



Е.В. Рагулина

ИССЛЕДОВАНИЕ МЕТОДА РОЯ ЧАСТИЦ ДЛЯ ЗАДАЧ МАРШРУТИЗАЦИИ ТРАНСПОРТА НА ПРИМЕРЕ СОСТАВЛЕНИЯ РАСПИСАНИЯ ДВИЖЕНИЯ АГЕНТОВ

(ФГБОУ ВПО «Уфимский государственный авиационный
технический университет»)

Решение задачи составления расписания рассматривается как частный случай задачи маршрутизации транспорта (Vehicle Routing Problem, VRP) — задачи комбинаторной оптимизации, в которой для множества транспортных средств (агентов), расположенных в одном или нескольких депо, должен быть определен набор маршрутов до нескольких отдаленных точек — клиентов [1]. VRP — хорошо известная задача целочисленного программирования, относящаяся к классу NP-трудных задач, это означает, что вычислительная сложность задачи зависит от размера входных данных экспоненциально.

Постановка задачи. Транспортная сеть задана полным графом $G = (V, E)$, $V = v_0, \dots, v_n$, где вершина v_0 интерпретируется как склад, а v_1, \dots, v_n — клиенты.

Каждому ребру $e_{ij} \in E; i, j = 0 \dots n$ сопоставлена характеристика времени движения из пункта v_i в v_j , где $i, j = 0 \dots n$. Задано множество агентов $A = (a_1, \dots, a_q)$.

Требуется найти такое разбиение множества V на непересекающиеся

подмножества $V = \bigcup_{k=1}^q V^{a_k}; V^{a_k} \cap V^{a_l} = \emptyset; l \neq k; l, k = 1 \dots q$ и для каждого V^{a_k} определить последовательность вершин, то есть маршрут движения R^{a_k} , $k = 1 \dots q$ агента по клиентам. Расписанием R назовем множество маршрутов $\{(R)^{a_1} \dots R^{a_q}\}$ для всех агентов.

Необходимо найти расписание R , для которого самый долгий по времени маршрут будет минимальным.

Метод роя частиц. Рой частиц представляет собой множество объектов, обладающих роевым интеллектом [2]. В отличие от популяционных методов, алгоритм роя частиц не обновляет существующие популяции для создания новых. Он работает с одной статической популяцией, члены которой постепенно улучшаются с появлением информации о пространстве поиска. Данный метод представляет собой вид направленной мутации. Положение частицы в пространстве является аналогом генотипа в эволюционных алгоритмах. Также определены модуль и направление скорости, в котором частица перемещается в каждый момент времени [3].

Отслеживаются следующие моменты:

- наиболее приспособленное положение x^* , найденное частицей x ;



- наиболее приспособленное положение x^+ , найденное всеми информаторами частицы x . Информаторами считается небольшое множество частиц, случайно выбираемых на каждой итерации;
- наиболее приспособленное положение $x^!$, найденное всеми частицами.

Используются следующие коэффициенты:

α – определяет, какая сохраняется доля исходного вектора скорости.

β – определяет приоритет лучшего положения, найденного самой частицей.

γ – определяет приоритет лучшего положения, найденного информаторами.

δ – определяет приоритет лучшего положения, найденного всеми частицами (глобальное лучшее).

ε – определяет как быстро частица движется. Обычно, ε равно 1.

Способы применения PSO к задаче расписания агентов. Основной проблемой в реализации алгоритма для решения задачи расписания был вопрос о способе перехода от задачи в вещественном пространстве к дискретной задаче на графе, на котором построен алгоритм роя частиц. В качестве такого способа предлагается использовать декодер с матрицами весов [1].

Вектор координат, изначально заполненный случайными числами в диапазоне $[-1;1]$, хранится в одномерном массиве, из элементов которого будут составлены матрицы $A=\{a_{ij}\}$; $C=\{c_{ji}\}$; где a_{ij} – значимость агента i для клиента j ; c_{ji} – значимость клиента j для агента i , $i=1..m$; $j=1..n$.

Соотносим агентов и клиентов. В i -й строке матрицы C находим максимальный элемент, пусть этот элемент находится в столбце j . Тогда клиента i посетит агент j . Таким образом, имеем для каждого агента список посещаемых клиентов.

Сортируем список, определяющий порядок обхода. В i -й строке матрицы A сортируем элементы, соответствующие списку посещаемых клиентов по убыванию. Имеем порядок посещения.

Составляем матрицу расписаний R размерности $m*(n+3)$.

$$R = \begin{pmatrix} 0 & \{\text{список}\} & 0 \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix}, \text{ где } \{\text{список}\} \text{ – это отсортированный список с номерами}$$

вершин клиентов, то есть маршрут. Во всех строках матрицы R расположены маршруты для каждого агента.

Далее вычисляем целевую функцию по времени, как длительность прохождения пути самым медленным агентом:

$$T = \max_i \left(\sum_{j=1}^{n+2} D_{R_i,j} R_{i,j+1} \right) \longrightarrow \min, \text{ где } D \text{ – вес ребра } (c_j, c_{j+1}).$$

Недостаток данного метода состоит в низкой скорости вычислений, в связи с высокой сложностью вычисления времени расписания, равной $o(m*n^2)$. Предлагается другой способ кодирования. Вектор положения частицы делится на 2 участка размерности n (количество клиентов) и $m+1$ (количество агентов +1) соответственно, которые задают веса сортировки.



Первый участок задает порядок обхода клиентов, второй задает деление расписания по агентам. Таким образом, размерность вещественного пространства в алгоритме PSO сокращается с $2*n*m$ до $m+n+1$.

Численный эксперимент. В ходе проведения численного эксперимента было решено порядка 150 примеров, которые впоследствии были разбиты на классы в зависимости от размерности исходных данных и количества агентов.

Анализ вычислительного эксперимента показал в среднем на исследуемых тестовых примерах улучшение значения целевой функции методом роя частиц по сравнению с жадным алгоритмом: среднее улучшение по всем примерам составляет 15,98% (наименьшее – 0,11%, наибольшее – 89,51%).

Использование метода роя частиц в поставленной задаче составления расписания позволило в целом получать более эффективные расписания движения агентов (транспортных средств), чем это возможно при помощи жадного алгоритма.

Метод роя частиц удобен тем, что позволяет находить экстремум любой, даже очень сложной, функции, без знания ее структуры. Поскольку метод роя частиц является стохастическим методом, он не гарантирует нахождение глобального минимума функции, однако настройка метода дает возможность улучшить его эффективность для конкретной функции.

Литература

1. Рагулина Е.В. Метод роя частиц в задачах составления расписания// Сб. ст. седьмой Всероссийск. зимн. шк.-семинара аспирантов и молодых ученых «Актуальные проблемы науки и техники», 14–16 февраля 2012. Информационные и инфокоммуникационные технологии. Т. 3. – Уфа: УГАТУ, 2012. – с. 326 – 329
2. Зайцев А.А., Курейчик В.В., Полупанов А.А. Обзор эволюционных методов оптимизации на основе роевого интеллекта // Изв. ЮФУ. Технические науки, 2010. – № 12. – с. 7 – 12. – Библиогр.: с. 10 (17 назв.)
3. Luke S. Essentials of Metaheuristics. – 1st ed. – San Francisco (Calif): Creative Commons., 2009. – 237 p.: ill; Bibliogr: p. 214 – 216