МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Юго-Западный государственный университет

(ЮЗГУ)

Кафедра вычислительной техники

МЕТОД РОЯ ЧАСТИЦ. МЕТОД СЛУЧАЙНЫХ БЛУЖДАНИЙ

Методические указания для самостоятельной работы

для студентов направления подготовки 09.03.01

Курск 2016

УДК 621.3
Составитель: Э.И. Ватутин

Рецензент
Кандидат технических наук, доцент *А.Г. Спеваков*

Метод роя частиц. Метод случайных блужданий: методические указания для самостоятельной работы по дисциплине «Основы комбинаторной оптимизации» / Юго-Зап. гос. ун-т; сост.: Э.И. Ватутин; Курск, 2016. 7 с.

Излагаются методические рекомендации к выполнения самостоятельной работы при освоении методов дискретной комбинаторной оптимизации.

Предназначены для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника».

Текст печатается в авторской редакции

Подписано в печать _____. Формат 60х84 1/16.
Усл. печ. л. Уч. – изд.л. Тираж 30 экз. Заказ . Бесплатно.
Юго-Западный государственный университет
305040, Курск, ул. 50 лет Октября, 94.

Содержание

Метод случайных блужданий

Одним из методов является метод случайных блужданий. Он активно применяется в ряде разделов физики (например, для описания броуновского движения, диффузии, квантовой теории поля), экономики (например, для попытки описания динамики фондовых индексов), биологии (например, при описании движений глаз, полетов птиц) и др. В простейшем одномерном случае он может быть представлен в следующем виде

$$x_n = x_0 + \sum_{i=1}^{n} r_i \,, \tag{1}$$

где x_0 – начальное значение параметра, описывающего состояние системы; x_n – значение параметра в n-й дискретный момент времени; r_i – некоторый ряд случайных величин с заданным законом распределения. Для многомерного случая вместо одиночного значения параметра в рассмотрение вводится вектор параметров $X_t = \left[x_t^{(1)}, x_t^{(2)}, ..., x_t^{(N)}\right]^{\mathrm{T}}$, что не меняет в общем случае смысла формулы (1).

При решении задач непрерывной оптимизации вектор X_t задает значения аргументов целевой функции $f\left(X_t\right)$, которые меняются случайным образом по мере выполнения итерационного процесса, реализуя траекторию движения некоторой точки в N-мерном пространстве по аналогии, например, с броуновским движением частиц. При этом на каждой итерации $0 \le t \le C_{\max}$, где $C_{\max}$ – заданное число итераций, определяется значение целевой функции $f\left(X_t\right)$ и запоминается рекорд $f^+ = f\left(X^+\right)$ – минимальное или максимальное (в зависимости от постановки задачи) значение целевой функции f^+ и соответствующие ему значения вектора параметров X^+, полученные на одной из итераций. После выполнения заданного числа итераций $C_{\max}$ указанный рекорд полагается результатом работы метода.

При решении задач дискретной оптимизации указанный подход, базирующийся на формуле (1), напрямую неприменим. Вместо этого при переходе от решения X_t к решению X_{t+1} могут быть использованы модифицирующие операции [2, 7–9], специфичные для решаемой задачи (аналогичные модифицирующие операции используется в методе имитации отжига [7, 10], в генетическом методе в рамках оператора мутации [8, 11] и в методе пчелиной колонии при выполнении разведки пространства «вокруг» заданного решения [9, 12, 13]). При этом одна итерация метода случайных блужданий эквивалентна применения одной модифицирующей операции, выбранной случайным образом.

С целью апробации метода случайных блужданий может быть выбрана известная задача поиска кратчайшего пути кратчайшего пути $P(G) = \left[a_{i_1} = a_{нач}, a_{i_2}, ..., a_{i_Q} = a_{кон}\right]$ в графе $G = \langle A, V \rangle$, где $A = \{a_1, a_2, ..., a_N\}$ – множество вершин, $|A| = N$ – число вершин (размерность задачи), $V = \{v_1, v_2, ..., v_M\} \subseteq A \times A$ – множество дуг, $|V| = M$ – число дуг, $v_i = \left(a_{нач}^{(i)}, a_{кон}^{(i)}\right)$, $a_{нач}^{(i)} \in A$, $a_{кон}^{(i)} \in A$, причем дуги взвешены значением длины $L\left(v_i\right) > 0$, $i = \overline{1, M}$. Целевой функцией в указанной задаче является длина пути $L(P) = \sum_{j=1}^{Q-1} L\left(a_{i_j}, a_{i_{j+1}}\right) \to \min$, в качестве ограничения выступает плотность графа $d(G) = \dfrac{M}{N(N-1)} \in [0; 1]$, т.к. для графов малой плотности большое количество решений оказываются запрещенными. С целью выполнения объективного сравнения качества

