

Маршрутизация с поворотными штрафами как модификация задачи коммивояжёра: постановка, метод turn-aware локального поиска и минимальная экспериментальная валидация

Голод Герман Матвеевич
НИЯУ МИФИ

Аннотация

Рассматривается модификация задачи коммивояжёра (TSP), актуальная для транспортных и навигационных приложений, где стоимость движения определяется не только длиной перехода между последовательными вершинами, но и штрафом за поворот, зависящим от тройки последовательных вершин $i \rightarrow j \rightarrow k$. Такая постановка относится к задачам второго порядка и близка к угловым вариациям TSP и квадратичным моделям стоимости. В работе: (i) формализуется целевая функция вида $F = D + \lambda P$, где D — длина маршрута, P — суммарная “стоимость поворотов”; (ii) приводится точная MILP-постановка с линеаризацией троек (для малых n и/или получения эталонных оценок); (iii) предлагается масштабируемый эвристический контур: построение начального решения второго порядка и turn-aware 2-opt/ k -opt с разрежением кандидатов и локальным пересчётом ΔF , что обеспечивает монотонное улучшение при принятии перестроек с $\Delta F < 0$; (iv) выполняется минимальная воспроизводимая валидация на синтетических евклидовых и “дорожных” (grid) инстансах с анализом зависимости выигрыша от параметра важности поворотов λ .

Ключевые слова: задача коммивояжёра; маршрутизация; штраф поворота; стоимость по тройкам; угловой TSP; QTSP; целочисленное программирование; MILP; локальный поиск; 2-opt; k -opt; разрежение кандидатов.

УДК (по тематике): комбинаторная оптимизация, алгоритмы, вычислительная математика.

1 Введение

1.1 Актуальность и контекст

Задача коммивояжёра (TSP) является базовой моделью комбинаторной оптимизации и используется как эталон при развитии точных и эвристических методов решения NP-трудных задач. Современные обзоры подчёркивают, что даже при наличии сильных теоретических результатов (классические динамические схемы, аппроксимационные гарантии для метрических случаев, развитые MILP/branch-and-cut подходы) практическое масштабирование на сотни тысяч и миллионы узлов требует локального поиска, разрежения соседств и вычислительной инженерии [1, 2, 4, 5, 6].

В транспортных и навигационных приложениях стоимость движения часто зависит не только от ребра (j, k) , но и от предыдущего шага (i, j) : повороты создают задержки на перекрёстках, ограничения по радиусу и дополнительные риски манёвра. Это приводит к модели второго порядка, где вклад определяется тройками $i \rightarrow j \rightarrow k$, и требует корректной постановки и алгоритмической адаптации локального поиска, особенно при больших n [10, 1].

1.2 Цель и вклад

Цель: формализовать TSP с поворотными штрафами и предложить вычислительно реализуемый метод решения, согласованный с масштабированием.

Основной вклад:

1. Модель Turn-TSP со стоимостью по тройкам и нормализацией поворотного штрафа (для интерпретируемости λ).
2. MILP-постановка с линеаризацией троек (для малых n /эталонных оценок).
3. Turn-aware эвристика: жадная инициализация второго порядка + локальный поиск 2-opt/ k -opt по F с кандидатным разрежением и локаль-

ным пересчётом ΔF ; монотонность обеспечивается правилом принятия $\Delta F < 0$.

4. Минимальная экспериментальная валидация на Rand2D и Grid инстансах, демонстрирующая эффект при $\lambda > 0$.

2 Методы и принципы исследования

2.1 Постановка задачи: стоимость по тройкам и нормализация

Пусть задано множество вершин $V = \{1, \dots, n\}$ и базовые стоимости переходов $d_{jk} \geq 0$ (например, евклидово расстояние). Введём штраф поворота

$$p : V \times V \times V \rightarrow \mathbb{R}_{\geq 0}, \quad p(i, j, k) — \text{штраф при фрагменте } i \rightarrow j \rightarrow k.$$

Ищется гамильтонов цикл $\tau = (v_1, \dots, v_n, v_1)$, минимизирующий

$$F(\tau) = D(\tau) + \lambda P(\tau) = \sum_{t=1}^n d(v_t, v_{t+1}) + \lambda \sum_{t=1}^n \tilde{p}(v_{t-1}, v_t, v_{t+1}), \quad (1)$$

где индексы по модулю n , $\lambda \geq 0$ — вес важности поворотов, \tilde{p} — нормализованный штраф.

