Теорія оптимальних рішень

УДК 519.8

О.В. ШИЛО

АЛГОРИТМ ГЛОБАЛЬНОГО РАВНОВЕСНОГО ПОИСКА ДЛЯ РЕШЕНИЯ ЗАДАЧИ О Р-МЕДИАНЕ

Рассматривается задача о p-медиане, предложен алгоритм, основанный на методе глобального равновесного поиска для её решения. Проведено сравнение со случайной генерацией решений и генерацией решений с помощью процедуры GRASP. Проведен вычислительный эксперимент на тестовых задачах, обсуждаемых в литературе.

Постановка задачи. Задача о p-медиане определяется следующим образом. Пусть заданы: множество F, состоящее из m ресурсов; множество U, состоящее из n пользователей; определена функция расстояния $d: U \times F \to R^1$ и константа $p \le m$. Необходимо найти подмножество $S \subset F$, удовлетворяющее следующим условиям: $S \subset F$, $|S| = p$ ($|S|$–мощность обозначает множества S), $\sum_{u \in U} \min_{f \in S} [d(u,f)]) \le \sum_{u \in U} \min_{f \in Q} [d(u,f)])$, для всех подмножеств $Q \subset F$, что $|Q| = p$. Другими словами, надо выбрать (открыть) p ресурсов из F так, чтобы сумма расстояний от каждого пользователя из U до ближайшего открытого ресурса была минимальной.

Так как эта задача относится к классу NP-сложных [1], то для её решения на практике оказывается эффективным применение различного рода эвристических процедур. Одними из наиболее распространенных и эффективных подходов являются эвристики, использующие локальный поиск, предложенный в [2]. К ним относятся подходы, использующие метод табу [3], процедуру GRASP (Greedy Randomized Adaptive Search Procedure) [4], метод варьированного локального поиска [5].

В данной работе предложен эвристический алгоритм, основанный на использовании метода глобального равновесного поиска (ГРП) [6] для решения задачи о p-медиане. Метод ГРП генерирует случайные решения,

которые адаптируются в процессе решения задачи, используя уже полученные решения. В основе метода лежит температурный цикл: при начальных значениях температуры генерируются абсолютно случайные решения, а в конце температурного цикла – решения, близкие к лучшим, найденным за весь цикл. В начале статьи описан предложенный алгоритм, далее приведены результаты вычислительного эксперимента по решению известных тестовых задач, обсуждаемых в литературе для задачи о p-медиане. Рассмотрены варианты алгоритмов со случайной генерацией решений, генерацией решений на основе GRASP процедуры и с помощью алгоритма ГРП.

Алгоритм глобального равновесного поиска. Рассмотрим решение задачи в виде двоичного m-мерного вектора x, j-й элемент которого равен единице, если в F j-й ресурс открыт; в противном случае он равен нулю. Таким образом, допустимым решением для задачи о p-медиане может служить любой двоичный вектор x с p ненулевыми элементами.

Алгоритм ГРП генерирует решения с заданной вероятностью, основанной на результатах предшествующей истории поиска. Обозначим p_j вероятность того, что в конструируемом решении j-й элемент соответствующего ему двоичного вектора x будет равен единице. Задавшись вектором $\{p_j\}$, где $j = 1...m$, можно легко построить процедуру для генерации случайных допустимых решений.

В основе метода ГРП лежит температурный цикл, на каждом этапе которого задаётся температура μ, которая влияет на вероятности решений, генерируемых на данном этапе. Эти температуры выбираются таким образом, чтобы на первом этапе генерируемые решения были абсолютно случайными, а на последнем этапе генерировалось лучшее найденное решение. Опишем подробнее, как рассчитываются вероятности для генерации решений.

Обозначим $p_j(\mu)$ вероятность того, что в конструируемом решении j-я координата равна единице на этапе с температурой μ. Если решение представляется двоичным вектором x, то обозначим $f(x)$ стоимость этого решения (сумма расстояний от каждого пользователя из U до ближайшего открытого ресурса).

Зададим числа K и μ_0, $\mu_1,\ldots,\mu_K$, такие, что $\mu_0 < \mu_1 < \ldots < \mu_K$. Пусть F – некоторое известное множество решений задачи, $F_j^1 = \{x \in F:\ x_j = 1\}$, $F_j^0 = \{x \in F:\ x_j = 0\}$.

