УДК 519.8
DOI: 10.32626/2308-5878.2019-19.35-41

И. В. Козин, д-р физ.-мат. наук,
С. И. Полюга, канд. физ.-мат. наук,
В. И. Сардак, аспирант

Запорожский национальный университет, г. Запорожье

ФРАГМЕНТАРНАЯ МОДЕЛЬ РАЗМЕЩЕНИЯ ПРОИЗВОДСТВА

Рассмотрена двумерная задача размещения производственных объектов в дискретной постановке. Показано, что дискретная задача размещения производства сводится к задаче покрытия графа звездами и имеет фрагментарную структуру. Для поиска приближенного решения задачи предложены модификация эволюционного алгоритма на перестановках с геометрическим оператором кроссовера и алгоритм муравьиной колонии на фрагментарной структуре. Приводятся результаты численного эксперимента по сравнению алгоритмов.

Ключевые слова: *фрагментарная модель, задача размещения производства, эволюционный алгоритм, геометрический кроссовер, алгоритм муравьиной колонии.*

Введение. Задача размещения производственных объектов часто возникает в экономике, производстве, строительстве. В последнее время, учитывая децентрализацию управления страны, эта задача актуальна при выборе мест объектов коллективного пользования в территориальных образованиях. Существует много различных постановок задачи [1–3]. Несмотря на относительную простоту описания, задача размещения производства сложная практически в любой из постановок. Существует множество подходов к поиску оптимальных решений задачи [3]. Но для больших размерностей задача, как правило, решается приближенно с помощью метаэвристик различного вида. Большое количество дополнительных условий в конкретных постановках приводит к необходимости использовать вероятностные методы и эвристические процедуры.

Постановка задачи. Рассмотрим одну из наиболее простых дискретных постановок задачи размещения производства на эвклидовой плоскости. Имеется конечное множество точек плоскости, каждая из которых характеризуется двумя эвклидовыми координатами. Каждая из точек представляет собой или будущего потребителя продуктов производства, или возможную точку размещения производства. Все объекты в этой постановке считаются точечными. В качестве целевой функции задачи принимаются затраты на открытие производства в заданной точке либо за доставку продукции к определенному клиенту. Все точки-потребители распределяются между точками производства таким образом, что:

а) каждый потребитель приписан одной и только одной точке производства;
б) распределение потребителей неизменно;
в) каждой точке приписана стоимость открытия производства в этой точке;
г) стоимость доставки продукции из точки производства к точке — потребителю прямо пропорциональна эвклидову расстоянию между точками.

Таким образом, каждое допустимое решение задачи может быть представлено на плоскости графом, который имеет вид объединения непересекающихся в вершинах звезд. Центром каждой звезды являются точки производства, а лучи связывают точки производства с точками — потребителями продукта (рисунок).

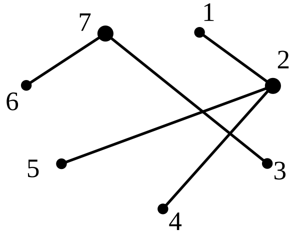

Рисунок. *Одно из допустимых решений задачи размещения производства*

Покажем, что рассматриваемая задача может рассматриваться как задача на фрагментарной структуре и, соответственно, к ней могут быть применены универсальные алгоритмы на фрагментарной структуре, в основе которых лежат метаэвристики [4].

Фрагментарная структура. В соответствии с [5] фрагментарной структурой (X,E) на конечном множестве X будем называть семейство его подмножеств $E=\left\{E_1,E_2,...,E_n\right\}$ такое, что $\forall E_i\in E$, $E\neq\varnothing$ $\exists e\in E_i : E_i\setminus\left\{e\right\}\in E$.

Элементы из множества E будем называть допустимыми фрагментами. Таким образом, для любого допустимого фрагмента E_i существует нумерация его элементов $E_i=\{e_{i1},e_{i2},...e_{is_i}\}$ такая, что $\forall k=1,2,...s_i$ $\{e_{i1},e_{i2},...e_{ik}\}\in E$. Элементарным фрагментом будем называть допустимый фрагмент, состоящий из одного элемента. Максимальный фрагмент — допустимый фрагмент, который не является подмножеством никакого другого фрагмента.

