

Алгоритмы PSO и ACO для задачи TSP

Марин Кирилл

27 ноября 2025 г.

Аннотация

Задача коммивояжера (TSP) - одна из наиболее широко изучаемых NP-сложных задач комбинаторной оптимизации. Задача коммивояжёра имеет важное практическое значение в таких областях, как логистика (оптимизация маршрутов доставки), производство (минимизация времени перенастройки оборудования), телекоммуникации (планирование сетевых соединений) и биоинформатика (анализ геномных последовательностей), где поиск кратчайшего замкнутого пути критически влияет на сокращение издержек и повышение эффективности процессов.

Алгоритм оптимизации роя частиц (PSO) был разработан на основе изучения законов поведения птичьих стай, косяков рыб и человеческих сообществ. Оптимизация муравьиных колоний (ACO) — метод оптимизации, который начал применяться в начале 1990-х годов. Он был вдохновлён поведением реальных муравьиных колоний в процессе поиска пищи. Это поведение используется в искусственных колониях муравьёв для решения задач дискретной и непрерывной оптимизации, а также для важных задач в области телекоммуникаций, таких как маршрутизация и балансировка нагрузки.

В этой работе рассматриваются и сравниваются метаэвристики ACO и PSO применительно к задаче коммивояжера. В рамках работы реализованы пять модификаций ACO: классический Ant System (AS), Max-Min Ant System (MMAS), Elitist Ant System (EAS), AS-RANK и Ant Colony System (ACS). Для PSO реализованы две версии: базовая (скорость представляется последовательностью транспозиций) и гибридная модификация, интегрирующая генетические операторы скрещивания и направленной мутации.

1 Введение

1.1 ACO

Основные идеи ACO:

- **Феромонная матрица.** Каждое ребро графа ассоциируется с величиной феромона, отражающей его привлекательность для муравьев.
- **Эвристическая информация.** На каждое ребро накладывается локальный эвристический вес (обычно обратная длина ребра), который влияет на вероятность выбора перехода.
- **Стochasticкий выбор пути.** На шаге построения решения каждый муравей перемещается по вершинам графа, выбирая следующий узел с вероятностью

$$p_{ij} = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_k \tau_{ik}^\alpha \cdot \eta_{ik}^\beta}, \quad (1)$$

где τ_{ij} — текущее количество феромона на ребре (i, j) , $\eta_{ij} = 1/d_{ij}$ — эвристическая оценка (длина ребра d_{ij}), а α, β — параметры, задающие относительную значимость феромона и эвристической информации.

- **Обновление феромона.** После завершения построения всех путей производится глобальное обновление феромона:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^K \Delta \tau_{ij}^{(k)}, \quad (2)$$

где ρ — коэффициент испарения, а вклад k -ого муравья обычно равен $\Delta\tau_{ij}^{(k)} = 1/L_k$ при прохождении ребра (i, j) в пути длины L_k .

ACO эффективен для решения задач комбинаторной оптимизации, таких как задача коммивояжёра, маршрутизация транспортных средств и задача распределения ресурсов. Ключевыми преимуществами метода являются его распределённость, адаптивность к динамическим изменениям и способность находить хорошие решения даже в сложных ландшафтах целевой функции.

1.2 PSO

Основные принципы PSO заключаются в следующем:

- **Частицы.** Популяция частиц (потенциальных решений) инициализируется случайным образом в пространстве поиска. Каждая частица обладает собственной позицией x_i и скоростью v_i .
- **Обновление скорости.** На каждом шаге скорость частицы обновляется с учётом её лучшей известной позиции p_i и глобальной наилучшей позиции g :

$$v_i(t+1) = \omega \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g - x_i(t)), \quad (3)$$

где ω — инерционный вес, c_1, c_2 — коэффициенты когнитивной и социальной составляющих, r_1, r_2 — случайные значения на $[0, 1]$.

- **Обновление позиции.** Положение частицы изменяется по формуле:

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (4)$$

Частицы «помнят» свою лучшую позицию и обмениваются информацией с другими частицами, стремясь к глобальному или локальному оптимуму.