Для $q = 0,1,\ k = 0,\ldots,K,\ j = 1,\ldots,m$ определим следующие величины:

$$Z_{kj}^q = \sum_{x \in F_j^q} \exp\{-\mu_k f(x)\},\ G_{kj}^q = \sum_{x \in F_j^q} f(x)\exp\{-\mu_k f(x)\},\ E_{kj}^q = \frac{G_{kj}^q}{Z_{kj}^q}$$

В рамках метода ГРП [6] вероятности для проведения случайного поиска рассчитываются при $j = 1,...,m,\ k = 1,...,K$ по формуле

$$p(\mu_k) = \frac{1}{1 + \dfrac{1 - p_j(\mu_0)}{p_j(\mu_0)} \exp\left\{-0.5 \sum_{i=0}^{k-1} \left(E_{ij}^0 + E_{i+1j}^0 - E_{ij}^1 - E_{i+1j}^1\right)\left(\mu_{i+1} - \mu_i\right)\right\}}.$$

Принципиальная схема алгоритма глобального равновесного поиска показана на рисунке.

```
1:  начальная_установка(M,K,nstarts,p(μ0))
2:  while критерий останова не выполнен do
3:    repeat
4:      s = 0
5:      for k = 0 to K do
6:        μ = M[k], найти вектор вероятностей p(μ)
7:        for i = 0 to n_starts do
8:          x =генерация_решения(p(μ))
9:          x =локальный_поиск(x)
10:         if x ∉ F then
11:           F = F ∪ x
12:         end if
13:         if решение x лучше всех решений из F then
14:           s = 0
15:         end if
16:       end for
17:     end for
18:     s = s + 1
19:     F={лучшее решение из F}
20:   until s < max_fail
21:   F = ∅
22: end while
```

РИСУНОК. Алгоритм ГРП

В первой строке рассматриваемой схемы проводится инициализация алгоритма, задаются температуры, количество генераций решений на каждом этапе температурного цикла. Далее идёт цикл, который завершается при достижении критерия останова (максимальное число итераций, время работы алгоритма). В строках 5–17 приведен температурный цикл алгоритма. На каждом его этапе сначала выбирается температура и для неё находится вектор вероятностей по вышеописанным формулам, после чего проводится серия генераций решений с заданными вероятностями (строка 8). По окрестности каждого сгенерированного решения проводится поиск более хороших решений (строка 9), информация о новом найденном решении запоминается (строки 10–11) путем модификации

величин Z_{kj}^q, G_{kj}^q. В конце каждого температурного цикла оставляем информацию о лучшем найденном решении (строка 19). Параметром max_nfail задается количество повторений температурного цикла без улучшения решений в F. После того, как было проведено max_nfail температурных циклов без улучшения, «забываем» всю информацию об истории поиска (строка 21), и алгоритм снова начинает работу с чистого листа.

Особенности локального поиска. Для улучшения случайно генерируемых решений использовался локальный поиск, описанный в [2]. Если S – множество открытых ресурсов в исходном решении, то рассматриваются решения, получаемые при открытии ресурса, не принадлежащего S и закрытии какого-то ресурса из S. Из этой окрестности выбирается лучшее решение и далее рассматривается его окрестность. Таким образом, перебрав все возможные пары (f, q), где $f \in S$, $q \in F \setminus S$, и рассчитав для каждой пары стоимость решения, получаемого при замене в S ресурса f на q, можно получить лучшее решение из окрестности. В работе [7] предлагается схема, позволяющая существенно ускорить локальный поиск. Эта схема достаточно громоздка и поэтому не приводится в данной статье. Во всех реализациях алгоритмов была применена схема реализации локального поиска из [7] с использованием разреженной матрицы и предварительной сортировкой расстояний. Несмотря на некоторое увеличение требований к памяти, этот подход позволяет достигнуть ускорения локального поиска на некоторых задачах в 100 раз.

