

# Topologías Dinámicas en la Optimización por Cúmulo de Partículas para Problemas con Restricciones

## Borrador I y Proyecto Final

**Arturo Márquez-Flores**

Maestría en Inteligencia Artificial

Universidad Veracruzana

CIIA – Centro de Investigación en Inteligencia Artificial

Sebastián Camacho No 5, Xalapa, Ver., México 91000

15 de Mayo del 2019

## Introducción

La optimización por cúmulo de partículas es un método de optimización estocástica basado en poblaciones propuesto por Kennedy y Eberhart en [KE95]. Dos de sus ventajas más notables son su fácil implementación y su rápida convergencia a soluciones aceptables.

Desde 1995 se han propuesto numerosas modificaciones y variantes al modelo original. Como se menciona en la revisión de literatura en [BM17], las principales limitaciones de PSO están relacionadas con su convergencia y con su invarianza ante transformaciones. Para superar estas limitaciones se han propuesto cambios en los parámetros; en particular, en la topología de comunicación entre los individuos de la población, los coeficientes del modelo y el tamaño poblacional. Por otro lado se han propuesto modificaciones a las reglas de actualización de velocidad y posición. Por último, se han propuesto modelos de *hibridización*.

En general, las modificaciones y variantes que se han propuesto se han enfocado en problemas de optimización libres de restricciones. Aunque también existe literatura que estudia el desempeño del PSO en problemas con restricciones, esta es más limitada.

Dentro del enfoque sin restricciones, la topología de comunicación en el PSO se ha limitado, por lo general, al caso del grafo completo (global best). Sin embargo, se ha estudiado cómo la modificación de la topología impacta sobre

la convergencia y diversidad de la población. El consenso, según Blackwell y Kennedy, en [BK18], es que un grafo completo tiende a generar una convergencia más rápida, mientras estructuras con más *localidad* como la topología de anillo tienden a favorecer la diversidad. A pesar de este consenso general, esta es un área sin resultados contundentes y existen todavía preguntas abiertas. Esto es aún más marcado tanto para el caso de problemas con restricciones como para el caso de topologías adaptativas.

La importancia de estudiar la dinámica de las topologías de comunicación trasciende al caso específico del algoritmo PSO, e incluso al campo de cómputo evolutivo. En efecto, su relevancia general se muestra en la literatura reciente sobre *cerebros sólidos y líquidos* en [SMF19] y [VEMF19], que propone estudiar las diferencias entre *redes cognitivas* con conexiones estables (como las redes neuronales) y aquellas con conexiones inestables, determinadas por el movimiento de sus componentes, como las colonias de abejas, termitas, o sistemas inmunológicos.

En este trabajo se propone estudiar con detalle el impacto de una topología de comunicación dinámica del PSO sobre la convergencia y diversidad de la población para el caso de problemas con restricciones.

## Revisión Bibliográfica

Desde la primera formulación del PSO en 1995 ([KE95]), la primera investigación formal sobre el impacto de la topología de comunicación en el PSO sobre su desempeño aparece en [Ken99], en el cual se estudian cuatro topologías diferentes: local, global, rueda y estrella. Estas topologías se experimentan en cuatro funciones de referencia, y la conclusión general es que la topología global fomenta la explotación del problema, mientras la topología local fomenta la exploración del problema. De tal manera, la convergencia bajo una topología global es más rápida que bajo una local. Siguiendo esta dirección, en [KM02] y [MKN04] se investigó el mismo descubrimiento con una formulación más general del PSO y más topologías. Sin embargo, dichos estudios fueron criticados por su falta de rigor estadístico. En [Eng13] una investigación estadísticamente más extensiva mostró que la diferencia en el desempeño del PSO con estas topologías es altamente dependiente de la función objetivo en cuestión. No obstante, más recientemente se muestra en [BK18] que cuando se utiliza el modelo estándar de PSO el desempeño de topologías con más *localidad* como la topología local es mejor en problemas *más difíciles* ya que estas persisten cuando una topología global ya habría convergido. De tal manera, la rapidez de la topología global es una ventaja cuando los problemas son relativamente simples.