Преимуществами PSO являются простота реализации, малое число параметров и высокая эффективность на задачах с непрерывным пространством. Метод используется в инженерной оптимизации, машинном обучении, экономических задачах и других прикладных областях.

2 ACO

В рамках нашего исследования мы сравниваем следующие классические модификации алгоритма муравьиной колонии (ACO):

- **AS** (Ant System) - классический алгоритм;
- **EAS** (Elitist Ant System) - Внедрена концепция «элитности», которая подразумевает, что определенные муравьи, нашедшие наилучшие решения за итерацию, получают особое внимание при обновлении феромонов. В отличие от базового алгоритма AS, где все муравьи одинаково влияют на обновление феромонов, в EAS лучшие муравьи вносят больший вклад в это обновление. Это позволяет сходиться к оптимальным решениям быстрее.
- **AS-RANK** (Rank-based Ant System) - Применяется ранговая селекция, при которой муравьи оцениваются по качеству найденного решения. Лучшие муравьи имеют больший вес при обновлении феромонов, что способствует повышению эффективности поиска.
- **MMAS** (Max-Min Ant System) - Вводит ограничение на максимальное и минимальное количество феромонов для улучшения устойчивости к локальным минимумам.
- **ACS** (Ant Colony System) - вводятся два вида обновлений феромонов глобальное и локальное. Глобальное обновление происходит после нахождения лучшего решения (глобально лучшего муравья), а локальное - во время построения маршрутов. Это позволяет ускорить сходимость к лучшим решениям.

Кроме того, для каждой из этих модификаций реализован вариант с применением локального поиска 2-opt к лучшему результату на итерации.

На рисунках 1–2 представлены кривые изменения длины лучшего найденного пути в зависимости от номера итерации для всех рассмотренных алгоритмов и их 2-opt-модификаций.

В дополнение к графикам мы собрали сводную статистику по пяти модификациям ACO в таблице 1. В ней приведены результаты для каждого метода и каждого набора задач (tsp_51_1, tsp_100_3, tsp_200_2).

Из всех модификаций наилучшие результаты на тестовых данных имеет модификация ACS. Также на графиках можно заметить, что, в отличие от других алгоритмов, ACS имеет достаточно хорошие значения оптимизируемой функции уже на первых итерациях.

3 PSO

В настоящей работе мы сравниваем две популярные реализации PSO:

- **Discrete PSO** — метод, предложенный Onwubolu, Babu и Clerc, в котором частицы оперируют перестановками вершин, а шаги перемещения реализованы через специальные операторы обмена и вставки [2].
- **IHPSO** — гибрид PSO с генетическим оператором селекции и кроссовера, представленный Wei и соавторами в 2021 году, что позволяет улучшать локальный поиск за счёт операторов генетической мутации [3]. Расстояния между вершинами используются для генерации некоторых вероятностных распределений в каждой вершине. Операторы используют информацию о ближайших соседях вершин и соответствующих вершинам распределениях. Такие модификации помогают избегать застревания в локальных минимумах и поддерживать разнообразие в популяции частиц.

В этой реализации дополнительно применяется *кольцевая топология* (ring topology) обмена информацией между частицами, где каждая частица связана только с двумя соседями из кольца [1].

Стоит отметить, что в обоих методах применялась "жадная инициализация" частиц для ускорения работы алгоритма на достаточно объемных данных. Если инициализировать частицы случайными путями, значения длин пути на малых итерациях будут больше, а сходимость к минимуму будет происходить за большее число итераций.

По сравнению с Discrete PSO алгоритм IHPSO имеет лучшую динамику сходимости (3, 4) и лучшие результаты на тестовых данных (2).

По графикам 4, 5 можно про наблюдать эффект использования топологии кольца: алгоритм не так часто застывает в локальных минимумах оптимизируемого функционала.

4 Общее сравнение

В целом муравьиный алгоритм в лучшей реализации справляется с решением задачи TSP лучше, чем PSO за одинаковое число шагов, показывая вместе с тем меньшее время работы. При этом PSO способен решать более широкий класс задач и имеет возможность улучшать результаты для конкретной задачи за счёт использования дополнительных модификаций алгоритма.

