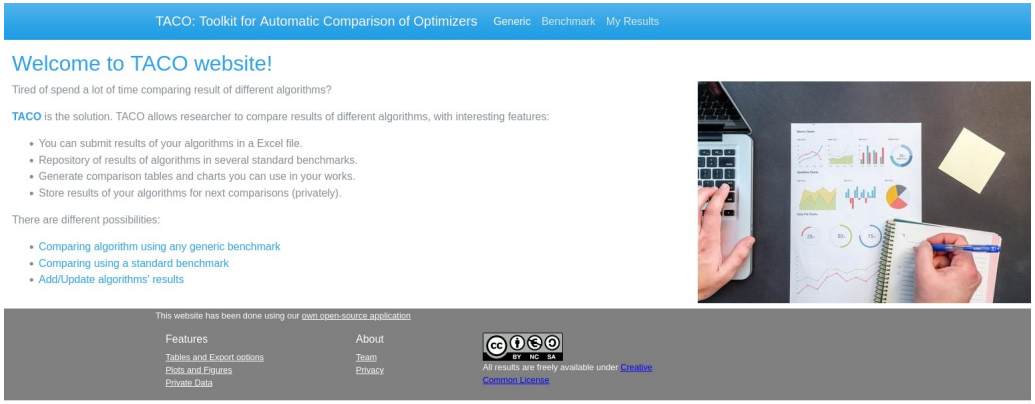
# Ejemplo

```
> library('tibble')

> data=tibble("alg"=c(rep("A1",7), rep("A2", 7), rep("A3", 7)),
"value"=c(c(3,2,2,3,1,3,3), c(1, 1, 3, 1, 2, 2, 1), c(2, 3, 1, 2, 3, 1,
2)))

> data$alg = as.factor(data$alg)

> # Sin post-hoc pairwise.wilcox.test(data$value, data$alg)

# Con post-hoc

> pairwise.wilcox.test(data$value, data$alg,
p.adjust="hochberg")
```

# Ejemplo

```
> library('tibble')

> data=tibble("alg"=c(rep("A1",7), rep("A2", 7), rep("A3", 7)),
"value"=c(c(3,2,2,3,1,3,3), c(1, 1, 3, 1, 2, 2, 1), c(2, 3, 1, 2, 3, 1,
2)))

> data$alg = as.factor(data$alg)

> # Sin post-hoc pairwise.wilcox.test(data$value, data$alg)

# Con post-hoc

> pairwise.wilcox.test(data$value, data$alg,
p.adjust="hochberg")
```

# Tacolab https://tacolab.org




- Permite realizar los tests a partir de un Excel.
- Posee resultados de algunos algoritmos y benchmarks, pero se puede usar de forma genérica.
- Genera tablas exportables a Excel y Latex.

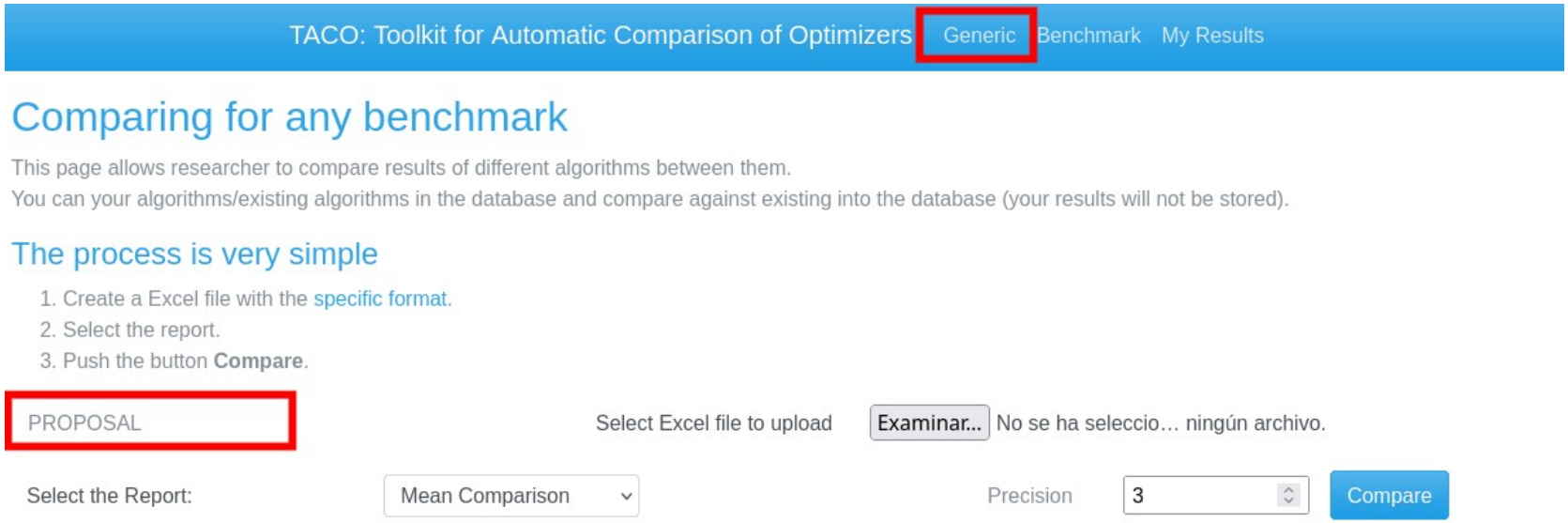
# Formato de tabla

| alg | F1 | F2 | F3 | F4 | F5 | F6 | F7 |
|---|---|---|---|---|---|---|---|
| A1 | 30 | 2.5 | 40 | 70 | 5 | 0.3 | 5 |
| A2 | 5 | 1.5 | 60 | 40 | 7 | 0.2 | 3 |
| A3 | 12 | 8 | 5 | 60 | 12 | 0.1 | 4 |