решений необходимо организовать вычислительный эксперимент, целью которого являлось получение выборки $\Lambda = \{G_1, G_2, ..., G_K\}$ графов с псевдослучайной структурой и заданными параметрами *N* и *d*, с последующей оценкой усредненного качества решений $\bar{L} = \frac{1}{K}\sum_{i=1}^{K} L_i$, где $L_i = L\left(P\left(G_i\right)\right)$ – оценка качества решения (длины пути), найденного в графе G_i одним из методов. Указанная задача имеет точное решение, получаемое полиномиально с использованием алгоритмы Дейкстры, что позволяет использовать ее в качестве тестовой при оценке качества решений эвристических методов.

С учетом приведенных обозначений обобщенный алгоритм, соответствующий методу случайных блужданий в задаче поиска кратчайшего пути, может быть представлен в следующем виде.

1. (инициализация) Задать номер итерации $i := 1$, текущий путь $P := [a_{нач}, a_{кон}]$, лучший путь $P^+ := [a_{нач}, a_{кон}]$.
2. Произвести случайную модификацию текущего пути *P* путем применения к нему одной из модифицирующих операций.
3. При необходимости привести решение к корректному путем удаления циклов [2, 8].
4. Если $L(P) < L\left(P^+\right)$, положить $P^+ := P$.
5. Произвести модификацию номера итерации $i := i + 1$.
6. Если $i < C_{\max}$, перейти к п. 2.
7. Конец алгоритма.

В качестве начального решения используется путь, соединяющий заданные начальную и конечную вершины. В качестве модифицирующих операций в текущей задаче используются операции добавления вершины в путь, удаления вершины из пути и замены одной из вершин пути на другую вершину, не входящую в его состав. После применения некоторых модифицирующих операций может потребоваться процедура удаления циклов, которые запрещены по условиям решаемой задачи. При выборе одной из модифицирующих операций на этапе 2 алгоритма могут быть использованы различные стратегии. Простейшая из них подразумевает равновероятный выбор между одной из трех модифицирующих операций (в дальнейшем будем обозначать ее как RW1, сокр. от Random Walks version 1). Следуя более сложной стратегии, вероятности p_1, p_2, p_3 выбора модифицирующих операций можно тонко настроить в ходе метаоптимизации (в дальнейшем будем обозначать данную версию метода как RW2.

В рассмотренной выше дискретной постановке метод случайных блужданий в достаточной степени схож с методом случайного перебора (англ. Random Search, сокр. RS): в методе случайного перебора равновероятно выбирается следующая вершина пути при его последовательном формировании, а в методе случайных блужданий – одна из модифицирующих операций. Также метод случайных блужданий имеет достаточную степень сходства с методом имитации отжига при начальных больших значениях температуры, когда принимаются модификации текущего решения, дающие как улучшение, так и ухудшение качества решения.

Метод роя частиц

Одним из методов является метод роя частиц (англ. Particle Swarm Optimization, сокр. PSO), предложенный Эберхартом и Кеннеди в 1995 г. Он базируется на имитации группового поведения конечного множества агентов (например, комаров, рыб или птиц в живой природе). При решении оптимизационных задач вида $f(X) \to \min$, где f – целевая

функция, $X=[x_1, x_2, ..., x_N]$ – ее аргументы, текущее положение каждого *i*-го агента (частицы), $i=\overline{1, Z}$, Z – размер колонии (роя), характеризуется вектором с координатами $X_i=\left[x_1^{(i)}, x_2^{(i)}, ..., x_N^{(i)}\right]$. При этом частица имеет текущую скорость $V_i=\left[v_1^{(i)}, v_2^{(i)}, ..., v_N^{(i)}\right]$, а ее положение на *t*-м шаге алгоритма определяется как $X_i^{(t)}=X_i^{(t-1)}+V_i^{(t-1)}$. Скорость частицы на *t*-м шаге определяется как

$$V_i^{(t)}=\alpha V_i^{(t-1)}+\beta r_k\otimes\left(X_i^*-X_i\right)+\gamma r_{k+1}\otimes\left(X^{**}-X_i\right), \qquad (2)$$

где «$\otimes$» – обозначение прямого (покомпонентного) произведения векторов:

$$X\otimes Y=[x_1, x_2, ..., x_N]\otimes[y_1, y_2, ..., y_N]=[x_1y_1, x_2y_2, ..., x_Ny_N];$$

$X_i^*=\arg\min\limits_{\tau=\overline{0, t}} f\left(X_i^{(\tau)}\right)$ – наилучшее положение *i*-й частицы роя за время его движения (локальный рекорд); $X^{**}=\arg\min\limits_{i=\overline{1, Z}} f\left(X_i^*\right)$ – наилучшее положение среди всех частиц роя (глобальный рекорд); α, β, γ – настроечные параметры, имеющие смысл движения частицы по инерции и притяжения к локальному и глобальному рекорду соответственно; $r_k\in[0; 1]$ – очередное псевдослучайное число.