Otras topologías han sido consideradas en la literatura, como la jerarquía dinámica en [JM05], la topología libre de escala en [ZY11] y la topología de mundo pequeño en [GZ13].

Por otro lado, se han propuesto topologías dinámicas para aprovechar las ventajas de exploración y explotación de las topologías local y global, respectivamente. En [Cle06] se propone el algoritmo TRIBES en el cual la topología se adapta en el tiempo de acuerdo con la retroalimentación del desempeño. Por otro lado, en [Sug99] se desarrolla un algoritmo que inicialmente toma la topología local y gradualmente incrementa la vecindad hasta converger, al final de las iteraciones, en una topología global. Esto permite aprovechar la exploración en la fase inicial del algoritmo y la explotación en la fase final. En [MM13] se propone un algoritmo similar. En [LI14] los autores estudian diferentes grados de conectividad en la topología para variar la preferencia por la exploración o la explotación del algoritmo PSO-ITC.

En el área de PSO para problemas con restricciones (COP por sus siglas en inglés), [PV02] fue uno de los primeros abordajes. En él se utilizaron funciones de penalización para lidiar con las restricciones. Otros métodos para la satisfacción de restricciones como la preservación de soluciones factibles, la cercanía a la región factible y funciones de penalización fueron utilizados en [HE02], [PC04] y [CH03], respectivamente. Otras aproximaciones a este problema se encuentran en [PE07], [HW07], [BLM13] y [ESMM13].

En general, la investigación que se encuentra en la intersección entre topo-

logías dinámicas y COPs es poca. Sin embargo, una referencia importante es el trabajo que comienza en [BM14] con el desarrollo de EMLPSO, que se basa en manejo de restricciones con nivel  $\epsilon$  (ELCH). Este algoritmo es extendido en [BLM14] con una topología dinámica que extiende la utilizada en [Sug99].

## Marco Teórico

Un problema de optimización (minimización) se define de la siguiente manera. Dada una función  $f : S \rightarrow \mathbb{R}$  ( $S \subseteq \mathbb{R}^d$ ), se busca un  $x \in S$  tal que  $\forall x' \in A$  se tiene que  $f(x) \leq f(x')$ . El algoritmo de PSO ([KE95]) cuenta con una *población* de  $N$  soluciones potenciales que contienen la siguiente información:

- Posición -  $x_t^i \in S$  de la partícula  $i$  en la iteración  $t$ .
- Velocidad -  $v_t^i$  es la dirección y magnitud del movimiento de la partícula  $i$  en la iteración  $t$ .
- Mejor posición personal -  $p_t^i$  es la mejor posición que la partícula  $i$  ha visitado en las iteraciones  $1 \dots t$ .

En su forma más general, en cada iteración, y para cada partícula, se realiza una actualización de estos valores de acuerdo con las siguientes ecuaciones:

$$v_{t+1}^i = \mu(x_t^i, v_t^i, N_t^i) \quad (1)$$

$$x_{t+1}^i = \xi(x_t^i, v_t^i) \quad (2)$$

$$p_{t+1}^i = \begin{cases} x_{t+1}^i & \text{if } f(x_{t+1}^i) < f(p_t^i) \\ p_t^i & \text{if } f(x_{t+1}^i) \geq f(p_t^i) \end{cases} \quad (3)$$

En la ecuación 1,  $N_t^i$  es el conjunto de vecinos de la partícula  $i$  en el tiempo  $t$ ; es decir, el conjunto de partículas que son adyacentes a  $i$  bajo cierta topología en el tiempo  $t$ <sup>1</sup>. Esto contiene la información de la mejor posición personal de cada vecino en dicha vecindad. De tal manera, la ecuación 1 actualiza la velocidad de cada partícula, la ecuación 2 actualiza la posición de cada partícula, y la ecuación 3 actualiza la mejor posición personal obtenida para cada partícula.