На объёмных графах оба алгоритма могут работать достаточно медленно. В обоих случаях рост времени работы будет полиномиальным при сохранении размера популяций. При этом слишком маленькое их число может негативно сказаться на качестве алгоритма.

Список литературы

- [1] KENNEDY, J., AND MENDES, R. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)* (2002), vol. 2, IEEE, pp. 1671–1676.

- [2] ONWUBOLU, G. C., BABU, B., AND CLERC, M. Discrete particle swarm optimization, illustrated by the traveling salesman problem. *New optimization techniques in engineering* (2004), 219–239.
- [3] WEI, B., XING, Y., XIA, X., AND GUI, L. A novel particle swarm optimization with genetic operator and its application to tsp. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* 15, 4 (2021), 1–17.

Приложение

GitHub Проекта: <https://github.com/artlvruran/ACO-and-PSO-for-TSP>

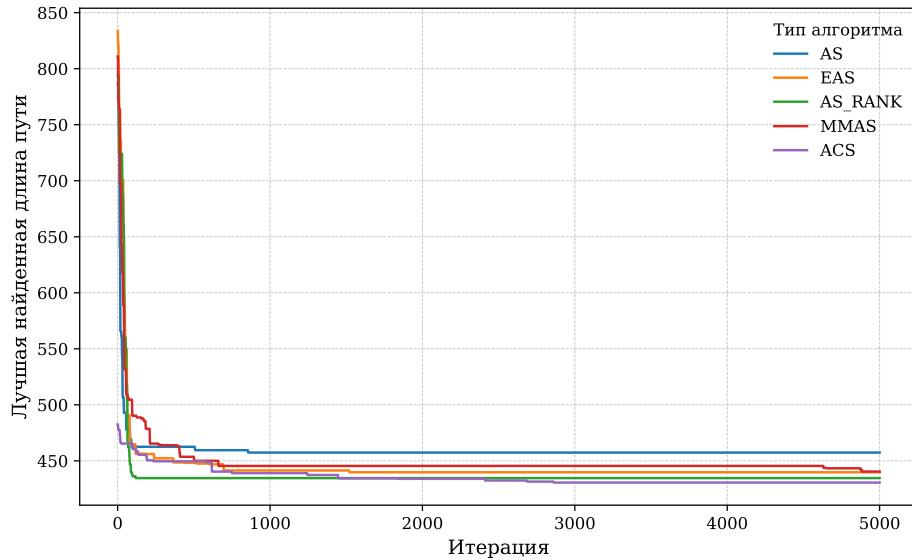


Рис. 1: Сравнение динамики длины лучшего пути для разных вариантов АСО. (датасет tsp_51_1)

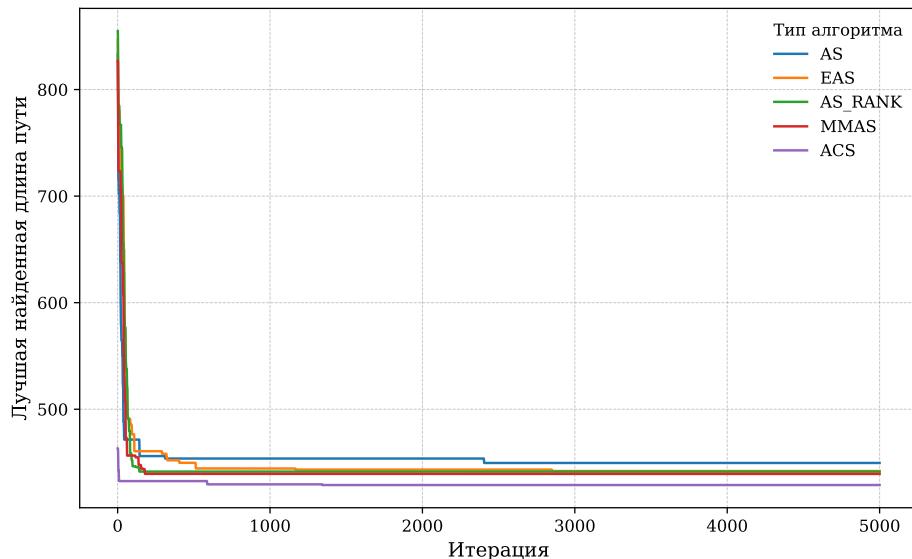


Рис. 2: Сравнение динамики длины лучшего пути с использованием локального поиска 2-opt. (датасет tsp_51_1)