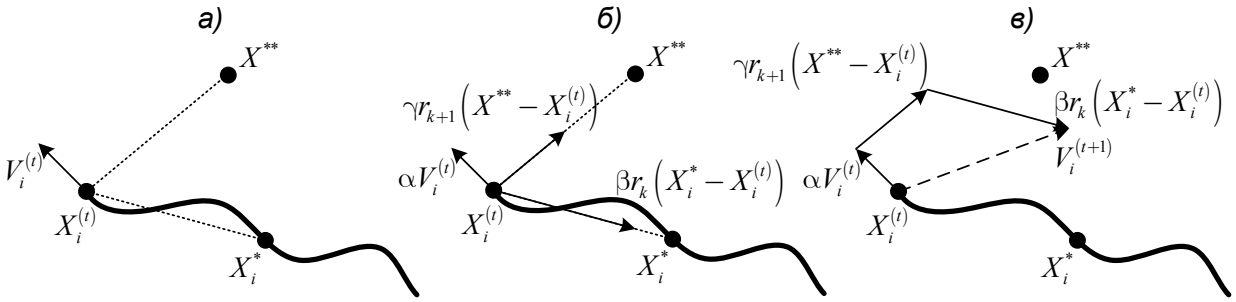

Рис. 1. Схематичное изображение, поясняющее движения частицы роя: жирная кривая – траектория движения частицы, вектор ее текущей скорости $V_i^{(t)}$ и текущее положение $X_i^{(t)}$, штриховые линии – направления (вектора) в сторону глобального и локального рекордов (а); компоненты вектора скорости, соответствующие инерции $\alpha V_i^{(t)}$ и движению в направлении локального $\beta r_k\left(X_i^*-X_i^{(t)}\right)$ и глобального $\gamma r_{k+1}\left(X^{**}-X_i^{(t)}\right)$ рекордов (б); результирующий вектор скорости частицы на следующем шаге $V_i^{(t+1)}$ (в)

Работа метода производится за заданное число итераций $C_{\max}$, результирующим решением считается глобальный рекорд X^{**}. Приведенное описание метода является базовым, в некоторых случаях [4] применяются многороевая или мультистарт стратегии, различные топологии соседства частиц, механизмы локального (градиентного поиска) с целью улучшения текущего положения частиц и пр.

Указанная стратегия движения частиц в непрерывном пространстве параметров целевой функции с успехом применяется при решении ряда задач непрерывной оптимизации [5–9], однако в дискретных задачах она напрямую неприменима ввиду ряда сложностей с понятием скорости движения частицы в дискретном пространстве (наиболее ярким примером являются задачи, в которых аргументы могут принимать только булевы значения, а вектора скоростей $V_i\in\{-1, 1\}$ теряют смысл). Для их решения приведенные выше формулы не подходят и применяются специализированные подходы.

Первое направление подходов оперирует движением частиц роя в непрерывном пространстве $\mathbb{R}_1$ с последующим отображением координат частиц в дискретное пространство $\mathbb{R}_2$ параметров целевой функции $(\mathbb{R}_1 \to \mathbb{R}_2)$. Указанное отображение может быть выполнено различными способами, например, с использованием сигмоидальной функции $\sigma(y) = \frac{1}{1+e^{-y}}$, $0 < \sigma(y) < 1$, $y \in \mathbb{R}_1$, где $\sigma(y)$ определяет вероятность принадлежности дискретного значения $x \in \mathbb{R}_2$ одному из бинарных значений

$$x = \begin{cases} 1, r_k < \sigma(y); \\ 0, \textit{иначе}. \end{cases} \tag{3}$$

Если по условию задачи параметры целевой функции должны принимать дискретные значения из некоторого множества значений $\{v_1, v_2, ..., v_Q\}$, то формула (3) может быть преобразована к виду

$$x = \begin{cases} v_1, 0 < r_k\sigma(y) \le \frac{1}{Q}; \\ v_2, \frac{1}{Q} < r_k\sigma(y) \le \frac{2}{Q}; \\ ... \\ v_Q, \frac{Q-1}{Q} < r_k\sigma(y) \le 1, \end{cases} \tag{4}$$

что эквивалентно выбору одного из значений с использованием правила «рулетки» [1] пропорционально значению $\sigma(y)$.