En la versión original del PSO ([KE95]), el conjunto  $N_t^i$  es igual al conjunto total de partículas involucradas; es decir, se considera una topología completamente conectada. Además, la ecuación 1 se define de la siguiente manera:

$$v_{t+1}^i = v_t^i + \phi_1 R_{1,t}^i (p_t^i - x_t^i) + \phi_2 R_{2,t}^i (g_t^i - x_t^i) \quad (4)$$

En esta ecuación  $\phi_1$  y  $\phi_2$  son dos números reales, mientras  $R_{1,t}^i$  y  $R_{2,t}^i$  son dos matrices diagonales de dimensión  $d \times d$  con entradas aleatorias distribuidas  $\mathcal{U}(0, 1)$ . Por otro lado,  $g_t^i$  está dado por la siguiente ecuación:

---

<sup>1</sup>Asumimos que una partícula siempre es vecina de sí misma.

$$g_t^i = \operatorname{argmax}_{p_t^j \in P_t} \{f(p_t^j) : j \in N_t^i\} \quad (5)$$

En esta ecuación,  $P_t = \{p_t^k : k \in \{1, 2, \dots, N\}\}$ .

En [SR98], se modificó el algoritmo para incluir el coeficiente  $\omega$ , de tal manera que la ecuación 4 ahora se convierte en la siguiente:

$$v_{t+1}^i = \omega v_t^i + \phi_1 R_{1,t}^i(p_t^i - x_t^i) + \phi_2 R_{2,t}^i(g_t^i - x_t^i) \quad (6)$$

Con la ecuación 6, el pseudocódigo del algoritmo PSO es el que se muestra en el Algoritmo 1.

---

**Algorithm 1:** PSO

---

```

1 for cada partícula  $i$  en enjambre do
2   Inicializar posición,  $x_0^i$ , generando un vector distribuido uniformemente
   en  $[p_{lb}, p_{ub}]^d \subseteq S$ 
3   Inicializar la mejor posición,  $p_0^i$ , evaluando la posición en la función
   objetivo.
4   Inicializar velocidad,  $v_0^i$ , generando un vector distribuido uniformemente
   en  $[v_{lb}, v_{ub}]^d \subseteq S$ 
5   Inicializar el mejor global:  $g_0^i = \operatorname{argmax}_{p_0^j \in P_0} \{f(p_0^j) : j \in N_0^i\}$ 
6 while criterio de terminación no se cumpla do
7   for cada partícula  $i$  en enjambre do
8     Actualizar velocidad con la ecuación 1
9     Actualizar posición con la ecuación 2
10    Actualizar mejor posición con la ecuación 3
11    Actualizar el mejor global con la ecuación 5
12 return  $\operatorname{argmax}_{p_t^j \in P_t} f(p_t^j)$ 

```

---

El algoritmo 1 está diseñado para problemas de optimización sin restricciones. Un problema de optimización con restricciones (COP) es el siguiente. Dada una función  $f : S \rightarrow \mathbb{R}$  ( $S \subseteq \mathbb{R}^d$ ), se busca un  $x \in S$  tal que  $\forall x' \in A$  se tiene que  $f(x) \leq f(x')$  y además se satisface que para toda  $i = 1, \dots, p$ ,  $g_i(x) \leq 0$  y para toda  $j = 1, \dots, q$ ,  $h_j(x) = 0$ .

En este trabajo se propone una modificación simple al algoritmo general del PSO para resolver problemas de optimización con restricciones, siguiendo los pasos de [Sug99] y de [HE02]. Para esto especificamos varios aspectos relacionados al algoritmo general. Primeramente, dejamos los parámetros  $p_{lb}$  y  $p_{ub}$  indeterminados, pues estos dependerán del problema de optimización. Por otro

lado, los parámetros  $v_{lb}$  y  $v_{ub}$  los definimos en 0 y 1, respectivamente. Además, nuestro criterio de terminación está regido por el número de iteraciones realizadas.