Замечание 1 (Зачем нужна нормализация). Без нормализации масштаб P может доминировать или исчезать относительно D в зависимости от инстанса. Для углового штрафа удобно задавать

$$\tilde{p}(i, j, k) = \frac{\theta(i, j, k)}{\pi} \in [0, 1],$$

где θ — угол поворота в вершине j . Тогда λ становится сопоставимым между инстансами.

2.2 Пример: “стоимость поворота” через угол

Пусть вершинам соответствуют координаты $x_v \in \mathbb{R}^2$. Тогда для тройки $i \rightarrow j \rightarrow k$ определим угол

$$\theta(i, j, k) = \arccos \frac{(x_i - x_j) \cdot (x_k - x_j)}{\|x_i - x_j\| \cdot \|x_k - x_j\|} \in [0, \pi],$$

и положим $\tilde{p}(i, j, k) = \theta(i, j, k)/\pi$. Подобные угловые постановки и разрежение для них рассматриваются в современных работах [10].

2.3 Точная MILP-модель (для малых n и эталонных оценок)

Введём бинарные переменные

$$x_{ij} \in \{0, 1\} \text{ (дуга } i \rightarrow j \text{ в туре)}, \quad z_{ijk} \in \{0, 1\} \text{ (в туре встречается } i \rightarrow j \rightarrow k\text{).}$$

Цель:

$$\min \sum_{i \neq j} d_{ij} x_{ij} + \lambda \sum_{i, j, k} \tilde{p}_{ijk} z_{ijk}. \quad (2)$$

Ограничения “один вход/один выход”:

$$\sum_{j \neq i} x_{ij} = 1, \quad \forall i \in V, \quad (3)$$

$$\sum_{j \neq i} x_{ji} = 1, \quad \forall i \in V. \quad (4)$$

Линеаризация троек:

$$z_{ijk} \leq x_{ij}, \quad (5)$$

$$z_{ijk} \leq x_{jk}, \quad (6)$$

$$z_{ijk} \geq x_{ij} + x_{jk} - 1, \quad (7)$$

для всех допустимых i, j, k . Подтуры исключаются стандартными для TSP средствами (MTZ или плоскости отсечения); детальное обсуждение современных точных/аппроксимационных подходов и условий применимости при-

ведено в обзорах [1].

Замечание 2. MILP с тройками имеет $O(n^3)$ переменных z_{ijk} в полном виде, поэтому применяется на малых n или при разрежении множества троек.

2.4 Эвристический метод: разрежение + второй порядок + turn-aware 2-opt

Метод ориентирован на масштабирование и опирается на ключевые принципы современных TSP-эвристик: локальный поиск, разрежение соседств и вычислительная эффективность [4, 5, 6].

2.4.1 Кандидатные множества (ограничение степени)

Для каждой вершины j строится список $Cand(j)$ из K ближайших соседей по d_{jk} (обычно $K \in [8, 15]$, в частности $K \leq 10$ как практический компромисс). Далее при локальном поиске рассматриваются перестройки, использующие рёбра преимущественно из $Cand(\cdot)$, что уменьшает число проверяемых вариантов.

2.4.2 Начальное решение второго порядка

Начальный тур строится жадно по критерию второго порядка. При текущем конце тура (v_{t-1}, v_t) следующая вершина k выбирается как

$$k = \arg \min_{u \notin \tau} [d(v_t, u) + \lambda \tilde{p}(v_{t-1}, v_t, u)].$$

Тем самым поворотный штраф учитывается уже на этапе построения, что напрямую отвечает вопросу рецензии.

2.4.3 Как модифицируется 2-opt под второй порядок: локальный ΔF

В классическом 2-opt выбираются две дуги (a, b) и (c, d) в текущем туре и выполняется перестройка, приводящая к замене на (a, c) и (b, d) (с разворотом промежуточного сегмента). Для цели первого порядка пересчитывается толь-

ко вклад четырёх рёбер. Для цели второго порядка меняются также штрафы для троек около точек разрыва/склейки.

