



УНИВЕРСИТЕТ ИТМО

Решение задачи Travelling Thief Problem с помощью эволюционного алгоритма

Выполнил: ст. группы М4239 ФИТиП

Абзалтдинов Л.И.

Научный руководитель: к.т.н., доц. ФИТиП

Буздалов М. В

Актуальность

Сложные промышленные задачи:

- Оптимизация транспортировки водных резервуаров
- Дизайн печатных плат в сверхбольших интегральных схемах

Они не могут быть сведены полностью к теоретической задаче на 100%



Задача о воре

Постановка задачи

- Дано:
 - Имеется N городов и M предметов, для каждого предмета задан его вес, стоимость и в каких городах он находится
 - Вор берет в аренду рюкзак ограниченной вместимости
 - Скорость передвижения вора зависит от веса рюкзака
- Требуется обойти все города по одному разу, «украсть» в них предметы таким образом, чтобы они уместились в рюкзак, и при этом максимизировать общий доход, который складывается из стоимостей украденных предметов за вычетом стоимости аренды рюкзака.

Математическая модель (1/2)

TSP (подзадача коммивояжера):

- Дано:
 - Число городов: n
 - Матрица расстояний между городами $(d_{ij}), i, j \in N$
- Решение:
 - $\bar{\Pi} = (\pi_1, \dots, \pi_n)$ – кратчайший замкнутый маршрут передвижения вора, где x_i – номер города

KP (подзадача о рюкзаке):

- Дано:
 - Количество предметов: m
 - Вес каждого предмета: w_k
 - Стоимость каждого предмета: p_k
 - Вместительность рюкзака: W
- Решение:
 - $\bar{P} = (p_1, \dots, p_m)$ – бинарный вектор предметов, который показывает, какие из предметов собраны

Математическая модель (2/2)

ТТР₁ (задача о воре)^[1]:

- Дано:
 - Доступность предмета I_i в каждом городе: $A_i \subseteq \{1, \dots, n\}$
 - Скорость вора: $v_c = v_{max} - W_c \frac{v_{max} - v_{min}}{W}$, где W_c – текущий вес рюкзака
 - Стоимость аренды рюкзака: $\$R$ за ед. времени
- Найти
 - $\bar{\Pi} = (\pi_1, \dots, \pi_n)$ – маршрут передвижения вора,
 - $\bar{P} = (p_1, \dots, p_m), p_i \in \{0 \cup A_i\}$ – план упаковки предметов, где z_i показывает, из какого города предмет I_i должен быть «украден»
- Критерий оптимизации:

$$\max Z(\Pi, P) = profit(P) - R \cdot time(\Pi, P)$$

Сложность – $O(n + m)$

$profit(P)$ – общая стоимость собранных предметов,

$time(\Pi, P)$ – время прохождения маршрута

Цель и задачи работы

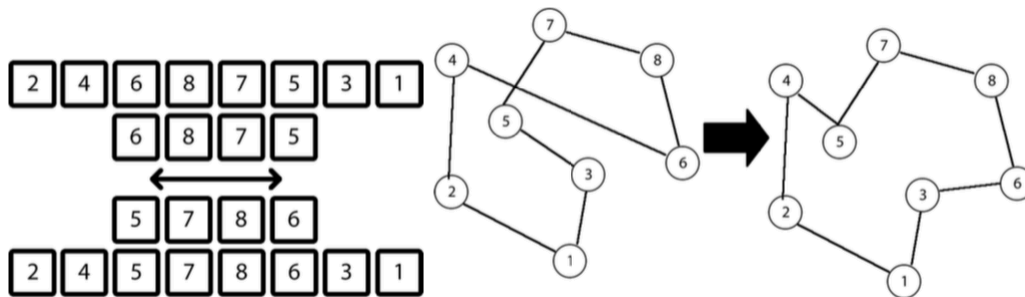
Цель: Разработка эволюционного алгоритма для решения задачи о воре и исследование его эффективности

Задачи:

- Выполнить аналитический обзор методов решения задачи о воре
- Предложить локальные операторы для решения задачи
- Разработать эволюционный алгоритм решения задачи о воре
- Провести сравнение эффективности решения задачи о воре разработанным алгоритмом по сравнению с существующими решениями

Проблема локальности операторов

- Оператор 2-OPT – переворачивает некоторую подпоследовательность маршрута.
 - В TSP имеет широкое применение и хорошие результаты^[1]
 - Для TTP сильно влияет на решение – важна последовательность городов



Проблема локальности операторов

- Оператор BitFlip – меняет статус случайного предмета в рюкзаке.
 - В КР придает небольшое изменение
 - Для ТТР может сильно повлиять – если предмет будет взят в начале маршрута



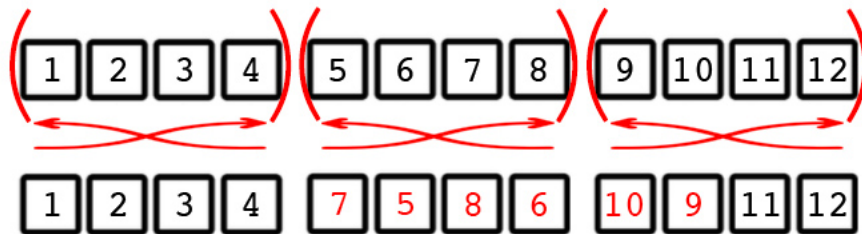
Необходимы более локальные операторы!