En este trabajo se adopta una topología dinámica que incrementa la *conectividad* de las partículas en el tiempo, similar a la utilizada en [Sug99]. En particular, se considera una topología local que está caracterizada por las medidas de centralidad de grado de las partículas. Asumimos que la centralidad para todas las partículas es la misma. Inicialmente se considera la topología de centralidad 1 y la centralidad aumenta sucesivamente hasta llegar a  $N - 1$ ; es decir, la topología converge a un grafo completamente conectado. Esta topología determina  $N_t^i$  para toda  $t$  y toda  $i$ , y, por lo tanto, también  $g_t^i$ . Para determinar el cambio en la topología aumentamos el grado de los nodos por  $\lceil (N - 1)/G \rceil$  cada generación siempre y cuando el grado resultante sea menor a  $N - 1$ ; en otro caso, se asigna un grado de  $N - 1$ .

La segunda modificación relevante sigue el trabajo en [HE02], donde se propone un método de preservación de soluciones factibles para lidiar con problemas con restricciones (COPs). Esto afecta prácticamente todo el Algoritmo 1 ya que tanto las inicializaciones como la actualizaciones se realizan considerando sólo las soluciones que son factibles en el problema. Particularmente hay un cambio fuerte en el ciclo for de las líneas 1 a 5. La inicialización de la mejor posición sea realiza sólo si  $x_0^i$  es factible, y el mejor global se determina, para cada  $i$  considerando sólo el subconjunto de  $N_0^i$  que es factible. Por otro lado, la actualización de la velocidad en la línea 8 se ve afectada y se desglosa en 4 casos posibles que se muestran en las siguientes ecuaciones, para cada  $i$ . Asumamos que  $NF_t^i \subseteq N_t^i$  es el subconjunto de soluciones factibles de  $N_t^i$ , para toda  $t$  e  $i$ .

Si al tiempo  $t$  la partícula  $i$  ha sido factible en alguna iteración (es decir,  $p_t^i$  ha sido inicializada) entonces:

$$v_{t+1}^i = \omega v_t^i + \phi_1 R_{1,t}^i(p_t^i - x_t^i) + \phi_2 R_{2,t}^i(g_t^i - x_t^i) \quad (7)$$

Si al tiempo  $t$  la partícula  $i$  no ha sido factible en alguna iteración (es decir,  $p_t^i$  no ha sido inicializada) y  $NF_t^i$  no es vacío entonces:

$$v_{t+1}^i = \omega v_t^i + \phi_2 R_{2,t}^i(g_t^i - x_t^i) \quad (8)$$

Si al tiempo  $t$  la partícula  $i$  no ha sido factible en alguna iteración (es decir,  $p_t^i$  no ha sido inicializada) y  $NF_t^i$  es vacío entonces:

$$v_{t+1}^i = \omega v_t^i + r_t^i \quad (9)$$

Donde  $r_t^i$  es un vector de dimensión  $d$  con entradas distribuidas  $\mathcal{U}(-1, 1)$ .

Con estas dos modificaciones, tenemos el algoritmo modificado en Algoritmo 2.

---

**Algorithm 2:** PSO

---

```

1 Inicializar topología local con grado  $k = 1$ 
2 for cada partícula  $i$  en enjambre do
3   Inicializar posición,  $x_0^i$ , generando un vector distribuido uniformemente
     en  $[p_{lb}, p_{ub}]^d \subseteq S$ 
4   if  $x_0^i$  es factible then
5     Inicializar la mejor posición,  $p_0^i$ , evaluando la posición en la función
     objetivo.
6   Inicializar velocidad,  $v_0^i$ , generando un vector distribuido uniformemente
     en  $[v_{lb}, v_{ub}]^d \subseteq S$ 
7   if  $N_0^i \neq \emptyset$  then
8     Inicializar el mejor global:  $g_0^i = \operatorname{argmax}_{p_0^j \in P_0} \{f(p_0^j) : j \in N_0^i\}$ 
9 for cada generación  $g$ , hasta alcanzar  $G$  do
10  for cada partícula  $i$  en enjambre do
11    if  $p_t^i$  ha sido inicializada then
12      Actualizar velocidad con la ecuación 7
13    else
14      if  $N_t^i \neq \emptyset$  then
15        Actualizar velocidad con la ecuación 8
16      else
17        Actualizar velocidad con la ecuación 9
18    Actualizar posición con la ecuación 2
19    if  $x_{t+1}^i$  es factible then
20      Actualizar mejor posición con la ecuación 3
21    if  $N_{t+1}^i \neq \emptyset$  then
22      Actualizar el mejor global con la ecuación 5
23  if  $k + \lceil (N - 1)/g \rceil < N - 1$  then
24    Aumentar grado topología a  $k + \lceil (N - 1)/g \rceil$ 
25  else
26    Aumentar grado topología a  $N - 1$ 
27 return  $\operatorname{argmax}_{p_t^j \in P_t} f(p_t^j)$ 