Более простой в вычислительном плане является стратегия округления значения y или $\sigma(y)$ до ближайшего значения без рандомизации

$$i = \lfloor yQ \rfloor + 1 \tag{5}$$

или

$$i = \lfloor \sigma(y)Q \rfloor + 1, \tag{6}$$

где $\lfloor x \rfloor$ – обозначение операции округления вниз (усечения), $x = v_i$ (при использовании более простой формулы (4) необходимо следить, чтобы значение y не выходило за пределы отрезка $[0;1]$).

Второе направление подходов отказывается от использования понятия скоростей и координат в непрерывном пространстве, оперируя конкретными решениями задачи с учетом ее специфики и производя модификацию решений X с целью получения решения X', более похожего на требуемое Y (в методе роя частиц – на глобальный или локальный рекорд). При этом вместо координат частиц роя используются конкретные решения, закодированные в терминах решаемой задачи, а в качестве скоростей выступают вероятности p_i применения той или иной модифицирующей операции o_i, специфичной для решаемой задачи, с целью получения нового положения решения $X' = o_i(X)$ в дискретном пространстве. При этом, с целью сохранения общей идеи метода роя частиц, должно выполняться условие $d(X', Y) < d(X, Y)$, где d – «расстояние» между парой решений в дискретном пространстве (аналог расстояний Хэмминга, Левенштейна, расстояния между графами при проверке изоморфизма и т.п.).

Рассмотрим применение метода роя частиц на примере решения задачи поиска кратчайшего пути кратчайшего пути $P(G) = \left[a_{i_1} = a_{нач}, a_{i_2}, ..., a_{i_Q} = a_{кон}\right]$ в графе

$G=\langle A,V\rangle$, где $A=\{a_1,a_2,...,a_N\}$ – множество вершин, $|A|=N$ – число вершин (размерность задачи), $V=\{v_1,v_2,...,v_M\}\subseteq A\times A$ – множество дуг, $|V|=M$ – число дуг, $v_i=\left(a_{нач}^{(i)},a_{кон}^{(i)}\right)$, $a_{нач}^{(i)}\in A$, $a_{кон}^{(i)}\in A$, причем дуги взвешены значением длины $L(v_i)>0$, $i=\overline{1,M}$. Целевой функцией в указанной задаче является длина пути $L(P)=\sum_{j=1}^{Q-1}L\left(a_{i_j},a_{i_{j+1}}\right)\to\min$, в качестве ограничения выступает плотность графа $d(G)=\frac{M}{N(N-1)}\in[0;1]$, т.к. для графов малой плотности большое количество решений оказываются запрещенными. Указанная задача имеет точное решение, получаемое полиномиально с использованием алгоритмы Дейкстры за квадратичное время, что позволяет использовать ее в качестве тестовой при оценке качества решений эвристических методов.

Согласно подходу, базирующемуся на отображении непрерывного пространства в дискретное, решение можно представить в виде упорядоченного набора (массива) позиций $B=[b_1,b_2,...,b_U]$, $2\le U\le N$ пути *P*, каждой из которых соответствует одна из вершин $a_i\in A$, $i=\overline{1,N}$ графа *G*, причем $b_1=a_{нач}$, $b_U=a_{кон}$. Соответствия между позициями b_i и вершинами a_j будем помечать значениями p_{ij}, имеющими смысл вероятностей присутствия *j*-й вершины в *i*-й позиции пути (аналогичный принцип используется в модифицированной версии алгоритма муравьиной колонии, базирующейся на использовании двудольного графа, в котором аналогичные соответствия отмечались феромоном τ_{ij}). Над вероятностями p_{ij} возможно выполнение действий по формуле (2), рассматривая их как текущие координаты частицы в пространстве $\mathbb{R}_1$, причем их значения могут быть произвольными неотрицательными величинами. Для перехода от непрерывных значений вероятностей к дискретным значениям номеров вершин в пути будем считать, что в *i*-й позиции пути стоит *j*-я вершина, если $j=\arg\max_{k=\overline{1,N}}p_{ik}$ (в качестве другого способа выбора номера вершины может быть использован выбор вершин пропорционально вероятностям p_{ik} с использованием правила рулетки; выполненные вычислительные эксперименты не показали статистически значимого отличия для указанных способов определения номера вершины). При подобном построении пути в его составе возможно образование циклов, что недопустимо по условиям задачи (не приводит к сокращению длины пути), поэтому при выборе очередной вершины в качестве кандидата на заполнение *i*-й позиции необходима либо проверка того, что выбранная вершина не была использована в одной из предыдущих позиций b_k, $1\le k<i$, либо процедура удаления циклов после формирования пути. Схематично процесс получения пути по набору вероятностей p_{ik} изображен на рис. 2.

