Structural Limiting Range of Perception in Particle Swarm Optimization

Mateusz Mastalerczyk[1][0009−0001−1273−0065], Malgorzata Zajecka[1][0009−0009−7914−2653], Sylwia Bielaszek[0000−0001−8947−6272], Marek Kisiel-Dorohinicki[1][0000−0002−8459−1877], and Aleksander Byrski[1][0000−0001−6317−7012]

AGH University of Krakow, Al. Mickiewicza 30, 30-059 Krakow, Poland
{mastalerczyk,mzajecka,olekb}@agh.edu.pl

Abstract. This research explores the impact of structural limiting perception on the performance of Particle Swarm Optimization by restricting the range of information sharing among particles. By introducing localized communication models through Ring and Tree topologies, the study demonstrates significant improvements over the standard global-best PSO, particularly on a range of Traveling Salesman Problem instances from the TSPLIB. The results show that constraining particle perception enhances both solution quality and convergence behavior, with the Tree topology emerging as the most effective structure. The topological modifications maintain swarm diversity, prevent premature convergence, and facilitate continuous exploration while exploiting promising search regions. These findings suggest that structural constraints on information sharing can enhance PSO's robustness and effectiveness without adding computational complexity, offering a flexible approach applicable to various PSO variants and problem domains beyond TSP.

Keywords: particle swarm optimization, diversity, communication

1 Introduction

Particle Swarm Optimization (PSO) has emerged as one of the most effective population-based metaheuristic algorithms for solving optimization problems. Inspired by the collective behavior of social organisms such as flocks of birds or schools of fish, PSO relies on the dynamic exchange of information among individuals (particles) within a search space. In the standard PSO framework, each particle adjusts its velocity based on both its personal best experience (pbest) and the experience of the best-performing individual in the swarm (gbest). There are many modifications of the basic PSO algorithm, making the swarm intelligence actually a broad class of computing methods (see, e.g. [1,16]).

Although this global information-sharing mechanism effectively guides the swarm toward optimal solutions, it can also lead to premature convergence and stagnation, especially in complex and high-dimensional search spaces. This occurs because all particles are influenced by the same global best solution, which

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

can limit the diversity of the search and cause the algorithm to become trapped in local optima.

A critical challenge in PSO design is balancing exploration and exploitation. The classic PSO approach, which uses an unrestricted global best (gbest) topology, often favors exploitation. While this accelerates convergence, it can result in suboptimal performance in multimodal and dynamically changing environments. One potential solution is to modify the interaction topology to limit the range of perception among particles. By constraining how particles share information, alternative topologies can promote a more diverse search and prevent premature convergence.

In this paper, we introduce a novel structural approach to limiting the range of perception in PSO by modifying its topology. Instead of relying on a global best (gbest) selection that considers the entire swarm, we implement localized communication models where each particle is influenced only by a subset of the swarm. This localized structure allows particles to form distinct neighborhoods, facilitating the exploration of multiple local optima rather than converging to a single global solution. By adjusting the topology, we can fine-tune the balance between exploration and exploitation, thereby enhancing the performance of PSO on a variety of complex optimization tasks.

Our proposed modifications to the PSO topology redefine how information propagates across the swarm, potentially improving robustness, adaptability, and scalability. Unlike traditional PSO, where each particle has full awareness of the swarm's best solution, our approach ensures that particles only have access to a constrained set of neighbors. This creates a more diverse evolutionary pressure that mitigates premature convergence while maintaining the algorithm's efficiency. Although our experiments focus on classic PSO, the proposed strategy for limiting the range of perception can be seamlessly extended to any PSO variant, making it a versatile tool for different problem domains.

In the next sections we show selected PSO-inspired algorithms using similar approach (limiting the perception of the swarm individuals). In relation to those referenced papers we propose our idea, then test it comparing the original PSO with the ones using novel technologies. It is to note, that the proposed communication structures may be applied far beyond the original PSO, which makes the contribution much more universal than only modifying a selected algorithm.

2 Perception in PSO

Metaheuristics are an excellent tool for solving difficult optimization problems. Despite perfection, there are difficulties, such as excessive convergence, which will lead to inaccurate solving problems and uncertainties of results. Metaheuristics have some possibilities to restore population diversity. If there is an effect of population uniformity in parallelized version of the evolutionary algorithm (PEA), then a good solution is to divide the population into several subpopulations. PEA restores population diversity by migrating individuals between islands [19].

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

Additionally, there is a constant pressure to accelerate metaheuristics work. That is why metaheuristics are modified and combined with other techniques. [22] presents a novel two-stage hybrid swarm intelligence optimization algorithm that combines the evolution ideas of the genetic algorithms, particle swarm optimization and ant colony optimization based on the compensation for solving the traveling salesman problem.

Particle Swarm Optimization (PSO) was first introduced by J. Kennedy et al. in 1995 as a nature-inspired optimization technique, drawing inspiration from swarm intelligence observed in biological phenomena such as bird flocking, fish schooling, and human social behavior [12]. The algorithm's foundation is closely related to evolutionary computation, and over time, PSO has emerged as a strong alternative to traditional genetic algorithms and other iterative optimization methods [5,3].