```

---



## Variables de Estudio y Métodos de Evaluación

Para hacer la evaluación nos guiaremos por el benchmark del [reporte técnico](#) en [\[WMS16\]](#), que incluye 28 COPs con dimensiones 10, 30, 50 y 100. Siguiendo el reporte, realizamos 25 corridas y se reporta la mejor solución encontrada después de 2000D, 10000D y 20000D evaluaciones en la función objetivo, donde D es la dimensionalidad del problema. Se reportará la mejor y la peor solución, la media, mediana y desviación estándar de las 25 corridas. Se incluirá el número de violaciones a las restricciones para diferentes grados de tolerancia (1, 0.01 y 0.0001). Además se reportará la tasa de factibilidad y la complejidad del algoritmo de acuerdo a lo definido en el reporte.

## Resultados Preliminares

Para obtener resultados preliminares en este primer borrador se utilizará un subconjunto de los COPs definidos en [WMS16], que excluye problemas rotados <sup>2</sup>. En particular, consideramos los problemas C01, C03, C04, C06, C07, C11, C13 y C19, para dimensiones 10 y 30 y con límites de evaluaciones en la función objetivo en 2,000, 10,000 y 20,000. Reportamos la mejor y la peor solución, la media, mediana y desviación estándar de las 25 corridas. Para estos resultados preliminares omitimos las otras métricas del reporte técnico.

En este experimento se utilizaron 200 partículas, 100 iteraciones,  $c_1 = 0.6$ ,  $c_2 = 0.3$  y  $w = 0.4$ . Los resultados pueden replicarse en [este colab](#), y la librería utilizada se encuentra en [este repositorio](#).

Cuadro 1: Resultados con dimensión = 10

FES	Estadística	C01	C03	C04	C06	C07	C11	C13	C19
<b>2x10<sup>3</sup></b>	<b>Mejor</b>	1728.45	inf	110.44	inf	inf	inf	inf	inf
	<b>Peor</b>	5111.24	inf	196.47	inf	inf	inf	inf	inf
	<b>Media</b>	3675.8	inf	151.51	inf	inf	inf	inf	inf
	<b>Mediana</b>	3875.8	inf	146.77	inf	inf	inf	inf	inf
	<b>Des. Est.</b>	963.4	inf	23.37	inf	inf	inf	inf	inf
<b>10x10<sup>3</sup></b>	<b>Mejor</b>	13.09	inf	66.24	inf	inf	inf	inf	inf
	<b>Peor</b>	579.09	inf	99.96	inf	inf	inf	inf	inf
	<b>Media</b>	163.93	inf	82.18	inf	inf	inf	inf	inf
	<b>Mediana</b>	132.38	inf	81.05	inf	inf	inf	inf	inf
	<b>Des. Est.</b>	122.05	inf	8.62	inf	inf	inf	inf	inf
<b>20x10<sup>3</sup></b>	<b>Mejor</b>	8.60	inf	53.82	inf	inf	inf	inf	inf
	<b>Peor</b>	220.06	inf	99.00	inf	inf	inf	inf	inf
	<b>Media</b>	75.88	inf	73.61	inf	inf	inf	inf	inf
	<b>Mediana</b>	60.97	inf	73.61	inf	inf	inf	inf	inf
	<b>Des. Est.</b>	61.06	inf	9.46	inf	inf	inf	inf	inf