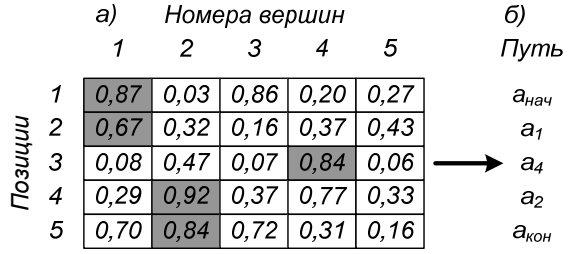

Рис. 2. Пример, иллюстрирующий получение пути (б) отталкиваясь от набора вероятностей p_{ik} (а)

Алгоритм, соответствующий приведенному выше описанию метода (сокр. PSO1), приведен ниже.

1. (инициализация) Инициализировать глобальный рекорд $P^{**} := [],\ L\left(P^{**}\right) := \infty$.
2. (инициализация роя) Для всех частиц роя $i = \overline{1, Z}$:
   2.1. Положить начальные значения вероятностей $p_{jl}^{(i)} := p_0\left(2r_k - 1\right)$ (случайное значение в диапазоне $\left[-p_0; p_0\right]$), $j, l = \overline{1, N}$, начальные значения скоростей $v_{jl}^{(i)} := v_0\left(2r_{k+1} - 1\right)$ (случайное значение в диапазоне $\left[-v_0; v_0\right]$), где p_0 и v_0 – настроечные параметры.
   2.2. Определить путь P_i, соответствующий набору вероятностей $p_{jl}^{(i)}$.
   2.3. Инициализировать локальный рекорд $P_i^* := P_i,\ L\left(P_i^*\right) := L\left(P_i\right)$.
   2.4. (обновление глобального рекорда) Если $L\left(P_i\right) < L\left(P^{**}\right)$, положить $P^{**} := P_i$, $L\left(P^{**}\right) := L\left(P_i\right)$.
3. (движение роя) Для всех итераций $C = \overline{1, C_{\max}}$:
   3.1. Для всех частиц роя $i = \overline{1, Z}$:
      3.1.1. Для всех позиций пути $j = \overline{1, N}$:
         3.1.1.1. Для всех номеров вершин $l = \overline{1, N}$:
            3.1.1.1.1. (инерционная компонента) Положить $v_{jl}^{(i)} := \alpha v_{jl}^{(i)}$.
            3.1.1.1.2. (движение в сторону локального рекорда) Если в позиции b_l пути P_i^* стоит вершина a_j, положить $v_{jl}^{(i)} := v_{jl}^{(i)} + \beta r_k$.
            3.1.1.1.3. (движение в сторону глобального рекорда) Если в позиции b_l пути P^{**} стоит вершина a_j, положить $v_{jl}^{(i)} := v_{jl}^{(i)} + \gamma r_k$.
            3.1.1.1.4. (корректировка вероятности) Положить $p_{jl}^{(i)} := p_{jl}^{(i)} + v_{jl}^{(i)}$.
      3.1.2. Определить путь P_i, соответствующий набору вероятностей $p_{jl}^{(i)}$.
      3.1.3. (обновление локального рекорда) Если $L\left(P_i\right) < L\left(P_i^*\right)$, положить $P_i^* := P_i$, $L\left(P_i^*\right) := L\left(P_i\right)$.
      3.1.4. (обновление глобального рекорда) Если $L\left(P_i\right) < L\left(P^{**}\right)$, положить $P^{**} := P_i$, $L\left(P^{**}\right) := L\left(P_i\right)$.
4. Положить результирующее решение равным глобальному рекорду P^{**}.
5. Конец алгоритма.

В приведенном описании алгоритма используется процедура определения пути P_i, соответствующего набору вероятностей $p_{jl}^{(i)}$, алгоритм которой приведен ниже.

1. (инициализация) Положить $P_i := \left[a_{нач}\right]$.
2. Для всех позиций пути $j = \overline{2, N}$:
   2.1. Выбрать номер вершины $j = \arg\max\limits_{\substack{k=\overline{1, N},\\ a_k \notin P_i}} p_{jk}$.
   2.2. Положить $P_i := P_i \odot \left[a_j\right]$ («$\odot$» – обозначение операции конкатенации путей).

2.3. Если $a_j = a_{кон}$, перейти к п. 3.

3. Конец алгоритма.

В приведенном описании алгоритма процедуры определения пути по набору вероятностей использован выбор очередной вершины по максимуму вероятности с запретом на повторное использование вершин, уже добавленных в путь. Как уже было отмечено выше, возможны и другие реализации процедуры. Так выбор вершин можно производить по правилу рулетки для набора вероятностей $\left[p_{j,1}, p_{j,2}, ..., p_{j,N}\right]$, а при построении пути можно не проверять повторяемость вершин, добавляя в конце процедуру удаления циклов.