Локальное улучшение маршрута *(Optimal Subtour Search)*

Оптимизация подпоследовательностей маршрута небольшой длины

1. Поделить маршрут на $\lceil n/k \rceil$ подпоследовательностей небольшой длины k ($k = 5$)
2. Сгенерировать $k!$ перестановок для каждой подпоследовательности маршрута
3. Выбрать наилучшую перестановку городов в каждой подпоследовательности **(оптимизация ТТР (не TSP!) с заданным рюкзаком)**

Сложность $O(n \cdot k!)$



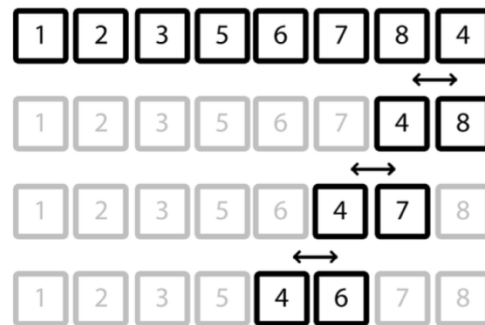
Локальное улучшение маршрута (*Insertion*)

Вставка каждого города в оптимальную позицию

Для каждого города i делать следующее:

1. Переместить в соседнюю позицию $\pi(i) + 1$ (или $\pi(i) - 1$)
2. Пересчитать целевую функцию ТТР ускоренным методом
3. Если результат стал лучше – запомнить маршрут
4. Повторить, пока $\pi(i) \neq 1$ (или $\pi(i) \neq n$)

Сложность $O(n^2)$



Локальные улучшения рюкзака *(Pack Later)*

Из одинаковых предметов лучше брать те, что ближе к концу маршрута

Препроцессинг: сгруппировать предметы с одинаковыми весами и стоимостями.

1. Каждую группу предметов отсортировать в порядке появления в городах маршрута $O(m \cdot \log(m))$
2. Переместить предметы из начала маршрута в конец

Сложность $O(m \cdot \log(m))$ – в худшем случае, реально – быстрее

Локальное улучшения рюкзака

(Pack Better)

Выбор более «хорошего» предмета вместо текущего

Один предмет «лучше» другого, если он одновременно

- а) легче;
- б) дороже;
- в) находится ближе к концу маршрута.

Реализация:

1. Сравним каждый предмет со всеми другими, если он «лучше» – добавим его в план упаковки рюкзака, удалив текущий.

Сложность $O(m^2)$, поэтому применяется, если $m < 10^4$

Существующие алгоритмы решения задачи

Алгоритмы, генерирующие единственное решение

- Random Local Search (RLS)
- (1+1) EA
- CS2SA
- S1-S5, C1-C6

Меметические алгоритмы

- MMAS
- MATLS
- MA2B

На данный момент не существует наилучшего алгоритма^[1]!

Схема предлагаемого алгоритма

1. Запустить алгоритм CS2SA
2. Запустить предложенные операторы:
 1. Optimal Subtour Search
 2. Insertion
 3. PackLater ($m > 10^4$) или PackBetter ($m \leq 10^4$)
3. Если найдено улучшение, то перейти к п. 1 с обновленным решением

Тестовые данные

Маршрут	Тип корреляции весов/стоимостей предметов	Кол-во предметов в городе	Вместимость рюкзака
$\begin{bmatrix} 81 \\ \text{экземпляр} \\ \text{из TSPLIB} \end{bmatrix}$	$\times \begin{bmatrix} \text{Uncorr} \\ \text{uncorr-similar-weights} \\ \text{bounded-strongly-corr} \end{bmatrix}$	$\times \begin{bmatrix} 1 \\ 3 \\ 5 \\ 10 \end{bmatrix}$	$\times \begin{bmatrix} 1 \\ \dots \\ 10 \end{bmatrix}$

= 9720 экземпляров TTP

Тестовые данные

Маршрут	Тип корреляции весов/стоимостей предметов	Кол-во предметов в городе	Вместимость рюкзака
$\left[\begin{array}{c} 11 \\ \text{экземпляров} \\ \text{из TSPLIB} \end{array} \right]$	$\times \left[\begin{array}{c} \text{Uncorr} \\ \text{uncorr-similar-weights} \\ \text{bounded-strongly-corr} \end{array} \right]$	$\times \left[\begin{array}{c} 3 \\ 10 \end{array} \right]$	$\times \left[\begin{array}{c} 3 \\ 7 \end{array} \right]$