Результаты вычислительного эксперимента. Для тестирования предложенного алгоритма была использована задача из библиотеки, расположенной по адресу http://www.iwr.uni-eidelberg.de/groups/comopt/software/TSPLIB95/. Мы рассматривали задачу fl1400, которая обычно обсуждается в работах по p-медиане. Она получена из координат 1400 точек на плоскости, которые изначально использовались для исследования задачи о коммивояжере. В случае задачи о p-медиане каждая точка является одновременно и ресурсом, и пользователем, используются эвклидовы расстояния. Решались задачи с p от 10 до 500. Для сравнения было реализовано три варианта алгоритма: с абсолютно случайной генерацией решений, генерацией решений с помощью варианта GRASP алгоритма, предложенного в [4], и с помощью алгоритма ГРП. Для каждого варианта проводилась серия из 5000 генераций. Вычислительный эксперимент осуществлялся с использованием компьютера Pentium 733 Mhz с 128 Мб оперативной памяти. Матрица расстояний во всех алгоритмах рассчитывалась предварительно и представлялась массивом типа double, без округления значений расстояний.

В табл. 1 приведены результаты для случайной генерации решений. Приведенные лучшие известные решения взяты из [3]. В столбце *„Случайная генерация"* содержатся лучшие решения, найденные при случайной генерации; в столбце *„Среднее число шагов локального поиска"* указывается, сколько в сред-

нем итераций требуется для достижения локального оптимума. *„Среднее улучшение"* показывает, на сколько процентов в среднем улучшает случайное решение локальный поиск; *„Количество локальных оптимумов"* – сколько различных локальных оптимумов было найдено на всех итерациях, *„ARE"* – среднее относительное отклонение полученных решений от лучшего известного решения ($ARE = < (f - f_{bks}) / f_{bks} >$, где f_{bks} – стоимость лучшего известного решения, а усреднение берётся по всем найденным решениям). Также приводится время для проведения 5000 случайных генераций и локального поиска.

ТАБЛИЦА 1. Результаты решения задачи fl1400 с помощью случайной генерации решений

p	Лучшее известное решение	Случайная генерация	Среднее число шагов локального поиска	Среднее улучшение, %	Количество локальных оптимумов	ARE, %	Время, с
10	101249,47	101249,55	14	55,99	2	0,09	5666
20	57857,55	57857,94	29	52,98	14	0,48	3288
30	44013,48	44013,48	42	51,92	1263	0,95	2795
40	35002,52	35002,52	56	54,03	1335	0,27	2626
50	29089,78	29090,23	65	55,46	3977	0,98	2519
60	25161,12	25183,37	73	56,93	4995	0,78	2374
70	22125,53	22130,35	83	57,99	4996	1,05	2341
80	19872,72	19875,16	92	59,08	5000	0,76	2376
90	17987,94	18020,93	101	59,82	5000	1,06	2397
100	16551,2	16559,04	112	60,54	5000	0,66	2435
150	12026,47	12053,35	144	62,62	5000	0,79	2575
200	9359,15	9380,47	177	64,48	5000	0,87	2768
250	7741,51	7754,07	205	65,50	5000	0,57	2939
300	6620,92	6645,53	212	65,82	5000	0,82	2889
350	5720,91	5761,97	227	66,16	4999	1,37	2970
400	5006,83	5050,04	247	66,44	5000	1,71	3082
450	4474,96	4498,87	258	66,41	5000	1,11	3649
500	4047,90	4068,00	260	65,97	5000	0,97	3128

Как видно из табл.1, локальный поиск существенно улучшает получаемые при случайной генерации решения, и даже при случайной генерации начальных решений получаются очень хорошие результаты (в среднем хуже лучших известных решений всего на 1%). Это свидетельствует о высокой эффективности использованного локального поиска для данной задачи.

Второй вариант, который тестировался, был предложен в [4]. Начиная с пустого множества S, первый ресурс выбирается случайным образом и добавляет-

ся в S, остальные ресурсы добавляются в S по одному. Для каждого пользователя из U рассчитывается разница между расстоянием до ближайшего ресурса из S и расстоянием после добавления в S лучшего для этого пользователя ресурса. Далее, согласно вероятностям, пропорциональным рассчитанным разностям выбирается некоторый пользователь и ближайший к нему ресурс добавляется в S. Процедура повторяется пока в S не будет ровно p пользователей. Результаты для этого варианта приведены в табл. 2. В столбце *«GRASP»* приведены лучшие результаты, полученные на всех генерациях. Как видно, среднее число итераций локального поиска до достижения локального оптимума существенно уменьшилось. Некоторое увеличение вычислительных затрат на генерацию решений компенсируется за счёт меньшего числа итераций локального поиска. Поэтому для некоторых задач вычислительные затраты оказались меньше, чем при случайной генерации. Однако при увеличении p это преимущество исчезает, так как увеличение времени на генерацию становится больше времени, которое выигрывается при сокращении числа итераций локального поиска.