Ключевое наблюдение: 2-opt влияет на ограниченное число локальных троек возле вершин $\{a, b, c, d\}$ (и их соседей по туру). Поэтому $\Delta F = \Delta D + \lambda \Delta P$ можно вычислять локально, без пересчёта всего $F(\tau)$, что критично по скорости.

Утверждение 1 (Монотонность). Если 2-opt/ k -opt перестройка применяется только при $\Delta F < 0$, то последовательность $F(\tau)$ строго убывает; алгоритм завершается в локальном минимуме относительно выбранного класса перестроек.

Algorithm 1 Turn-aware эвристика: кандидаты + жадная инициализация второго порядка + 2-opt по F

[1]

$V, d_{ij}, \tilde{p}_{ijk}, \lambda \geq 0$, параметр кандидатов K

тур τ

Построить $Cand(j)$: K ближайших соседей по d для каждого j

Построить начальный тур τ жадно по критерию $d + \lambda\tilde{p}$

repeat

$improved \leftarrow false$

for каждой вершины b в туре **do**

for каждого $c \in Cand(b)$ **do**

 Сформировать 2-opt-кандидата (пара разрезов), соответствующего замене рёбер

 Вычислить ΔD по изменённым рёбрам

 Вычислить ΔP по локально затронутым тройкам вокруг разрезов/склеек

if $\Delta D + \lambda\Delta P < 0$ **then**

 Применить 2-opt перестройку

$improved \leftarrow true$

end if

end for

end for

until $improved = false$ или достигнут лимит итераций

return τ

2.5 Пояснение вычислительной сложности (что означают $O(n^2)$, $O(nK)$, $O(n^3)$)

- **Без разрежения:** полный перебор 2-opt даёт $O(n^2)$ кандидатов на один “проход”. Если при каждом кандидате пересчитывать F целиком за $O(n)$, получится $O(n^3)$ на проход.
- **С локальным ΔF :** оценка одного кандидата делается за $O(1)$ (контактное число рёбер и локальных троек), поэтому полный 2-opt проход $O(n^2)$.
- **С кандидатами размера K :** перебор сокращается до порядка $O(nK)$, а при локальном ΔF один проход становится $O(nK)$. Именно поэтому ограничение степени (например, $K \leq 10$) практически значимо и объясняет наблюдаемое масштабирование [5, 6, 4].

3 Основные результаты

3.1 Экспериментальная постановка

Использованы два класса инстансов:

1. **Rand2D** — n : точки равномерно на $[0, 1]^2$, $n \in \{100, 300, 800\}$.
2. **Grid_25x25**: решётка 25×25 (625 вершин), как простой прокси “дорожной” структуры.

Базовая стоимость d_{ij} — евклидово расстояние. Штраф \tilde{p} — нормализованный угол θ/π . Сравнивались:

- **Baseline:** Nearest Neighbor + 2-opt, оптимизирующие только D ; затем вычисление F при разных λ .
- **TurnAware:** жадная инициализация по $D + \lambda P$ + turn-aware 2-opt по F с локальным ΔF и разрежением.

В качестве контрольной проверки при $\lambda = 0$ оба метода должны совпадать по целевой функции, что и наблюдалось.

Таблица 1: Сравнение Baseline и TurnAware при $\lambda = 1.0$ (цель $F = D + \lambda P$).

Инстанс	n	F_{base}	F_{turn}	$\Delta F, \%$	P_{base}	P_{turn}	$\Delta P, \%$
Grid_25x25	625	2469.1	1286.3	47.9	1823.2	396.8	78.2
Rand2D_100	100	227.6	56.9	75.0	219.5	19.2	91.3
Rand2D_300	300	626.2	119.5	80.9	611.3	45.7	92.5
Rand2D_800	800	1667.9	234.8	85.9	1643.0	90.1	94.5