PSO has demonstrated robust performance across a diverse range of applications [4,20,15]. The algorithm begins by initializing a population of candidate solutions, referred to as particles, with randomized velocities. These particles navigate the search space iteratively, adjusting their positions based on their own best-discovered solution and the best-known solution found by the entire swarm. This mechanism enables PSO to efficiently explore and exploit the solution space, making it a widely used optimization technique in computational science.

The approach to using PSO [17] in problem solving has also been modified many times since its inception [6,2]. A crucial factor influencing PSO's performance is the gbest topology, where all particles are attracted to the best-performing individual in the swarm. While this accelerates convergence, it can also lead to stagnation if gbest becomes dominant too early, reducing search diversity.

Several studies have introduced modifications to the algorithm, its operators, or the underlying mechanisms to alter how information is exchanged among particles:

1. Velocity Operator Adjustments [7], customized velocity operators were introduced to refine how particles update their positions, influencing their movement dynamics based on their neighborhood structure.
2. Hierarchical and Hybrid Strategies[18], a two-level search strategy combined with a crossover elimination technique was used to scale PSO for solving larger optimization problems.
3. Clustered PSO [21], a two-phase heuristic algorithm was tested for solving the Capacitated Vehicle Routing Problem (CVRP), integrating clustering techniques (k-means, k-medians, k-medoids) to improve route calculations using PSO, Genetic Algorithm (GA), and Ant Colony Optimization (ACO). Interestingly, PSO performed competitively between a winning GA and a losing ACO, suggesting its effectiveness in structured search spaces.
4. Swarm Initialization and Reset Mechanisms [10], an improved swarm initialization method was proposed to periodically reset the swarm greedily and introduce cross-transformation operations, thereby mitigating stagnation.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

5. PSO for Combinatorial Optimization [9], a discrete PSO variant was analyzed for solving the Traveling Salesman Problem (TSP). This study introduced a novel method of measuring differences between tours using "edge exchanges" and computing a centroid of these differences. Notably, this approach eliminated inertia and relied solely on attraction to local and global best solutions.
6. Levy Flight for Enhanced Exploration [11], a PSO variant with Levy flight was introduced to improve population diversity. Levy flight, a type of stochastic random walk with heavy-tailed step sizes, was applied to a subset of particles (excluding gbest) to enhance exploration.
7. Lbest approach [14] enables particles to explore different regions independently, allowing the swarm to "flow around" local optima rather than becoming trapped.

Enhancing PSO through modifications remains a crucial research direction, particularly with the integration of modern AI techniques. Recent studies have demonstrated the effectiveness of combining metaheuristics with machine learning (ML) [8,13] to improve adaptability and performance. However, a significant gap remains in optimizing the influence of the global best (gbest) solution and its distribution across the swarm. Addressing this limitation through adaptive or structured information-sharing mechanisms could further enhance PSO's ability to balance exploration and exploitation, leading to more robust and scalable optimization methods

3 Structural limiting perception in PSO

3.1 Classical Particle Swarm Optimization for discrete problems

Particle Swarm Optimization model for discrete problems, in particular the Traveling Salesman Problem, is a quintuple $PSO = (T, X, \mathcal{S}, obj, \chi)$, where:

- $T = \{t_j\}_{j\in\mathbb{N}}$ - countable time space; when regarding elements of T we will denote $t_0 =: 0$, $t_1 =: 1$ etc.
- X - feasible position space: a discrete set of permutations $X \subset \mathfrak{S}_n$, with a common starting element. Feasible position space can be identified with a subset of $\mathbb{R}^n$ by an injective mapping $\mathfrak{S}_n \to \mathbb{R}^n$ coding each element in permutation as a real number.
- obj - objective function $obj : X \to \mathbb{R}$; the aim of optimization task is to minimize the objective function
- $\chi = \{\chi_t\}_{t\in T}$ - a family of time-indexed probability distributions
- $\mathcal{S}$ - a swarm with update rules, i.e. a quadruple $\mathcal{S} = (S, S_0, v, x)$ where S is a particle swarm of $\#I = N$ particles, S_0 is a set of initial states of particles, $v : I \times T \to X$ is a velocity update function and $x : I \times T \to X$ is a position update function.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

A particle swarm S is a function of time returning $S(t)$ - the *state* of the swarm at time t

$$S : T \ni t \mapsto S(t) = \big(\{s_i(t)\}_{i\in I}, gbest(t)\big) \in (X^3)^I \times X,$$

where I is an at least countable set of indices, $\{s_i(t)\}_{i\in I}$ is a set of particles at time t, and $gbest(t)$ is a global best at time t (see definitions below).
A particle $s_i : T \to X^3$, $i \in I$, is defined by its position $x_i = x(i,\cdot)$, velocity $v_i = v(i,\cdot)$, obtained from update functions, and personal best $pbest_i$, namely

$$s_i(t) = \big(x_i(t), v_i(t), pbest_i(t)\big) \in X^3.$$

Position $x_i(t)$ of particle s_i at time t is a single permutation from the position space X and its velocity $v_i(t)$ at time t is a vector of probabilities of each transposition $(j\ k)$, i.e. switching j-th with k-th element in a permutation.
Personal best $pbest_i(t) : T \to X$ of each particle s_i, $i \in I$ is the best position achieved by particle s_i up to time t, i.e.