ТАБЛИЦА 2. Результаты решения задачи fl1400 с помощью процедуры GRASP

p	Лучшее известное решение	GRASP	Среднее число шагов локального. поиска	Среднее улучшение, %	Количество локальных оптимумов	ARE, %	Время, с
10	101249,47	101249,55	14	30,73	2	0,09	4825
20	57857,55	57857,94	27	33,09	14	0,35	2760
30	44013,48	44013,48	40	30,19	1342	0,75	2490
40	35002,52	35002,52	50	31,29	1669	0,24	2384
50	29089,78	29090,22	59	31,83	3941	0,86	2326
60	25161,12	25166,78	65	32,04	4996	0,67	2220
70	22125,53	22126,03	74	32,12	4997	0,98	2226
80	19872,72	19875,06	81	32,10	5000	0,73	2261
90	17987,94	18006,97	87	32,16	4999	0,95	2305
100	16551,2	16552,35	95	31,77	5000	0,61	2348
150	12026,47	12050,90	123	28,98	5000	0,77	2623
200	9359,15	9376,27	149	28,38	4999	0,84	2916
250	7741,51	7755,42	170	27,48	5000	0,56	3174
300	6620,92	6645,32	173	26,24	5000	0,82	3249
350	5720,91	5765,36	181	25,73	5000	1,42	3408
400	5006,83	5059,59	192	25,56	5000	1,79	3583
450	4474,96	4498,87	198	25,40	2500	1,11	3726
500	4047,90	4066,83	197	24,51	5000	0,97	3797

В табл. 3 приведены результаты решения задачи fl1400 с помощью алгоритма ГРП. Тестирование проводилось со следующими параметрами: *max_nfail=3,*

K=20, n_starts=10. При получении нового лучшего решения при некоторой температуре проводилось дополнительно *n_starts* генераций решений с этой температурой. В столбце *„ГРП"* приведено лучшее найденное решение на всех генерациях. Для некоторых значений p найдены новые верхние оценки (в таблице они выделены жирным шрифтом). В столбце *„Resende"* приведены результаты, полученные в [4] с помощью GRASP алгоритма с использованием процедуры path-relinking. В столбце *„Mlad"* представлены результаты, полученные в [5] с помощью варьированного локального поиска.

ТАБЛИЦА 3. Результаты решения задачи fl1400 с помощью алгоритма ГРП и результаты для других алгоритмов

P	Лучшее известное решение [4]	ГРП	Время, с	Resende	Mlad
10	101249,47	101249,55	5693	101249,55	101249,47
20	57857,55	57857,55	3061	57857,94	57857,55
30	44013,48	44013,48	2400	44013,48	44086,53
40	35002,52	35002,52	2070	35002,60	35005,82
50	29089,78	29090,23	2791	29090,23	29130,10
60	25161,12	25165,64	2451	25164,02	25176,47
70	22125,53	22126,03	2279	22126,03	22186,14
80	19872,72	**19870,85**	2140	19876,57	19900,66
90	17987,94	17988,60	1754	17988,60	18055,94
100	16551,20	16553,07	1762	16559,82	16551,20
150	12026,47	12026,46	1926	12036,00	12035,56
200	9359,15	**9357,90**	2169	9360,67	9362,99
250	7741,51	**7738,37**	2701	7746,31	7746,96
300	6620,92	6623,42	2509	6623,98	6628,92
350	5720,91	5721,09	2626	5727,17	5739,28
400	5006,83	5010,13	2552	5010,22	5045,84
450	4474,96	4479,93	2853	4476,68	4489,93
500	4047,90	**4047,20**	2389	4049,56	4062,86

Из табл. 3 видно, что предложенный алгоритм на основе метода ГРП позволил получить очень качественные результаты для всех тестовых задач. Следует отметить, что в некоторых случаях результаты, полученные с помощью варьированного локального поиска, предложенного в [5], который тоже использует окрестность из [2], оказались в ряде случаев даже хуже, чем при использовании обычного локального поиска [2]. Алгоритм ГРП на всех задачах, кроме задачи с p =100, показал более качественные результаты по сравнению с результатами для абсолютно случайной генерации и для генерации с помощью варианта GRASP процедуры.