Максимальный фрагмент может быть построен с помощью следующего «жадного» алгоритма:

а) элементы множества X линейно упорядочиваются;
б) на начальном шаге выбирается пустое множество $X_0 = \varnothing$;
в) на шаге с номером $k+1$ выбирается первый по порядку элемент $x \in X \setminus X_k$, такой, что $X_k \cup \{x\} \in E$;
г) алгоритм заканчивает работу, если на очередном шаге не удалось найти элемент $x \in X \setminus X_k$ с требуемым свойством.

Результат работы алгоритма определяется заданным линейным порядком на множестве X. Таким образом, любой максимальный фрагмент может быть описан некоторой перестановкой элементов множества X. Пусть $A \in E$. Условие для элемента $x \in X$, при котором $A \cup \{x\} \in E$, будем называть условием присоединения элемента x.

Пусть теперь каждому фрагменту приписан вес, то есть задана функция $\rho : E \to R^1$. Будем предполагать, что функция ρ монотонна по включению (возрастающая или убывающая). Если $A, B \in E$ и $A \subseteq B$, то $\rho(A) \le (\ge) \rho(B)$.Задача оптимизации на фрагментарной структуре, это задача отыскания допустимого фрагмента максимального (минимального) веса. Очевидно, что для монотонных весов оптимальное решение будет являться максимальным фрагментом.

Фрагментарная модель. Покажем, что задача размещения производства в вышепредложенной постановке может быть представлена как задача оптимизации на фрагментарной структуре. В качестве множества элементарных фрагментов рассмотрим множество ребер полного графа с вершинами в заданных точках локации. Каждый допустимый фрагмент будем строить, соблюдая следующее условие присоединения. Очередное ребро присоединяется к выбранному набору ребер, если после присоединения полученный подграф представляет собой объединение непересекающихся в вершинах звезд. Если очередное ребро присоединить не удается, то переходим к следующему по порядку ребру. Алгоритм заканчивает работу, если список ребер исчерпан. Множества ребер, которые последовательно будут построены в результате работы такого алгоритма (множество E), образуют фрагментарную структуру. Центры полученных в результате работы алгоритма звезд и изолированные вершины (если в них локализованы получатели продукции) являются точками размещения производственных мощностей. Целевая функция задачи $F : E \to R^1$ — стоимость локализации организации производства в выбранных точках размещения производства плюс стоимость доставки продукции потребителям по лучам звезд. Очевидно, целевая функция является монотонной.

Любой максимальный фрагмент определяется заданным линейным порядком просмотра элементарных фрагментов. Этот порядок определяет результат работы фрагментарного алгоритма, который и построит требуемый максимальный фрагмент.

Каждый линейный порядок определяется некоторой перестановкой $s \in S_n$ укладываемых объектов (n — число ребер графа). Сопоставим каждой перестановке максимальный фрагмент, который ей порождается. Обозначим это отображение через $\varphi : S_n \to E$. Таким образом, имеет место естественная коммутативная диаграмма отображений

$$\begin{array}{ccc} S_n & & \\ \varphi \downarrow & \searrow F \circ \varphi , & \\ E \to R^1 & & \end{array}$$

которая превращает задачу оптимизации на фрагментарной структуре в задачу оптимизации на множестве перестановок. Причем любая перестановка является допустимой. Для больших значений n задача поиска оптимальной перестановки, как правило, является трудной. Предлагается использовать для поиска приближенных решений этой задачи варианты муравьиного и эволюционный алгоритмов на перестановках определенного вида [4].

Эволюционный алгоритм. Базовое множество X эволюционной модели — это множество $S_n = \{ i_1, i_2 ..., i_n \}$ всех перестановок чисел $1, 2, \dots, n$. Оператор построения начальной популяции выделяет произвольное подмножество заданной мощности Q из множества X.