$$pbest_i(t) = \underset{\{x_i(\tau):\tau\in T,\tau\le t\}}{\arg\min} \big(obj(x_i(\tau))\big).$$

Global best $gbest(t) : T \to X$ is the best position achieved by all particles in the swarm up to time t, i.e.

$$gbest(t) = \underset{\{x_i(\tau):\tau\in T,\tau\le t, i\in I\}}{\arg\min} \big(obj(x_i(\tau))\big).$$

The initial state of the particle s_i is a tuple $(x_i^0, v_i^0) \in X \times X$ setting its position and velocity in time zero. *The set* S_0 *of initial states* is $S_0 = \{(x_i^0, v_i^0)\}_{i\in I}$.
Velocity update function $v : I \times T \to X$ assigns i-th particle at time t the probability vector $v_i(t) = (\rho_{(j\ k)})_{\substack{j,k=1,\dots,n\\ j<k}}$ where $\rho_{(j\ k)}$ is a probability of transposition $(j\ k)$. The discrete velocity update function should mimic the behavior of the classical PSO velocity $v(i,t+1) = \omega v_i(t) + r_1 c_p\big(pbest_i(t) - x_i(t)\big) + r_2 c_g\big(gbest(t) - x_i(t)\big)$.
Thus one of possible approaches to define the discrete velocity update function is:

$$\begin{cases} v(i,0) = \quad v_i(0) \quad\ = v_i^0 \\ v(i,t+1) = v_i(t+1) = \lfloor \omega v_i(t) + r_1 c_p P_p(i,t) + r_2 c_g P_g(i,t) \rceil_{[0,1]} \end{cases}$$

where $\omega \in [0,1]$ is an inertion constant, $c_p, c_g \in [0,1]$ are personal and global accelerator constants independent of i and t, and r_1, r_2 are stochastic factors sampled from distribution $\chi(t)$. Transposition occurence vector $P_p(i,t) = (oc_{(j\ k)})_{\substack{j,k=1,\dots,n\\ j<k}}$ (respectively, $P_g(i,t)$) represents transpositions transforming $x_i(t)$ to position $pbest_i(t)$ (respectively, $gbest(t)$). Namely, let $\Theta_p(i,t)$ ($\Theta_g(i,t)$) be a combination of transpositions transforming position

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

$x_i(t)$ to position $pbest_i(t)$ ($gbest(t)$). If $(j\ k) \in \Theta_p(i,t)$ ($\Theta_g(i,t)$), then in $P_p(i,t)$ ($P_g(i,t)$) occurence $oc_{(j\ k)} = 1$, otherwise $oc_{(j\ k)} = -1$. The rounding $\lfloor\cdot\rceil_{[0,1]}$ rounds vector terms greater than 1 down to 1 and terms smaller than 0 up to 0.

Position update function $x : I \times T \to X$ changes current position $x_i(t)$ to position $x_i(t+1)$ in time $t+1$. It is defined recursively:

$$\begin{cases} x(i,0) = x_i(0) & = x_i^0 \\ x(i,t+1) = x_i(t+1) & = x_i(t) + v_i(t+1), \end{cases}$$

where $+v_i(t+1)$ means performing on permutation $x_i(t)$ transpositions based on their probability given by velocity $v_i(t+1)$.

3.2 Structural limiting perception in PSO

Our structural approach to limit the range of perception in PSO is based on modification of its topology where the i-th particle can interact only with a subset S_i of $\#I_i$ particles of the swarm. Thus, the particle does not have access to $gbest(t)$, the best position achieved by all particles in the swarm. We need to define a new, indexed function $gbest : I \times T \to X$ - the best position achieved up to time t by particles accessible by i-th particle, i.e.

$$gbest(i,t) = gbest_i(t) = \underset{\{x_i(\tau):\tau\in T_i, \tau\le t, i\in I_i\}}{\arg\min} \big(obj(x_i(\tau))\big).$$

The subset S_i of particles accessible by the i th particle depends on the topology of the entire particle swarm and the chosen radius r.

Ring topology In *Ring topology*, all N particles are arranged in a form of a ring (see Figure 1). For the radius $r < N/2$, the *neighborhood of radius r of a particle* s_i is defined simply as a $2r+1$ element subset $S_i = \{s_{i-r}, ..., s_i, ..., s_{i+r}\}$, where for non-positive indices we define $s_j := s_{N-|j|}$, $j = 0, -1, -2, ...$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

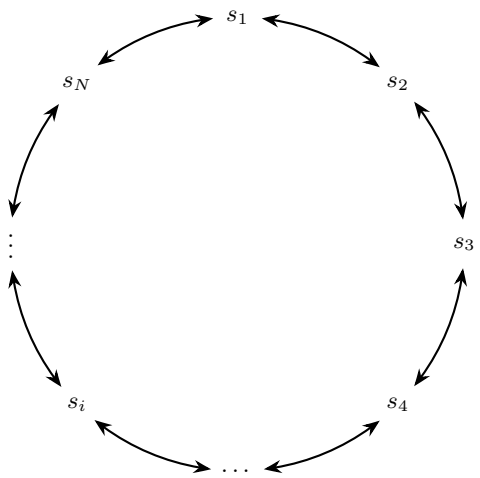

Fig. 1: Ring topology. The neighborhood of particle s_2 of radius $r = 2$ is $S_2 = \{s_N, s_1, s_2, s_3, s_4\}$.