Como es evidente, el algoritmo propuesto no es competitivo para los COPs del benchmark. Estos resultados son preliminares y están limitados en dos sentidos importantes: primero, en el hecho de que los parámetros no han sido modificados, y segundo, en el hecho de que los límites de evaluaciones en la función objetivo son más restrictivos que en el benchmark. No obstante, el algoritmo muestra una debilidad muy fuerte en problemas con restricciones de igualdad como el C03, C06, C07 y C11, y en problemas con restricciones de desigualdad más complejos como el C13 y C19.

<sup>2</sup>Además excluimos los desplazamientos del problema.

Cuadro 2: Resultados con dimensión = 30

FES	Estadística	C01	C03	C04	C06	C07	C11	C13	C19
<b>2x10<sup>3</sup></b>	<b>Mejor</b>	14148.54	inf	403.90	inf	inf	inf	inf	inf
	<b>Peor</b>	36858.93	inf	567.68	inf	inf	inf	inf	inf
	<b>Media</b>	24310.30	inf	493.71	inf	inf	inf	inf	inf
	<b>Mediana</b>	24310.30	inf	499.35	inf	inf	inf	inf	inf
	<b>Des. Est.</b>	5271.50	inf	34.58	inf	inf	inf	inf	inf
<b>10x10<sup>3</sup></b>	<b>Mejor</b>	640.94	inf	275.07	inf	inf	inf	inf	inf
	<b>Peor</b>	2580.97	inf	366.10	inf	inf	inf	inf	inf
	<b>Media</b>	1501.79	inf	327.44	inf	inf	inf	inf	inf
	<b>Mediana</b>	1313.43	inf	334.75	inf	inf	inf	inf	inf
	<b>Des. Est.</b>	622.95	inf	23.64	inf	inf	inf	inf	inf
<b>20x10<sup>3</sup></b>	<b>Mejor</b>	454.26	365255.92	269.90	inf	inf	inf	inf	inf
	<b>Peor</b>	2174.91	inf	347.04	inf	inf	inf	inf	inf
	<b>Media</b>	962.57	inf	314.55	inf	inf	inf	inf	inf
	<b>Mediana</b>	840.84	inf	327.52	inf	inf	inf	inf	inf
	<b>Des. Est.</b>	397.25	inf	24.18	inf	inf	inf	inf	inf

Esto se explica de manera intuitiva porque el algoritmo considerado inicializa las posiciones sobre el espacio de soluciones de manera uniforme y posteriormente las partículas siguen un movimiento aleatorio si ninguna de ellas es factible. Para problemas con restricciones de igualdad y restricciones de desigualdad con conjuntos factibles *muy pequeños*.

Una posible solución a este problema sería modificar el algoritmo de tal manera que primero se busquen las regiones factibles mediante la minimización del valor absoluto de las restricciones, tanto de igualdad como de desigualdad. Este proceso podría realizarse hasta tener al menos una posición factible dentro del cúmulo, y, a partir de ahí, seguir con el planteamiento original del algoritmo. Potencialmente esto podría tener un impacto fuerte y negativo sobre el costo computacional del algoritmo pero se conjetura que podría mejorar considerablemente el desempeño del algoritmo en problemas complejos.

Por último, para el próximo borrador se hará una búsqueda de mejores parámetros para los problemas planteados en el benchmark.

## Hipótesis y Objetivos

*Hipótesis:* El algoritmo planteado es competitivo en términos del benchmark definido en [WMS16].

*Objetivo:* Diseñar e implementar el algoritmo y mostrar mediante las variables de estudio y los métodos de evaluación planteadas el desempeño de tal.