Для тестирования выбрано
= 132 экземпляра TTP

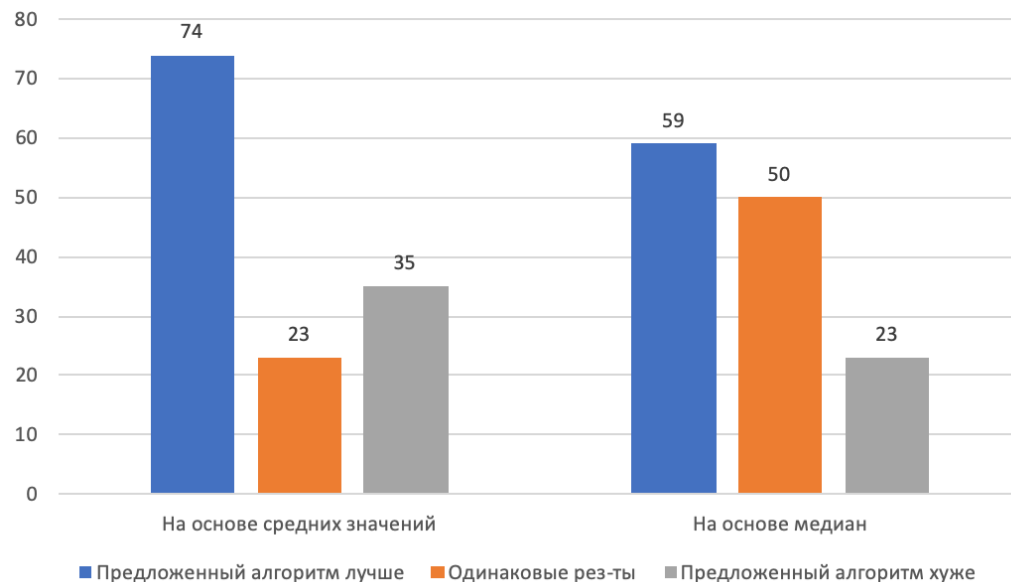
Результаты сравнения алгоритмов

Для сравнения выбрана
модификация CS2SA с
множественным запуском в
течение 10 минут – CS2SA-R^[1]

Ограничение по времени
– 10 минут

Каждая задача тестировалась
по 10 раз

Сравнение результатов запусков алгоритмов по
задачам



Статистический анализ результатов сравнения

Критерий Уилкоксона-Манна-Уитни (Mann-Whitney U test)

H_0 – распределения двух выборок одинаковы

X – результаты предложенного алгоритма, Y – результаты CS2SA-R

$\alpha = 0,0005$ – уровень значимости

H_1 - доминирование X над Y

H_1 - доминирование Y над X

H_1 принята в 28 случаях из 132

H_1 принята в 9 случаях из 132

В ≈ 3 раза чаще предложенный алгоритм лучше, нежели хуже

Статистический анализ результатов сравнения

Поправка Холма-Бонферрони (Holm-Bonferroni correction)

Чем больше гипотез – тем больше вероятность ошибки хотя бы в одной.

$\alpha = 0,05$ – уровень значимости

H_1 - доминирование X над Y

H_1 - доминирование Y над X

H_1 принята в 28 случаях из 132

H_1 принята в 9 случаях из 132

В ≈ 3 раза чаще предложенный алгоритм лучше, нежели хуже

Выводы

- Достигнуто:
 - Предложено 4 оператора локальной оптимизации;
 - Разработан эволюционный алгоритм, который показывает примерно в 3 раза чаще результаты лучше, нежели хуже.
- Дальнейшая работа:
 - Провести полноценное сравнение с другими существующими алгоритмами;
 - Выяснить проблемы в задачах, где не найдены улучшения.

Спасибо за внимание!

ITMO *re than a*
UNIVERSITY

Публикации

- ✓ *Абзалтдинов Л.И.* Решение задачи о воре с помощью эволюционного алгоритма // Неделя науки СПбПУ : материалы научной конференции с международным участием. Институт компьютерных наук и технологий. — СПб. : Изд-во Политехн. ун-та, 2018. — с. 41-42.
- ✓ *Абзалтдинов Л.И.* Эволюционный алгоритм для решения задачи Travelling Thief Problem // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. - 2019 [Электронный ресурс]. - Режим доступа: <https://kmu.itmo.ru/digests/article/668>, своб.

Быстрый пересчет целевой функции

При неизменном плане упаковки рюкзака

$$Z(\Pi, P) = \overset{\text{const}}{\text{profit}(P)} - R \cdot \sum_{i=1}^n \frac{d(\pi_i, \pi_{i+1})}{v_{\max} - W_i \frac{v_{\max} - v_{\min}}{W}}$$

$W_i = W_{i-1} + \overset{\text{const}}{\sum_{j \in A(\pi_i)} w_j}$ - текущий накопленный вес предметов

Вес предметов в городах не изменяется => при изменении подпоследовательности маршрута длины k пересчет целевой функции - $O(k)$