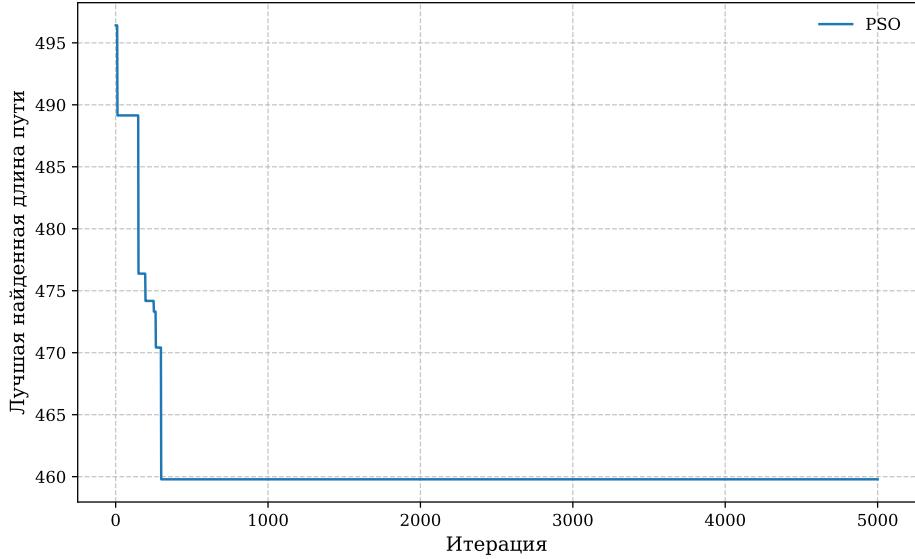


Рис. 3: Динамика длины лучшего пути для Discrete PSO (датасет tsp_51_1)

Таблица 1: Сводная статистика ACO-алгоритмов по трём TSP-наборам (10 запусков, 5000 итераций, 50 муравьев)

Датасет	Алгоритм	Средняя лучшая длина	Минимальная лучшая длина	Порог-1	Порог-2	Среднее время работы (с)
tsp_51_1	AS	447.72	441.76	482	430	1760.65
tsp_51_1	EAS	441.03	436.79			1810.35
tsp_51_1	AS-RANK	434.65	431.61			1258.30
tsp_51_1	MMAS	441.04	428.98			1226.87
tsp_51_1	ACS	433.97	428.98			618.56
tsp_100_3	AS	21611.07	21449.83	23433	20800	1675.95
tsp_100_3	EAS	21440.47	21410.99			1491.04
tsp_100_3	AS-RANK	21751.35	21368.69			1543.75
tsp_100_3	MMAS	21376.42	21108.05			1556.40
tsp_100_3	ACS	21004.78	20750.76			1054.38
tsp_200_2	AS	32719.26	31902.34	35985	30000	3177.32
tsp_200_2	EAS	32255.57	31780.44			2907.17
tsp_200_2	AS-RANK	32013.94	31100.43			2900.62
tsp_200_2	MMAS	33025.84	32443.83			2863.54
tsp_200_2	ACS	30550.39	30008.07			1166.74

Таблица 2: Сводная статистика PSO-алгоритмов по трём TSP-наборам (10 запусков, 5000 итераций, 50 частиц)

Датасет	Алгоритм	Средняя лучшая длина	Минимальная лучшая длина	Порог-1	Порог-2	Среднее время работы (с)
tsp_51_1	Discrete PSO	465.02	459.79	482	430	1662.15
tsp_51_1	IHPSO	439.87	431.67			1266.23
tsp_51_1	IHPSO+Ring	430.05	428.98			784.63
tsp_100_3	Discrete PSO	23192.46	23164.50	23433	20800	3198.35
tsp_100_3	IHPSO	22719.61	22125.60			1768.25
tsp_100_3	IHPSO+Ring	21278.93	20881.61			1260.44
tsp_200_2	Discrete PSO	35252.77	34855.65	35985	30000	5144.90
tsp_200_2	IHPSO	32988.09	32030.85			2218.16
tsp_200_2	IHPSO+Ring	32045.33	31375.74			2046.30