Tree topology In the *Tree topology* the particles are arranged as nodes of a complete binary tree (see Figure 2). The neighborhood of node s_i of radius r is the set S_i of all nodes of the tree accessible from s_i by moving along at most r edges. The size of the particular set S_i depends on the placement of the particle s_i in the tree.

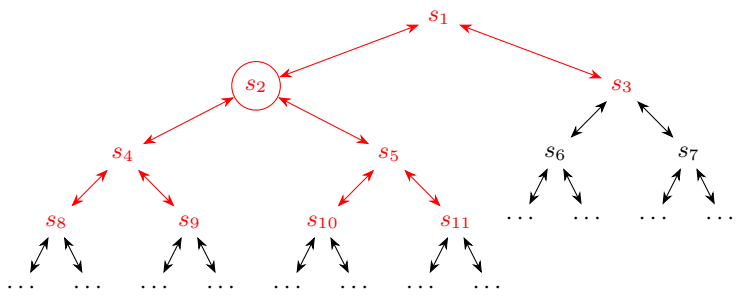

Fig. 2: Tree topology. The neighborhood of particle s_2 of radius $r = 2$ is shown in red.

4 Experimental results

The computing node utilized on the Ares supercomputer, provided by the Academic Computing Center "Cyfronet", is a robust and highly capable system. It

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

boasts an `x86_64` CPU architecture with 8 cores per socket, totaling 32 CPUs overall.

The Traveling Salesman Problem Library (TSPLIB)[1] , maintained by the University of Heidelberg, provides benchmark instances essential for TSP algorithm research and standardization. The collection ranges from small problems to large-scale instances with thousands of nodes, incorporating both real-world geographic data and synthetic test cases. Each dataset includes documentation of its origin, node count, and characteristics (symmetry, distance metrics), enabling consistent evaluation and comparative analysis across the optimization research community.

In our experiments, neighborhood configuration is governed by two primary parameters: the neighborhood radius and the neighborhood topology. The radius defines the spatial extent of each particle's neighborhood, determining how far-reaching each particle's interactions are within the network. The topology parameter specifies the structural arrangement of connections within this radius, shaping how nodes are organized and interact with one another. Together, these parameters control the local connectivity and structural dynamics of the neighborhood, providing a foundation for analyzing their combined impact on algorithmic performance and network behavior.

When comparing the influence of the radius of the neighborhood in different topologies (see Figure 3), we observed distinct behaviors in the Ring and Tree topologies. In the Ring topology, a significant improvement in performance was noted when the neighborhood radius was increased beyond a value of 17, with the algorithm consistently achieving better results as the radius expanded. Figure 3 illustrates this relationship through box plots, where each box represents the distribution of solution qualities across multiple runs for a specific radius value. The whiskers extend to show the full range of variation (minimum to maximum values), while the boxes indicate the interquartile range with the median marked as a horizontal line. This suggests that larger neighborhoods in the Ring topology enhance the information-sharing process, leading to more effective convergence. In contrast, for the Tree topology, we did not observe a similar drastic improvement in performance with changes to the radius. The Tree topology appeared to be less sensitive to radius adjustments, indicating that its hierarchical structure may already provide an optimal level of neighborhood connectivity, regardless of the radius size.

Comparing the performance of PSO using different neighborhood topologies across 27 experiments (see Table 1 and Fig. 4,5), we found that the Tree and Ring topologies produced varied outcomes. The Tree topology yielded the best results in 14 of the experiments, showing a strong performance in guiding the swarm toward optimal solutions. Meanwhile, the Ring topology was the top performer in 12 experiments (see the best fitness and the final results observed for all the problems tackled in Figs. 4, 5 tracking the convergence of the cost function across iterations for each approach, clearly illustrating how the optimization trajectories differ between topologies and displays box plots that capture the statistical