- *alg*: Nombre de algoritmo (o en pestaña).
- Columnas Fx: Funciones/Instancias.
- Puede haber por algoritmo varias salidas, para los tests estadísticos.

# Usando Tacolab

- Elegir General



# Usando Tacolab

PROPOSAL

Select Excel file to upload Examinar... datos_comparar.xlsx

Select the Report: Mean Comparison

Precision 3 Compare

## Evaluations: 100%

| Functions | A1 | A2 | A3 |
|---|---|---|---|
| F01 | 3.000e+01 | 5.000e+00 | 1.200e+01 |
| F02 | 2.500e+00 | 1.500e+00 | 8.000e+00 |
| F03 | 4.000e+01 | 6.000e+01 | 5.000e+00 |
| F04 | 7.000e+01 | 4.000e+01 | 6.000e+01 |
| F05 | 5.000e+00 | 7.000e+00 | 1.200e+01 |
| F06 | 3.000e-01 | 2.000e-01 | 1.000e-01 |
| F07 | 5.000e+00 | 3.000e+00 | 4.000e+00 |
| Best | 1 | 4 | 2 |

Excel Latex

# Usando Tacolab

Select the Report: Ranking Comparison

Precision 3 Compare

## Evaluations: 100%

| Functions | A1 | A2 | A3 |
|---|---|---|---|
| F01 | 3 | 1 | 2 |
| F02 | 2 | 1 | 3 |
| F03 | 2 | 3 | 1 |
| F04 | 3 | 1 | 2 |
| F05 | 1 | 2 | 3 |
| F06 | 3 | 2 | 1 |
| F07 | 3 | 1 | 2 |
| Mean | 2.429 | 1.571 | 2.000 |

Excel Latex

# Usando Tacolab

Select the Report: Non-Parametric Tests

Precision 3 Compare

## Wilcoxon p-values

| Algorithms | A1 | A2 | A3 | Accumulated Error (%) |
|---|---|---|---|---|
| A1 | 1.000 | 0.397 | 0.297 | 9.8 |
| A2 | 0.397 | 1.000 | 0.375 | 9.8 |
| A3 | 0.297 | 0.375 | 1.000 | 9.8 |

Excel Latex

## Description of tables

- **Wilcoxon p-values** shows the result of Wilcoxon' test (the p-value) of the algorithm in each row (first one) with the algorithm in each column (second one).
  - Meaning of the colors: Blue value indicates that first algorithm is better than second one, and red when the first one is worse than second one, with a signific a threshold of 10%. Black colors mean not detected significant differences.
  - Accumulated error: For each comparison there is a statistical error, thus compare one algorithm with all the other implies an accumulated error (shown in las
- **Multiple comparison's table** shows the results of p-values with different corrections to maintain reduced the total/accumulated error in the comparisons. It select In that case, the complete error is lower than 5%, so it is more conservative than Wilcoxon' test.

  It is only shown if any significant difference between the algorithms is detected (using Iman-Davenport's test).

# Usando Tacolab

Select the Report: Non-Parametric Tests  Precision 3  Compare

## Wilcoxon p-values

| Algorithms | AEO | EBOwithCMAR | ELSHADE_SPACMA | Accumulated Error (%) |
|---|---|---|---|---|
| AEO | 1.000 | 2.702E-06 | 1.863E-09 | 9.8 |
| EBOwithCMAR | 2.702E-06 | 1.000 | 0.226 | 9.8 |
| ELSHADE_SPACMA | 1.863E-09 | 0.226 | 1.000 | 9.8 |

Excel Latex

## Multiple comparisons' tests p-values (Global Error: 5%)

| Algorithms vs EBOwithCMAR | Original | Holm | Hommel | Hochberg |
|---|---|---|---|---|
| AEO | 2.702E-06 | 5.405E-06 | 5.405E-06 | 5.405E-06 |
| ELSHADE_SPACMA | 0.226 | 0.226 | 0.226 | 0.226 |

Excel Latex

# Conclusiones

- Comparar algoritmos no es trivial.
- Paradoja: muchas propuestas no son útiles.
- Necesario aportar ventajas usando tests estadísticos.
- Utilizar tests no paramétricos.
- Evitar comparar varias sin usar post-hoc.
- Uso con R, o usando tacolab.

# Muchas gracias


Daniel Molina Cabrera
dmolinac@ugr.es