Этап инициализации (присваивание начальных значений вероятностей $p_{jl}^{(i)}$) алгоритма выполняется за время порядка $O\left(N^2 Z\right)$, алгоритм формирования пути по набору вероятностей для выбранной частицы роя имеет асимптотическую временную сложность $O\left(N^2\right)$, этап движения роя характеризуется временной сложностью $O\left(N^2 Z C_{\max}\right)$. Таким образом, асимптотическая временная сложность алгоритма в целом составляет $O\left(N^2 Z + N^2 Z C_{\max}\right) \simeq O\left(N^2 Z C_{\max}\right)$. Для работы алгоритма требуется хранения массива вероятностей $p_{jl}^{(i)}$, на что необходимо $O\left(N^2 Z\right)$ ячеек памяти. Хранение текущих путей и локальных рекордов требует порядка $O\left(NZ\right)$ памяти. Таким образом, емкостная сложность алгоритма составляет $O\left(N^2 Z\right)$, что больше, нежели чем в рассмотренных ниже вариантах реализации PSO2 и PSO3. При практической программной реализации следствием указанной асимптотической емкостной сложности являются существенные затраты оперативной памяти (до 700 МБ RAM при обработке графов из $N = 500$ вершин).

Еще один вариант реализации метода роя частиц (сокр. PSO2) может быть разработан на базе округления значения координат текущей частицы с использованием (4). При этом не используются вероятности $p_{jl}^{(i)}$, вместо которых применяются координаты $x_j^{(i)}$ положения частицы в *N*-мерном пространстве. Принцип вычисления скоростей и корректировки координат соответствует (1) с наложением дополнительного условия $1 \le x_j^{(i)} \le N$ (при его несоблюдении производится принудительный возврат координаты частицы в обозначенную разрешенную область значений). Пример отображения непрерывного пространства в дискретное приведен ниже:

$$X = [2{,}03;\ 2{,}73;\ 6{,}72;\ 3{,}19;\ 1{,}62;\ 3{,}72;\ 4{,}26;\ 0{,}82] \Rightarrow$$

$$\Rightarrow P = \left[a_{нач}, \underbrace{a_3, a_7, a_3}_{цикл\,1}, a_2, \underbrace{a_4, a_4}_{цикл\,2}, a_{кон}\right] \Rightarrow$$

$$P' = \left[a_{нач}, a_3, a_2, a_4, a_{кон}\right].$$

Алгоритм, соответствующий указанной реализации метода роя частиц, приведен ниже.

1. (инициализация) Инициализировать глобальный рекорд $P^{**} := [],\ L\left(P^{**}\right) := \infty$.

2. (инициализация роя) Для всех частиц роя $i = \overline{1, Z}$:

   2.1. Положить начальные значения координат $x_j^{(i)} := r_k\left(N-1\right)+1$ (случайное значение в диапазоне $\left[1; N\right]$), $j = \overline{1, N}$, начальные значения скоростей $v_j^{(i)} := v_0\left(2r_{k+1}-1\right)$ (случайное значение в диапазоне $\left[-v_0; v_0\right]$), где v_0 – настроечный параметр.

2.2. Определить путь $P_i = [b_1, b_2, ..., b_U]$, соответствующий набору координат $x_j^{(i)}$, полагая $b_j = \begin{cases} a_{нач}, j = 1, \\ \left\lfloor x_j^{(i)} \right\rfloor + 1, j > 1 \end{cases}$ до тех пор, пока $b_U \neq a_{кон}$.

2.3. Если $\forall b_j \in P_i$: $b_j \neq a_{кон}$, положить $b_N = a_{кон}$.

2.4. Если в пути P_i присутствуют циклы, применить процедуру их удаления [1, 26].

2.5. Инициализировать локальный рекорд $P_i^* := P_i$, $L\left(P_i^*\right) := L\left(P_i\right)$.

2.6. (обновление глобального рекорда) Если $L\left(P_i\right) < L\left(P^{**}\right)$, положить $P^{**} := P_i$, $L\left(P^{**}\right) := L\left(P_i\right)$.

3. (движение роя) Для всех итераций $C = \overline{1, C_{\max}}$:

3.1. Для всех частиц роя $i = \overline{1, Z}$:

3.1.1. Для всех позиций пути $j = \overline{1, N}$:

3.1.1.1. (инерционная компонента) Положить $v_j^{(i)} := \alpha v_j^{(i)}$.

3.1.1.2. (движение в сторону локального рекорда) $v_j^{(i)} := v_j^{(i)} + \beta r_k \left(x_j^{(i)*} - x_j^{(i)}\right)$, где $x_j^{(i)*}$ – номер j-й вершины в пути P_i^*.