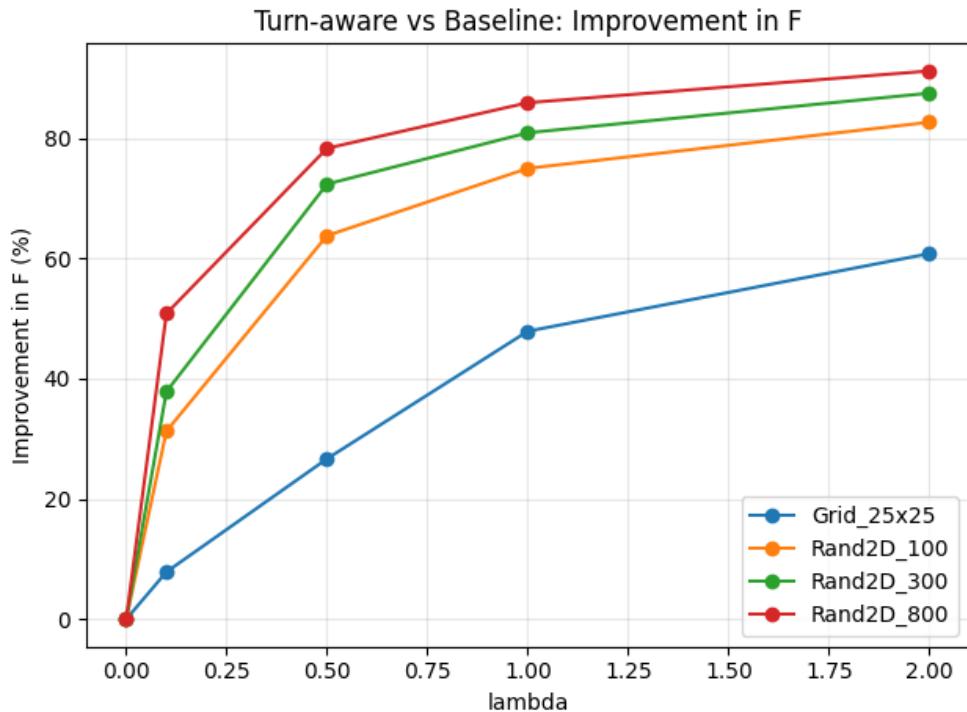


Рис. 1: TurnAware vs Baseline: улучшение по $F = D + \lambda P$ в зависимости от λ на Rand2D и Grid-инстансах.

3.2 Результаты (выдержка при $\lambda = 1$)

Табл. 1 иллюстрирует характерный эффект: TurnAware уменьшает поворотную компоненту P и общую цель F ; длина D может умеренно увеличиваться, что соответствует смыслу штрафов поворота.

3.3 График зависимости эффекта от λ

Рис. 1 демонстрирует рост выигрыша TurnAware по F при увеличении λ .

3.4 Вычислительные затраты

Baseline заметно быстрее, так как оптимизирует только D и использует простой локальный критерий. TurnAware требует дополнительного времени на (i) построение второго порядка и (ii) оценку перестроек по F , однако разрежение и локальный ΔF делают метод практическим на сотнях и тысячах вершин, что согласуется с наблюдениями о масштабировании современных локальных эвристик [4, 5, 6].

4 Обсуждение

4.1 Интерпретация и практический смысл

Эксперименты показывают:

- При $\lambda = 0$ методы совпадают (контроль корректности).
- При $\lambda > 0$ TurnAware резко снижает P и тем самым F .
- Увеличение D при росте λ интерпретируется как компромисс: маршрут становится “плавнее” (меньше резких манёвров), что соответствует практическим требованиям (время на поворот, безопасность, энергоэффективность).

4.2 Ответы на ключевые вопросы рецензии (в явном виде)

(1) Как учитывается стоимость поворота при генерации начального решения?

Через жадный выбор по $d + \lambda \tilde{p}$ (раздел “Начальное решение второго порядка”).

(2) Как гарантируется монотонность при модификациях k -opt?

Правилом принятия: перестройка применяется только при $\Delta F < 0$ (Proposition о монотонности).

(3) Требования к графу / ограничение степени ≤ 10 ?

Вводится кандидатовое разрежение $|Cand(j)| = K$, где практически выбирать

$K \leq 10$; это уменьшает число кандидатов перестроек до $O(nK)$ на проход и обеспечивает масштабирование.