Следует объяснить, почему для ряда задач время, требуемое для генерации 5000 решений алгоритмом ГРП с последующим применением локального поиска, оказалось меньше, чем при абсолютно случайной генерации. Это связано с тем, что алгоритм ГРП начинает генерировать абсолютно случайные решения при начальных температурах, зато при больших температурах генерируемые решения оказываются близкими к лучшему найденному решению. Поэтому число итераций, необходимых для достижения локального оптимума, заметно уменьшается в конце температурного цикла. Тот же эффект был замечен при тестировании GRASP алгоритма, но так как среднее число итераций локального поиска для алгоритма ГРП меньше, чем для GRASP алгоритма, то для некоторых задач ему потребовалось существенно меньше времени для проведения 5000 генераций. Таким образом, для ряда задач оказалось быстрее провести серию генераций решений с помощью алгоритма ГРП, чем такое же число абсолютно случайных генераций.

Выводы. Результаты вычислительного эксперимента свидетельствуют о высокой эффективности локального поиска из [2] для задачи о p-медиане. За счёт этого даже абсолютно случайная генерация решений позволяет получать очень качественные решения. Исследованный вариант процедуры GRASP позволяет улучшить качество генерируемых решений, однако для некоторых задач лучшие найденные решения оказались даже хуже, чем при абсолютно случайной генерации. Напротив, алгоритм глобального равновесного поиска показал устойчивое поведение на всех тестируемых задачах.

О.В. ШИЛО

АЛГОРИТМ ГЛОБАЛЬНОГО РІВНОВАЖНОГО ПОШУКУ
ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ ПРО Р-МЕДІАНУ

Запропоновано алгоритм розв'язування задачі про p-медіану, що базується на використанні методу глобального рівноважного пошуку. Проведено порівняння розробленого алгоритму з алгоритмом, що генерує абсолютно випадкові розв'язки, та з реалізацією GRASP процедури. Результати експериментальних розрахунків свідчать про перспективність запропонованого підходу. Для деяких задач отримано нові верхні оцінки. Розроблений алгоритм дозволяє одержувати якісні розв'язки за прийнятний час.

O.V. SHYLO

A GLOBAL EQUILIBRIUM SEARCH ALGORITHM FOR THE P-MEDIAN PROBLEM

The paper proposes an algorithm based on a global equilibrium search for the p-median problem. A comparison with random generation scheme and GRASP generation scheme is conducted. Computational experiments on a standard problem instances show that this approach is a competitive tool for the p-median problem, for some instances new upper bound were obtained. The proposed algorithm allows to obtain qualitive solutions in a reasonable ammount of time.

1. *Kariv O., Hakimi S.L.* An Algorithmic Approach to Network Location Problems; Part 2. The *p*-Medians // *SIAM J. on Applied Mathematics.* – 1969. – № 37. – P. 539–560.
2. *Teitz M. B., Bart P.* Heuristic methods for estimating the generalized vertex median of a weighted graph // *Oper. Res.* – 1968. – № 16(5). – P. 955–961.
3. *Rolland E., Schilling D.A., Current J.R.* An efficient tabu search procedure for the *p*-median problem // *Eur.J. of Oper. Res.* – 1996. – № 96. – P. 329–342.
4. *Resende M. G., Werneck R.* A GRASP with path-relinking for the *p*-median problem // AT&T Labs Research Technical Report TD-5E53XL. – 2002.–30 p.
5. *Hansen P., Mladenovi´c N., Perez-Brito D.* Variable neighborhood decomposition search // *J. of Heuristics.* – 2001. – № 7(3). – P. 335–350.
6. *Шило В.П.* Метод глобального равновесного поиска // Кибернетика и системный анализ. – 1999. – №1. – С. 74–80.
7. *Resende M. G., Werneck R.* On the implementation of a swap-based local search procedure for the *p*-median problem // Technical Report TD-5E4QKA, AT&T Labs Research. – 2002.–29 p.

Получено 22.07.2004