3.1.1.3. (движение в сторону глобального рекорда) $v_j^{(i)} := v_j^{(i)} + \gamma r_k \left(x_j^{**} - x_j^{(i)}\right)$, где x_j^{**} – номер j-й вершины в пути P^{**}.

3.1.1.4. (корректировка координат) Положить $x_j^{(i)} := x_j^{(i)} + v_j^{(i)}$.

3.1.1.5. (проверка координат) Положить $x_j^{(i)} := \begin{cases} 0, x_j^{(i)} < 0; \\ N-1, x_j^{(i)} \geq N; \\ x_j^{(i)} \textit{ иначе.} \end{cases}$

3.1.2. Определить путь P_i, соответствующий набору координат $x_j^{(i)}$ (аналогично п. 2.2–2.4).

3.1.3. (обновление локального рекорда) Если $L\left(P_i\right) < L\left(P_i^*\right)$, положить $P_i^* := P_i$, $L\left(P_i^*\right) := L\left(P_i\right)$.

3.1.4. (обновление глобального рекорда) Если $L\left(P_i\right) < L\left(P^{**}\right)$, положить $P^{**} := P_i$, $L\left(P^{**}\right) := L\left(P_i\right)$.

4. Положить результирующее решение равным глобальному рекорду P^{**}.

5. Конец алгоритма.

В данном алгоритме процедура построения пути из координат частиц роя отличается от предыдущей. При ее использовании возможно как возникновение циклов в составе формируемого пути, которые требуется удалять, так и наличие ситуации, в которой результирующая вершина не будет присутствовать в составе пути и ее необходимо принудительно добавить.

На этапе инициализации алгоритма необходимо присваивание начальных значений координат и скоростей, на что требуется порядка $O(NZ)$ времени. Процедура удаления циклов состоит из двух частей: для поиска цикла необходимо выяснение повторений вершин в пути, на что необходимо $O\left(N^2\right)$ времени, удаление цикла производится за

$O(N)$ шагов. Ее асимптотическая временная сложность составляет $O\left(\left(N^2+N\right)A\right)\simeq O\left(N^2A\right)$, где A – число циклов. В большинстве случаев $A\ll N$, поэтому можно положить, что $O\left(N^2A\right)\simeq O\left(N^2\right)$. На движение роя (обновление координат частиц) необходимо $O\left(NZC_{\max}\right)$ шагов. Таким образом, асимптотическая временная сложность алгоритма составляет $O\left(Z\left(N+N^2\right)+ZC_{\max}\left(N+N^2\right)\right)\simeq O\left(N^2ZC_{\max}\right)$. На хранение текущих путей и локальных рекордов, как и в предыдущем алгоритме, требуется $O(ZN)$ ячеек памяти. Хранение координат частиц и скоростей требует $O(ZN)$ памяти. Таким образом, емкостная сложность алгоритма составляет $O(ZN)$, что в N раз меньше алгоритма PSO1.

Еще одна модификация метода роя частиц (сокр. PSO3) в рассматриваемой задаче может быть построена путем отказа от использования непрерывного пространства и движения в нем с использованием координат $x_j^{(i)}$ и скоростей $v_j^{(i)}$ путем перехода к дискретным решениям (путям в графе в рассматриваемой задаче). При этом необходима реализация специализированного оператора (процедуры), который будет осуществлять движение от заданного пути P_1 к пути P_2 в дискретном пространстве решений $\mathbb{R}_2$ с целью уменьшения расстояния (метрики) $d\left(P_1,P_2\right)$ между указанными путями (в окончательном случае $d\left(P_1,P_2\right)=0$ при $P_1=P_2$). Его алгоритм приведен ниже.

1. Если $U\left(P_1\right)<U\left(P_2\right)$, где $U(P)$ – число вершин в составе пути P, добавить в случайную позицию $j=\left\lfloor r_kU\left(P_1\right)\right\rfloor+1$ пути P_1 еще не использованную вершину a_i, такую что $a_i\notin P_1$: $P_1:=\left[b_1,...,b_{j-1},a_i,b_j,...,b_U\right]$.
2. Если $U\left(P_1\right)>U\left(P_2\right)$, удалить случайную вершину в позиции $j=\left\lfloor r_kU\left(P_1\right)\right\rfloor+1$ пути P_1: $P_1:=\left[b_1,...,b_{j-1},b_{j+1},...,b_U\right]$.
3. Если $U\left(P_1\right)=U\left(P_2\right)$, определить множество номеров позиций $S=\left\{b_{i_1},b_{i_2},...,b_{i_Q}\right\}$, в которых пути P_1 и P_2 не совпадают; выбрать одну из позиций случайным образом $j=\left\lfloor r_kQ\right\rfloor+1$; заменить вершину пути P_1 в позиции b_{i_j} на вершину в аналогичной позиции пути P_2.
4. Конец алгоритма.