Правило вычисления критерия селекции устроено следующим образом: по заданной перестановке фрагментов с помощью фрагментарного алгоритма строится максимальный допустимый фрагмент и вычисляется значение целевой функции задачи для этого фрагмента.

Опишем теперь оператор кроссовера. Пусть $U = (u_1, u_2, ..., u_n)$ и $V = (v_1, v_2, ..., v_n)$ — две произвольные перестановки. Перестановка-потомок строится следующим образом: последовательности U и V просматриваются в порядке следования элементов. На k-м шаге выбирается наименьший из первых элементов последовательностей и добавляется в новую перестановку-потомок. Затем этот элемент удаляется из двух последовательностей-родителей. Например, результатом кроссовера перестановок (2, 3, 6, 1, 7, 8, 4, 5) и (4, 6, 7, 1, 3, 2, 8, 5) будет перестановка (2, 3, 4, 6, 1, 2, 8, 5). В работе [5] показано, что определенный таким образом оператор кроссовера является геометрическим в инверсной метрике на перестановках [6]. Оператор мутации M выполняет случайную транспозицию в перестановке. Оператор селекции выбирает случайным образом набор пар из текущей популяции для последующего скрещивания.

Оператор эволюции упорядочивает элементы промежуточной популяции в последовательность по убыванию значения критерия селекции. В качестве новой текущей популяции выбираются первые Q элементов последовательности.

Обычное правило остановки — количество поколений достигло предельного значения. Лучшая по значению критерия селекции перестановка из последней построенной популяции определяет приближенное решение задачи.

Алгоритм муравьиной колонии [7]. Процедура вычисления оптимальной перестановки будет состоять из ряда циклов расчета. Каждый путь муравья между позициями 1, 2, ..., n будет определяться перестановкой $s = (i_1, i_2, ... i_n)$. Муравьи имеют собственную «память». У каждого муравья есть список уже посещенных позиций — список запретов. Обозначим $J_{i,k}^t$ список позиций, которые на цикле t необходимо посетить k-му муравью, находящемуся в позиции i.

Количество феромона в цикле с номером t при переходе из позиции i в позицию j определяется величиной $\tau_{ij}(t)$. На начальном этапе это количество можно задавать произвольно.

Вероятность перехода k-го муравья из позиции i в позицию j на цикле с номером t определяется следующим соотношением:

$$P_{ij,k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha}{\sum\limits_{l \in J_{i,k}^t} [\tau_{il}(t)]^\alpha}, \ j \in J_{i,k}^t, \\ 0, \quad j \notin J_{i,k}^t, \end{cases}$$

где α — параметр, задающий вес следа феромона.

Количество откладываемого феромона составляет величину:

$$\Delta\tau_{ij,k}(t) = \begin{cases} \dfrac{Q}{L_k(t)}, (i, j) \in T_k(t), \\ 0, (i, j) \notin T_k(t), \end{cases}$$

где Q — положительный параметр, $L_k(t)$ — значение накрывающего отображения на перестановке, соответствующей маршруту k-го муравья на цикле с номером t. Изменение количества феромона определяется следующим выражением:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij,k}(t),$$

где m — количество муравьев, ρ — коэффициент «испарения» $(0 < \rho < 1)$.

Алгоритм прекращает работу, когда выполнено некоторое правило остановки, например, достигнута граница числа циклов. Минимальная по значению накрывающего отображения перестановка, найденная на последнем цикле, преобразуется в решение исходной задачи.

Результаты работы. Для проверки качества предлагаемых метаэвристик было сгенерировано 100 задач со случайным размещение точек на плоскости и со случайными оценками стоимости организации производства. Использовались три вида эвристик: локальный алгоритм со случайным выбором начальной точки, муравьиный и эволюционный алгоритмы на фрагментарных структурах. В каждом подходе выполнялось примерно одинаковое количество вычислений значения целевой функции.

Решения, полученные в результате применения различных алгоритмов, сравнивались по значению целевой функции.