[1] `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

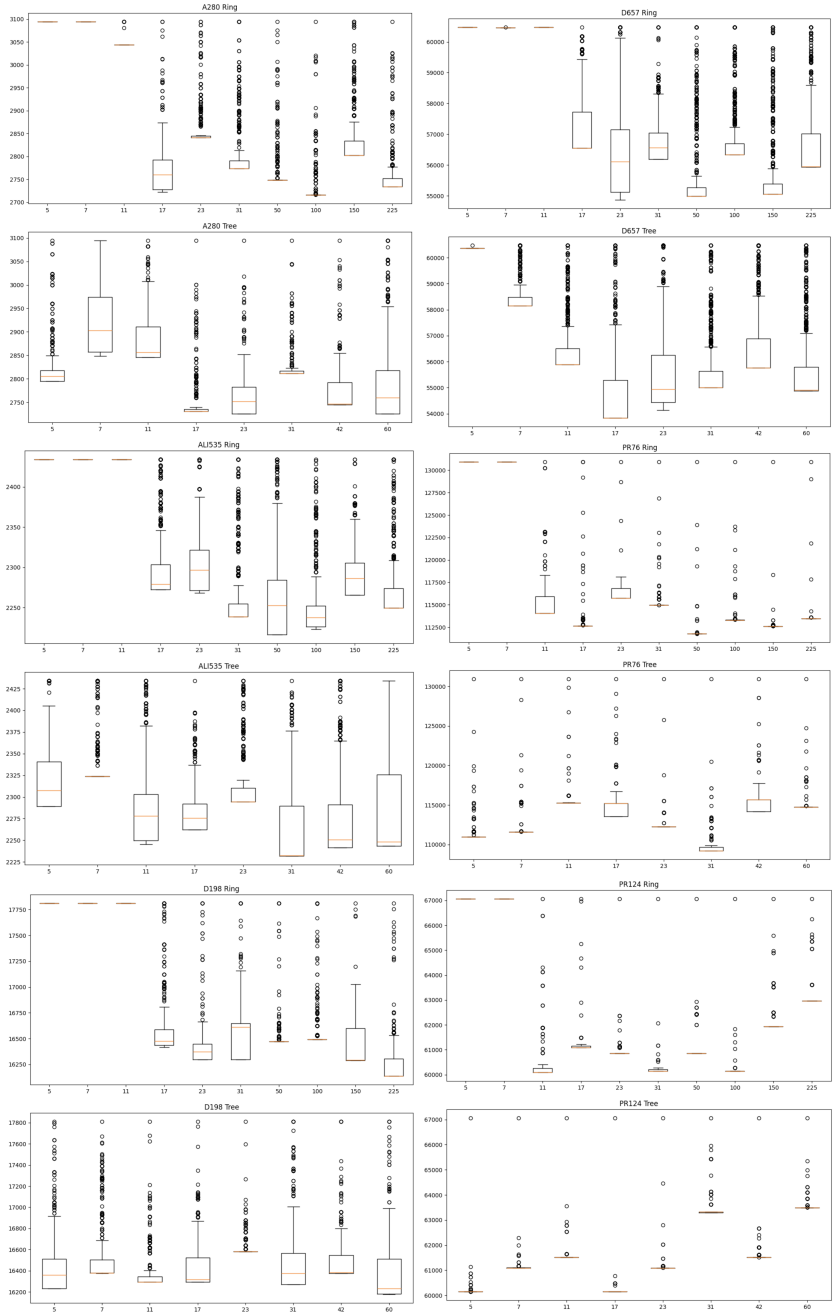

Fig. 3: Comparison of neighbourhood radius and distribution of cost function values in solutions for given topology.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

Table 1: Influence of different topologies on the efficacy of the algorithm

	Base	**Ring**	**% Impr.**	**Tree**	**% Impr.**
A280	2907.43	**2715.67**	6.60%	2725.28	6.27%
ALI535	2422.43	**2216.21**	8.51%	2231.77	7.87%
BERLIN52	7544.37	7544.37	0.00%	7544.37	0.00%
BIER127	125526.11	120534.37	3.98%	**119908.43**	4.48%
D198	16920.81	**16137.03**	4.63%	16176.32	4.40%
D493	39046.98	37573.93	3.77%	**36979.43**	5.30%
D657	59797.70	54862.37	8.25%	**53830.35**	9.98%
D1291	59039.96	**56799.90**	3.80%	56865.50	3.68%
D1655	74257.31	72099.90	2.91%	**71592.28**	3.59%
PR76	114168.38	111758.33	2.11%	**109187.20**	4.36%
PR124	61879.51	**60088.84**	2.89%	60144.91	2.80%
PR136	108121.22	**102704.27**	5.01%	103014.32	4.72%
PR152	76591.02	**74570.78**	2.64%	75394.09	1.56%
PR226	85932.47	81521.89	5.13%	**81294.46**	5.40%
PR264	54246.47	51648.98	4.79%	**51444.39**	5.17%
PR299	55269.41	51183.28	7.39%	**51020.27**	7.69%
PR439	123136.92	**115400.68**	6.28%	116630.74	5.28%
PR1002	308687.76	**288016.75**	6.70%	288845.45	6.43%
TSP225	4317.02	**4050.63**	6.17%	4097.47	5.09%
U159	47417.00	**44676.49**	5.78%	45222.16	4.63%
U574	44278.99	41420.18	6.46%	**40867.74**	7.70%
U724	50221.52	47510.58	5.40%	**46023.45**	8.36%
U1060	273657.28	254473.71	7.01%	**251944.42**	7.93%
U1432	185220.63	176764.05	4.57%	**173481.57**	6.34%
U1817	67459.73	**66432.23**	1.52%	66455.14	1.49%
U2152	77072.68	76696.36	0.49%	**75997.36**	1.40%
U2319	270048.52	270293.90	-0.09%	**269616.63**	0.16%

distribution of solution qualities), demonstrating its effectiveness in maintaining a balanced exploration across particles. These findings suggest that, while both topologies have strengths, the Tree topology may offer a slight advantage in achieving optimal results more consistently under the tested conditions.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