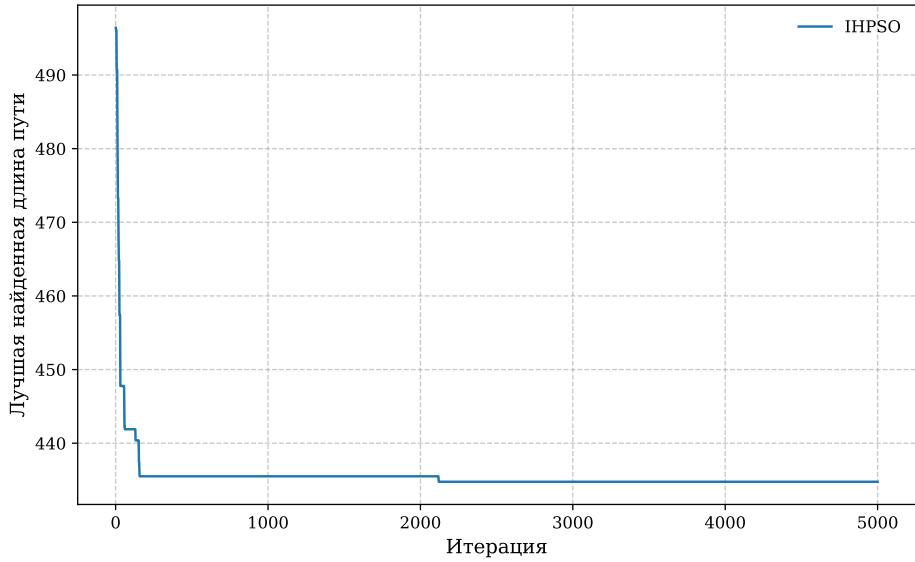


Рис. 4: Динамика длины лучшего пути для IHPSO (датасет tsp_51_1)

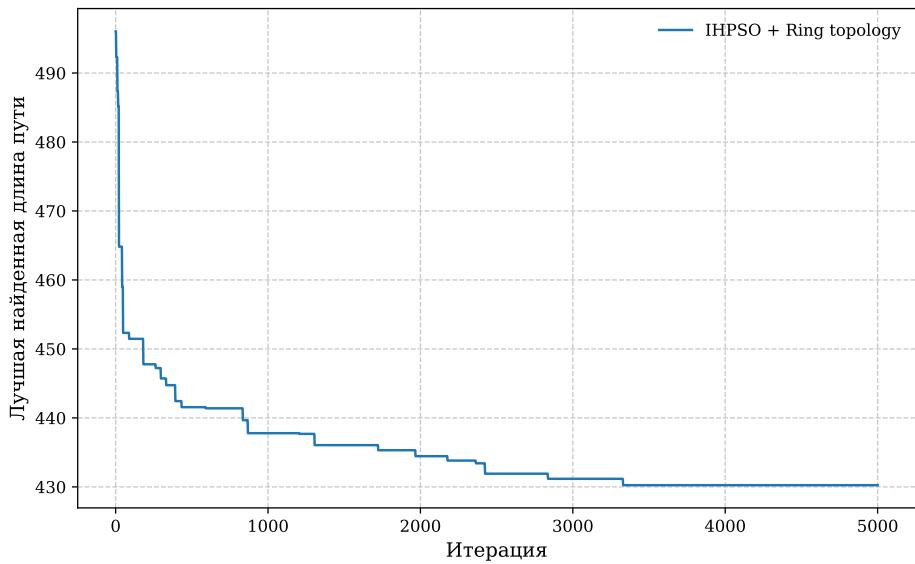


Рис. 5: Динамика длины лучшего пути для IHPSO + Ring topology (датасет tsp_51_1)