Результаты работы алгоритмов сравнивались по числу первых мест и по рейтингу, который рассчитывался как сумма набранных алгоритмом очков. За первое место предлагалось три очка, за второе — два, за третье — одно очко (таблица).

Таблица

Результаты тестирования алгоритмов

Алгоритм	Кол-во задач	Рейтинг
Локальный	100	269
Эволюционный	100	210
Муравьиный	100	121

Выводы. Теоретические результаты и результаты численных экспериментов показали достаточно высокую эффективность эволюционного алгоритма и алгоритма муравьиной колонии при решении задачи размещения производства. Учитывая простоту реализации и возможность учета дополнительных ограничений, рассматриваемый в статье подход может быть предложен для практического решения задач размещения объектов коллективного пользования с различными ограничениями.

Список использованной литературы:

1. Khumawala B. M. An Efficient Branch-Bound Algorithm for the Warehouse Location Problem. *Management Science*. 1972. Vol. 18. P. 718–731.
2. Krarup J., Pruzan P.M. The simple plant location problem: Survey and synthesis. *European Journal of Operational Research*. 1983. Vol. 12. P. 36–81.
3. Береснев В. Л., Гимади Э. Х., Дементьев В. Т. Экстремальные задачи стандартизации. Новосибирск : Наука, 1978. 333 с.
4. Козин И. В., Перепелица В. А., Максишко Н. К. Фрагментарные структуры в задачах дискретной оптимизации. *Кибернетика и системный анализ.* 2017. № 6. С. 125–131.

5. Козин И. В. Фрагментарные структуры и эволюционные алгоритмы. *Питання прикладної математики і математичного моделювання.* Дніпропетровськ, 2008. С. 138–146.
6. Dorigo M. Optimization, Learning, and Natural Algorithms. PhD Thesis, Dipartimento di Elettronica, Politechnico Di Milano, Italy. 1992. 140 p.
7. Moraglio A., Poli R. Inbreeding Properties of Geometric Crossover and Non-geometric Recombinations. *Foundations of Genetic Algorithms.* 2007. P. 1–14.

FRAGMENTAL MODEL PLACEMENT PRODUCTION

A two-dimensional problem of locating production objects in a discrete formulation is considered. It is shown that the discrete problem of locating production reduces to the problem of covering a graph with stars and has a fragmentary structure. To search for an approximate solution of the problem, a modification of the evolutionary algorithm on permutations with a geometric crossover operator and an ant colony algorithm on a fragmentary structure are proposed. The results of numerical experiment comparison of algorithms are given.

Key words: *fragmented model, task of locating production, evolutionary algorithm, geometric crossover, ant colony algorithm.*

Получено 29.01.2019

УДК 519.8
DOI: 10.32626/2308-5878.2019-19.41-46

О. М. Коломис, канд. фіз.-мат. наук

Інститут кібернетики імені В. М. Глушкова НАН України, м. Київ

ОЦІНКА ПОХИБКИ ЗАОКРУГЛЕННЯ АЛГОРИТМУ ОБЧИСЛЕННЯ ОЦІНКИ СПЕКТРАЛЬНОЇ ЩІЛЬНОСТІ

У роботі розглянуто ефективні за швидкодією алгоритми обчислення оцінок спектральних щільностей стаціонарних ергодичних випадкових процесів із нульовим середнім значенням. Найчастіше для їх обчислення використовують метод прямого перетворення Фур'є з використанням алгоритму швидкого перетворення Фур'є (ШПФ). Стаття продовжує дослідження і обґрунтування цього методу в напрямку отримання більш якісних оцінок похибок заокруглення. Наведена оцінка похибки заокруглення алгоритму обчислення оцінки спектральної щільності.

Ключові слова: *оцінка спектральної щільності, похибка заокруглення, швидке перетворення Фур'є.*

Вступ. Швидкі алгоритми розв'язання задач спектрального і кореляційного аналізу випадкових процесів почали з'являтися, в основному, після 1965 року, коли в обчислювальну практику увійшов алгоритм ШПФ [1, 2]. З його появою розроблено ряд обчислювальних