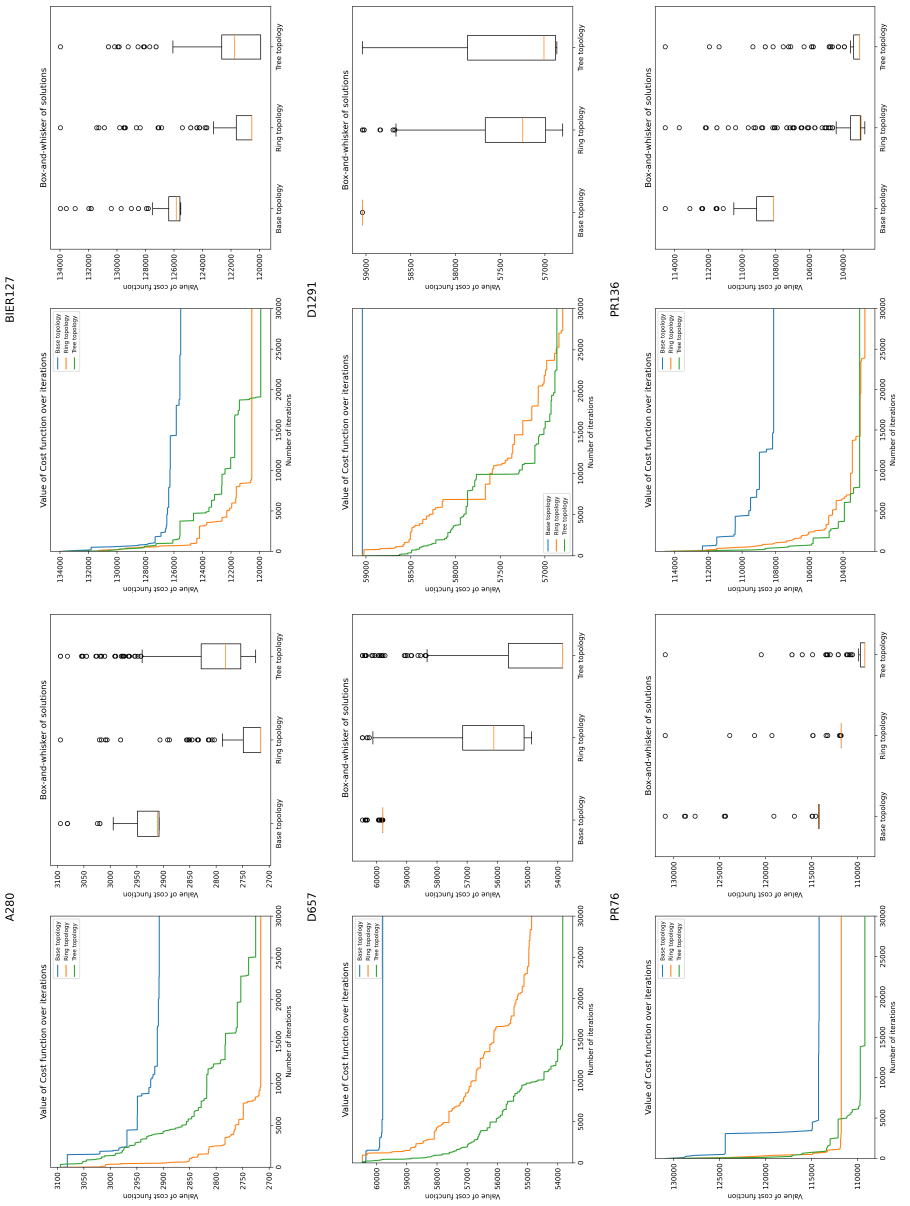

Fig. 4: Comparing efficacy of topologies for different problems tackled

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

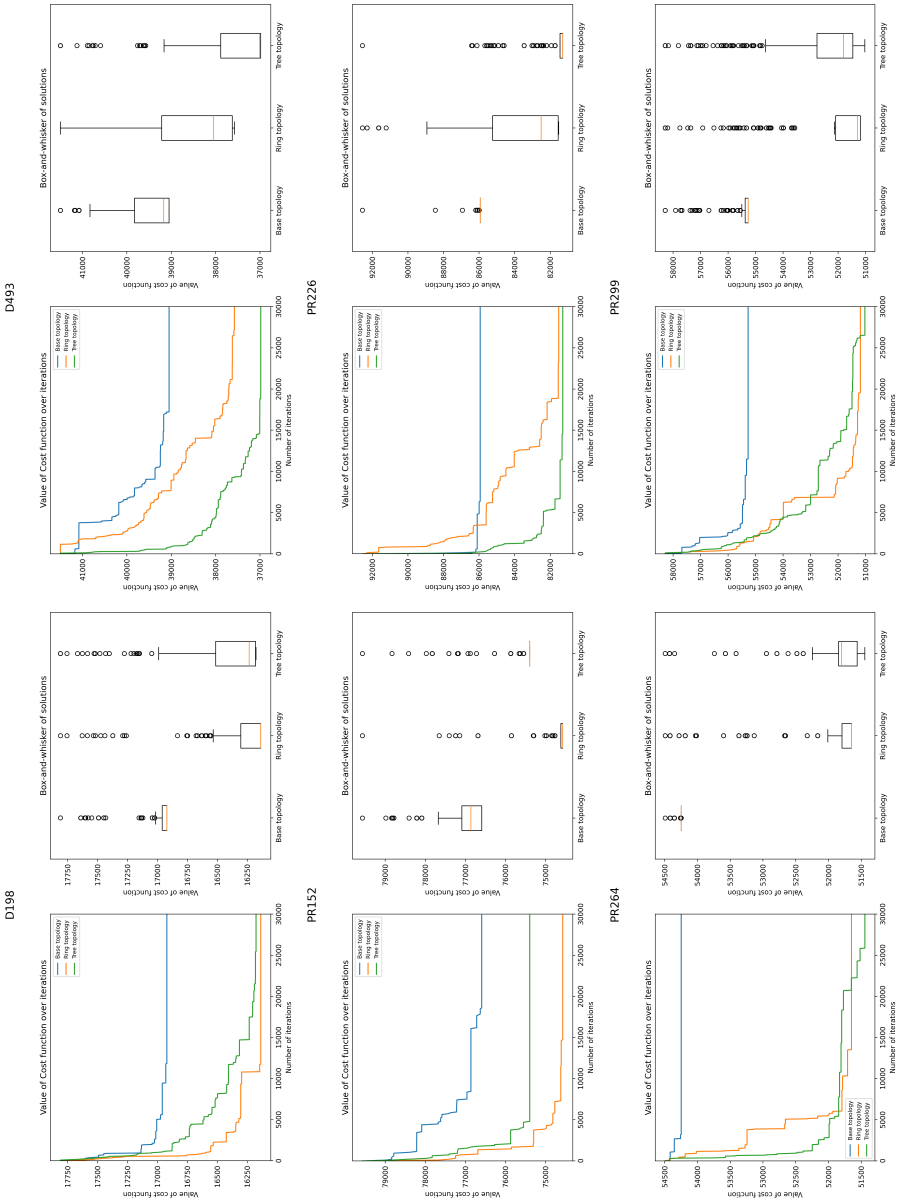

Fig. 5: Comparing efficacy of topologies for different problems tackled (cont.)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

In addition to the improvements in solution quality, both the Tree and Ring topologies demonstrated faster convergence to better results compared to the Base PSO approach. While Base PSO often became stuck in local optima, this tendency was more pronounced as the problem size increased, with the algorithm quickly losing diversity and getting trapped in suboptimal solutions. In contrast, the use of neighborhood topologies, whether Tree or Ring, mitigated this issue, maintaining a more balanced exploration of the solution space. These topologies enabled the swarm to converge more efficiently and avoid getting stuck in local optima as frequently, even in larger and more complex problems, highlighting their robustness in comparison to the Base approach.

5 Conclusions

Our research on structural limiting perception in Particle Swarm Optimization (PSO) demonstrates that restricting the range of information sharing among particles can significantly enhance algorithm performance on complex optimization problems. By implementing localized communication models through Ring and Tree topologies, we observed consistent improvements over the standard global-best PSO across a diverse set of TSP instances from the TSPLIB.

The experimental results clearly indicate that constraining particle perception brings substantial benefits in terms of both solution quality and convergence behavior. The Tree topology emerged as the most effective structure. Both topologies consistently outperformed the standard PSO implementation, with improvements becoming more pronounced as the complexity of the problem increased.

A key finding of our work is that the neighborhood radius plays a topology-dependent role in algorithm performance. In the Ring topology, increasing the radius of the neighborhood beyond 17 led to significant improvements, suggesting that expanded local information sharing improves convergence in this structure. Conversely, the Tree topology showed less sensitivity to radius adjustments, indicating that its hierarchical structure may inherently provide effective information dissemination regardless of neighborhood size.