## Referencias

- [BK18] T. Blackwell and J. Kennedy. Impact of communication topology in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2018.
- [BLM13] Mohammadreza Bonyadi, Xiang Li, and Zbigniew Michalewicz. A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 1–8, New York, NY, USA, 2013. ACM.
- [BLM14] Mohammad Reza Bonyadi, Xiang Li, and Zbigniew Michalewicz. A hybrid particle swarm with a time-adaptive topology for constrained optimization. *Swarm and Evolutionary Computation*, 18:22 – 37, 2014.
- [BM14] Mohammad reza Bonyadi and Zbigniew Michalewicz. *Locating Potentially Disjoint Feasible Regions of a Search Space with a Particle Swarm Optimizer*, pages 205–230. 12 2014.
- [BM17] Mohammad Reza Bonyadi and Zbigniew Michalewicz. Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation*, 25(1):1–54, 2017. PMID: 26953883.
- [CH03] G. Coath and S. K. Halgamuge. A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 4, pages 2419–2425 Vol.4, Dec 2003.
- [Cle06] M. Clerc. Standard pso 2006. Technical report, 2006.
- [Eng13] A. P. Engelbrecht. Particle swarm optimization: Global best or local best? In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pages 124–135, Sep. 2013.
- [ESMM13] Saber Elsayed, Ruhul Sarker, and Efrén Mezura-Montes. Particle swarm optimizer for constrained optimization. 06 2013.
- [GZ13] Yue-Jiao Gong and Jun Zhang. Small-world particle swarm optimization with topology adaptation. *GECCO 2013 - Proceedings of the 2013 Genetic and Evolutionary Computation Conference*, 07 2013.
- [HE02] Xiaohui Hu and R Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization. *Citeseer*, 2002:203–206, 01 2002.

- [HW07] Qie He and Ling Wang. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1):89–99, 2 2007.
- [JM05] S. Janson and M. Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1272–1282, Dec 2005.
- [KE95] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
- [Ken99] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1931–1938 Vol. 3, July 1999.
- [KM02] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No.02TH8600)*, volume 2, pages 1671–1676 vol.2, May 2002.
- [LI14] Wei Hong Lim and Nor Ashidi Mat Isa. Particle swarm optimization with increasing topology connectivity. *Engineering Applications of Artificial Intelligence*, 27:80 – 102, 2014.
- [MKN04] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, June 2004.
- [MM13] Yannis Marinakis and Magdalene Marinaki. A hybridized particle swarm optimization with expanding neighborhood topology for the feature selection problem. In María J. Blesa, Christian Blum, Paola Festa, Andrea Roli, and Michael Sampels, editors, *Hybrid Metaheuristics*, pages 37–51, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [PC04] G. T. Pulido and C. A. C. Coello. A constraint-handling mechanism for particle swarm optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 2, pages 1396–1403 Vol.2, June 2004.
- [PE07] Ulrich Paquet and Andries P. Engelbrecht. Particle swarms for linearly constrained optimisation. *Fundam. Inf.*, 76(1-2):147–170, January 2007.
- [PV02] Konstantinos E. Parsopoulos and Michael N. Vrahatis. Particle swarm optimization method for constrained optimization problems.

In *In Proceedings of the Euro-International Symposium on Computational Intelligence 2002*, pages 214–220. IOS Press, 2002.

- [SMF19] Ricard Solé, Melanie Moses, and Stephanie Forrest. Liquid brains, solid brains. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1774):20190040, 2019.
- [SR98] Yuhui Shi and Eberhart RC. A modified particle swarm optimizer. volume 6, pages 69 – 73, 06 1998.
- [Sug99] P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1958–1962 Vol. 3, July 1999.
- [VEMF19] William F. Vining, Fernando Esponda, Melanie E. Moses, and Stephanie Forrest. How does mobility help distributed systems compute? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1774):20180375, 2019.
- [WMS16] Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Suganthan. Problem definitions and evaluation criteria for the cec 2017 competition and special session on constrained single objective real-parameter optimization. 10 2016.
- [ZY11] Chenggong Zhang and Zhang Yi. Scale-free fully informed particle swarm optimization algorithm. *Inf. Sci.*, 181:4550–4568, 10 2011.