В результате применения указанного алгоритма сперва происходит выравнивание длин путей P_1 и P_2 путем добавления или удаления вершин пути P_1, а затем его вершины поочередно заменяются на вершины пути P_2, в результате чего после применения алгоритма C раз пути P_1 и P_2 оказываются равны.

Алгоритм метода роя частиц, базирующийся на движении в дискретном пространстве путем изменения решений, приведен ниже.

1. (инициализация) Инициализировать глобальный рекорд $P^{**}:=[]$, $L\left(P^{**}\right):=\infty$.
2. (инициализация роя) Для всех частиц роя $i=\overline{1,Z}$:

   2.1. Сформировать начальные решения P_i с использованием одного из методов.

   2.2. Инициализировать локальный рекорд $P_i^*:=P_i$, $L\left(P_i^*\right):=L\left(P_i\right)$.

2.3. (обновление глобального рекорда) Если $L(P_i) < L(P^{**})$, положить $P^{**} := P_i$, $L(P^{**}) := L(P_i)$.

3. (движение роя) Для всех итераций $C = \overline{1, C_{\max}}$:

3.1. Для всех частиц роя $i = \overline{1, Z}$:

3.1.1. Если $p^* r_k > p^{**} r_{k+1}$, где p^* и p^{**} – соответственно вероятности движения в сторону локального и глобального рекорда, осуществить модификацию пути P_i (движение частицы) в сторону локального рекорда P_i^*, в противном случае – в сторону глобального рекорда P^{**} с использованием рассмотренного выше алгоритма.

3.1.3. (обновление локального рекорда) Если $L(P_i) < L(P_i^*)$, положить $P_i^* := P_i$, $L(P_i^*) := L(P_i)$.

3.1.4. (обновление глобального рекорда) Если $L(P_i) < L(P^{**})$, положить $P^{**} := P_i$, $L(P^{**}) := L(P_i)$.

4. Положить результирующее решение равным глобальному рекорду P^{**}.

5. Конец алгоритма.

При программной реализации алгоритма для формирования начальных решений был использован метод случайного перебора с возвратами [1, 27] как обеспечивающий большой начальный разброс решений для диверсификации пространства поиска.

Временная сложность этапа инициализации приведенного алгоритма определяется сложностью метода, используемого для формирования начальных решений. В организованных экспериментах в качестве начальных решений были использованы решения, получаемые в результате выполнения одной итерации метода случайного перебора с возвратами, сложность которого составляет $O(N^2)$. Во время движения роя производится $O(ZC_{\max} \cdot e)$, где e – сложность модифицирующих операций. Операция добавления вершины выполняется за линейное время $O(N)$, удаление циклов, как уже было отмечено выше, характеризуется сложностью $O(N^2)$. Операция удаления вершины выполняется за линейное время $O(N)$. Операция замены одной из не совпадающих вершин пути выполняется за линейное время $O(N)$. Таким образом, асимптотическая временная сложность модификации текущего решения составляет $O(\max(N^2, N, N)) \simeq O(N^2)$, а сложность алгоритма в целом – $O(N^2 Z C_{\max})$. Несложно заметить, что асимтотическая временная сложность совпадает для всех реализации метода роя части. Емкостная сложность алгоритма складывается из затрат памяти на хранение текущих путей в составе каждой частицы, на хранение локальных и глобального рекордов, и составляет величину порядка $O(Z(N+N)+N) \simeq O(NZ)$. Емкостная сложность данной реализации алгоритма совпадает с реализацией PSO2 и в N раз меньше алгоритма PSO1.

Для рассмотренных выше трех вариантов реализации метода роя частиц необходимо разработать соответствующие программные реализации, интегрированные в состав соответствующего расчетного модуля. С их помощью необходимо организовать ряд вычислительных экспериментов, базирующихся на получении выборки $\Lambda = \{G_1, G_2, ..., G_K\}$ графов с псевдослучайной структурой и заданными параметрами N и d, с последующей оценкой усредненного качества решений $\overline{L} = \frac{1}{K} \sum_{i=1}^{K} L_i$, где

$L_i = L\left(P\left(G_i\right)\right)$ – оценка качества решения (длины пути), найденного в графе G_i одним из методов за $C_{\max} = 1\,000$ итераций, результаты которых рассмотрены ниже. Также необходимо произвести настройку параметров методов.

Библиографический список

1. Ватутин Э.И. и др. Основы дискретной комбинаторной оптимизации. М.: АРГАМАК-МЕДИА, 2016. 270 с.