The improved performance of both proposed topologies can be attributed to their ability to maintain swarm diversity and prevent premature convergence. Although standard PSO frequently became trapped in local optima, especially for larger problem instances, our topological modifications facilitated continued exploration of the search space while still allowing effective exploitation of promising regions.

These findings have important implications for the design and application of PSO. By implementing structural constraints on information sharing, we can significantly enhance PSO's robustness and effectiveness without increasing computational complexity. Furthermore, the approach is flexible enough to be applied to various PSO variants and problem domains beyond the TSP.

While our research focused on TSP instances, our primary goal was to examine whether limiting perception produces better optimization results, rather

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

than developing a TSP-specific solution. The promising outcomes suggest that these topological modifications could be readily translated to other PSO variants and problem domains. Future research should explore implementing these perception-limiting structures in different PSO algorithms applied to varied optimization challenges, from continuous function optimization to dynamic problems, to further validate the generalizability of our approach.

Acknowledgements

The research presented in this paper has been financially supported by: Polish National Science Center Grant no. 2019/35/O/ST6/00571 "Parallelization of metaheuristics with desynchronization." (SB); NAWA - Polish National Agency for Academic Exchange - Program: NAWA Preludium BIS 1 (SB); Polish Ministry of Science and Higher Education funds assigned to AGH University of Science and Technology (AB,MZ). We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing HPC facilities. ARTIQ project – Polish National Science Center:DEC-2021/01/2/ST6/00004, Polish National Center for Research and Development, DWP/ARTIQI/426/2023 (MKD)