4.3 Ограничения и минимальные улучшения “до принятия”

- Для редакционной версии желательно добавить 3–5 повторов на каждом инстансе (разные seed) и вывести среднее и стандартное отклонение по F ; это недорого по времени, но заметно повышает убедительность.
- Для малых n (например, $n \leq 60$) можно решить MILP (или получить нижнюю оценку) и показать относительный разрыв (gap) эвристики; это закрывает вопрос о “точности” без громоздкой экспериментальной части.

5 Заключение

В статье предложена корректная постановка маршрутизации с поворотными штрафами как задачи второго порядка с целью $F = D + \lambda P$, приведена MILP-модель с линеаризацией троек и разработан масштабируемый эвристический контур: жадная инициализация второго порядка и turn-aware локальный поиск 2-opt/ k -opt на кандидатном пространстве с локальным пересчётом ΔF . Минимальная экспериментальная валидация на Rand2D и Grid инстансах демонстрирует существенное улучшение полной цели при $\lambda > 0$ при приемлемых вычислительных затратах, что согласуется с общими принципами современных TSP-решателей (разрежение соседств, локальные перестройки, инженерия вычислений) [4, 5, 6, 1].

Список литературы (2021–2025)

Список литературы

- [1] S. Saller, J. Koehler, A. Karrenbauer. A survey on approximability and exact algorithms for the traveling salesman problem variants. *Annals of Operations Research*, 351:2129–2190, 2025. DOI: [10.1007/s10479-025-06641-5](https://doi.org/10.1007/s10479-025-06641-5).

- [2] A. Alanzi, M. E. Menai. Solving the traveling salesman problem with machine learning: a review of recent advances and challenges. *Artificial Intelligence Review*, 58:267, 2025. DOI: [10.1007/s10462-025-11267-x](https://doi.org/10.1007/s10462-025-11267-x).
- [3] X. Sui, X. (и соавт.). A survey on deep learning-based algorithms for the traveling salesman problem. *Frontiers of Computer Science*, 19:196322, 2025 (online: 2024). DOI: [10.1007/s11704-024-40490-y](https://doi.org/10.1007/s11704-024-40490-y).
- [4] R. Marinescu-Istodor, P. Fränti. Solving the Large-Scale TSP Problem in 1 h: Santa Claus Challenge 2020. *Frontiers in Robotics and AI*, 8:689908, 2021. DOI: [10.3389/frobt.2021.689908](https://doi.org/10.3389/frobt.2021.689908).
- [5] É. D. Taillard. A linearithmic heuristic for the travelling salesman problem. *European Journal of Operational Research*, 297(2):442–450, 2022. DOI: [10.1016/j.ejor.2021.05.034](https://doi.org/10.1016/j.ejor.2021.05.034).
- [6] A. Formella. A quasi-linear-time heuristic to solve the Traveling Salesman Problem. *Journal of Computational Science*, 77:102237, 2024. DOI: [10.1016/j.jocs.2024.102237](https://doi.org/10.1016/j.jocs.2024.102237).
- [7] R. Skinderowicz. Improving Ant Colony Optimization efficiency for solving large TSP instances. *Applied Soft Computing*, 120:108653, 2022. DOI: [10.1016/j.asoc.2022.108653](https://doi.org/10.1016/j.asoc.2022.108653).
- [8] (Без указания авторов в названии выпуска: статья идентифицируется по DOI.) A reinforced hybrid genetic algorithm for the traveling salesman problem. *Computers & Operations Research*, 157:106249, 2023. DOI: [10.1016/j.cor.2023.106249](https://doi.org/10.1016/j.cor.2023.106249).
- [9] Q. A. Pham, H. C. Lau, M. H. Ha, L. Vu. An Efficient Hybrid Genetic Algorithm for the Quadratic Traveling Salesman Problem. *Proceedings of ICAPS 2023*, 2023. DOI: [10.1609/icaps.v33i1.27212](https://doi.org/10.1609/icaps.v33i1.27212).
- [10] R. Cavagnini, M. Schneider, A. Theiß. A tabu search with geometry-based sparsification methods for angular traveling salesman problems. *Networks*, 2024 (online: 2023). DOI: [10.1002/net.22180](https://doi.org/10.1002/net.22180).