References

1. Adrdor, R.: The power of intelligence emerging from swarms. Computer Science **26**(1) (Apr 2025). https://doi.org/10.7494/csci.2025.26.1.6306, https://journals.agh.edu.pl/csci/article/view/6306
2. Akhand, M.A.H., Rahman, M.M., Siddique, N.: Advances on Particle Swarm Optimization in Solving Discrete Optimization Problems, pp. 59–88 (10 2022). https://doi.org/10.1007/978-3-031-09835-2_4
3. Boeringer, D., Werner, D.: Particle swarm optimization versus genetic algorithms for phased array synthesis. IEEE Transactions on Antennas and Propagation **52**(3), 771–779 (2004). https://doi.org/10.1109/TAP.2004.825102
4. Chang, B.C., Ratnaweera, A., Halgamuge, S.K., Watson, H.C.: Particle swarm optimisation for protein motif discovery. Genetic Programming and Evolvable Machines **5**(2), 203–214 (Jun 2004). https://doi.org/10.1023/B:GENP.0000023688.42515.92, https://doi.org/10.1023/B:GENP.0000023688.42515.92
5. Eberhart, R.C., Shi, Y.: Comparison between genetic algorithms and particle swarm optimization. In: Porto, V.W., Saravanan, N., Waagen, D., Eiben, A.E. (eds.) Evolutionary Programming VII. pp. 611–616. Springer Berlin Heidelberg (1998)
6. Gad, A.G.: Particle swarm optimization algorithm and its applications: A systematic review. Archives of Computational Methods in Engineering **29**, 2531 – 2561 (2022), https://api.semanticscholar.org/CorpusID:248274842
7. Goldbarg, E., Goldbarg, M., Souza, G.: Particle Swarm Optimization Algorithm for the Traveling Salesman Problem (09 2008). https://doi.org/10.5772/5580
8. He, C., Zhang, Y., Gong, D., Ji, X.: A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. Expert Systems with Applications **217**, 119495 (May 2023). https://doi.org/10.1016/j.eswa.2022.119495, https://www.sciencedirect.com/science/article/pii/S0957417422025143

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19

9. Hoffmann, M., Mühlenthaler, M., Helwig, S., Wanka, R.: Discrete particle swarm optimization for tsp: Theoretical results and experimental evaluations. In: Bouchachia, A. (ed.) Adaptive and Intelligent Systems. pp. 416–427. Springer Berlin Heidelberg (2011)
10. Jabor, F.K., Omran, G.A., Mhana, A., Gheni, H.M.: Optimization of particle swarms for travelling salesman problem. In: 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). pp. 1–6 (2022). https://doi.org/10.1109/HORA55278.2022.9800086
11. Jana, N.D., Sil, J.: Particle swarm optimization with lévy flight and adaptive polynomial mutation in gbest particle. In: Thampi, S.M., Abraham, A., Pal, S.K., Rodriguez, J.M.C. (eds.) Recent Advances in Intelligent Informatics. pp. 275–282. Springer International Publishing, Cham (2014)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948 vol.4 (1995). https://doi.org/10.1109/ICNN.1995.488968
13. Liang, A., Yang, H., Sun, L., Sun, M.: A Three-Layered Multifactorial Evolutionary Algorithm with Parallelization for Large-Scale Engraving Path Planning. Electronics **11**(11), 1712 (Jan 2022). https://doi.org/10.3390/electronics11111712, https://www.mdpi.com/2079-9292/11/11/1712, number: 11 Publisher: Multidisciplinary Digital Publishing Institute
14. Liu, H., Li, B., Ji, Y., Sun, T.: Particle swarm optimisation from lbest to gbest. In: Abraham, A., de Baets, B., Köppen, M., Nickolay, B. (eds.) Applied Soft Computing Technologies: The Challenge of Complexity. pp. 537–545. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
15. Parsopoulos, K., Vrahatis, M.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing **1**(2), 235–306 (2002). https://doi.org/10.1023/A:1016568309421, https://doi.org/10.1023/A:1016568309421
16. Pathak, S., Mani, A., Sharma, M., Chatterjee, A.: Quantum inspired chaotic salp swarm optimization for dynamic optimization. Computer Science **25**(2) (Jul 2024). https://doi.org/10.7494/csci.2024.25.2.5289, https://journals.agh.edu.pl/csci/article/view/5289
17. Pereira, G.: Particle swarm optimization (05 2011)
18. Shi, X., Liang, Y., Lee, H., Lu, C., Wang, Q.: Particle swarm optimization-based algorithms for tsp and generalized tsp. Information Processing Letters **103**(5), 169–176 (2007)
19. Skolicki, Z., De Jong, K.: The importance of a two-level perspective for island model design. pp. 4623–4630 (Oct 2007). https://doi.org/10.1109/CEC.2007.4425078
20. Sousa, T., Silva, A., Neves, A.: A particle swarm data miner. In: Pires, F.M., Abreu, S. (eds.) Progress in Artificial Intelligence. pp. 43–53. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
21. Tagorda, I.P., Elwyn Calata, L., Limjoco, W.J.R., Dizon, C.C.: Development of a vehicle routing system for delivery services. In: 2020 IEEE REGION 10 CONFERENCE (TENCON). pp. 1187–1191 (2020). https://doi.org/10.1109/TENCON50793.2020.9293747
22. Yan, X., Zhang, C.Y., Luo, W., Li, W., Chen, W., Liu, H.: Solve traveling salesman problem using particle swarm optimization algorithm (2012), https://api.semanticscholar.org/CorpusID:12794225

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_19