УДК 519.854

П.В. ШИЛО

ПОВТОРЯЕМЫЙ ИТЕРИРОВАННЫЙ АЛГОРИТМ ТАБУ ДЛЯ РЕШЕНИЯ КВАДРАТИЧНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Аннотация. Разработан новый алгоритм повторяемого табу для решения квадратичной задачи о назначениях. Проведенное сравнительное исследование данного алгоритма с лучшими в настоящее время алгоритмами решения этой задачи показало его конкурентоспособность как по быстродействию, так и по возможности получения лучших решений.

Ключевые слова: квадратичная задача о назначениях, табу, вычислительный эксперимент, сравнительное исследование алгоритмов.

Квадратичная задача о назначениях (quadratic assignment problem, QAP) впервые сформулирована в работе [1]. Она имеет многочисленные практические приложения, связанные с размещением различных объектов, балансированием турбин, анализом изображений и др. [1–7].

Рассматриваемая QAP является NP-сложной задачей дискретной оптимизации, к тому же очень трудной в вычислительном отношении. Отметим также, что ее частным случаем есть задача коммивояжера.

ПОСТАНОВКА ЗАДАЧИ

Содержательная постановка задачи QAP такова. Дано n объектов, которые необходимо разместить в n различных локациях (местах назначения). Известны величины потоков ресурсов a_{ij} между объектами i и j, $i, j = 1, \dots, n$, и расстояния b_{rs} между локациями r и s, $r, s = 1, \dots, n$. Требуется найти такое распределение объектов по локациям, чтобы сумма расстояний, умноженная на соответствующие потоки, была минимальной.

Следовательно, математическую постановку задачи QAP можно сформулировать следующим образом: найти

$$\min_{\pi \in \Pi_n} f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j},$$

где $A = [a_{ij}]$ и $B = [b_{rs}]$ — квадратные матрицы порядка n, Π_n — множество всех перестановок $\{1, \dots, n\}$ и π_i задает номер локации объекта i.

Квадратичные задачи о назначениях удается точно решить только для малых размерностей. Предложенная в [7] тестовая задача с 36 переменными точно не решена до сих пор. В связи с этим актуальны разработка, исследование новых и усовершенствование существующих приближенных алгоритмов решения задачи QAP.

ПРЕДЛАГАЕМЫЙ АЛГОРИТМ

Вычислительная сложность задачи QAP обусловлена большим объемом вычислений, необходимых для получения значения целевой функции (порядка $O(n^2)$) и исследования окрестности текущего решения. Отметим, что в алгоритмах для задачи QAP решение обычно представляется в виде некоторой перестановки $\pi \in \Pi_n$, а его окрестность $N(\pi)$ включает все перестановки, полученные путем обмена локациями между всеми парами объектов. Таким образом, окрестность содержит $C_n^2 + 1$ перестановку, а вычисление значений их целевых функций требует порядка $O(n^4)$ объема вычислений. При $n \geq 100$ проведение хотя бы n^2 итераций локального поиска уже становится трудной вычислительной задачей. Поэтому важно применять предложенные в [8, 9] способы пересчета приращений целевой функции при обмене локациями. Это позволяет снизить до $O(n^3)$ объем вычислений, необходимых для исследования окрестности. Его можно уменьшить еще больше в случае разреженных и симметричных матриц.

Данные рассуждения скорее относятся к вычислительной сложности, чем к оптимизационной. Определенный вклад в оптимизационную сложность задачи QAP вносит выбор ходов локальных алгоритмов — обмен между объектами своими локациями. Ясно, что такой выбор искусственный, слабо связан с решаемой задачей и считается худшим для задачи коммивояжера. Однако большинство задач QAP в оптимизационном плане просты и решаются современными алгоритмами. Тестовые задачи taiXX [8] в силу их генерации представляют особую сложность.

Из исследования тестовых задач в [10] следует, что для большинства задач taiXX «ландшафт» целевой функции характеризуется множеством несвязанных локальных оптимумов, отрицательной корреляцией между значением целевой функции локального оптимума и его расстоянием до глобального оптимума. Поэтому необходим алгоритм с сильной диверсификацией поиска, что позволит быстро находить отдельно расположенные области локальных оптимумов и тщательно их исследовать. В работе [11] предложена схема возмущения переменных, названная впоследствии итерированностью. Она успешно используется для диверсификации поиска решения. Для тщательного исследования района локального оптимума разработан повторяемый алгоритм табу, разновидность которого показала хорошие результаты при решении задач WMaxCut [12]. Эти предложения объединены в рассматриваемом алгоритме RITS (повторяемый итерированный табу-поиск).

Далее приведен алгоритм RITS в виде такой процедуры:

procedure RITS (x_{best}, ntm_ar, dd_ar)
1. $fbest = \infty$; задание начальных данных
2. while (*не выполнен критерий останова*){
3. $fmin = \infty$; $impr =$ false;
4. for ($\mu = 0$; $\mu < \mu max$; $\mu{+}{+}$){
5. $tmax = tm\_ar[\mu]$;
6. for ($t = 0$; $t < tmax$; $t{+}{+}$){
7. $x \leftarrow$ disturbation solution(x_{best}, $dd[\mu]$)
8. $x \leftarrow$ ReTS(x)
9. if ($f(x) < fbest$){$fbest = f$; $x_{best} = x$; break;}
10. }
11. }
12. }

Здесь значение $tm\_ar[0] = \infty$. Поэтому цикл итерированности (строки 4–12) будет выполняться при $\mu > 0$ только при каждом улучшении x_{best} (строка 9). Если заменить строку 7 строкой «$x \leftarrow$ disturbation solution ($x_{\min}$, $dd[\mu]$)», а строку 9 — строкой «if ($f(x) < fmin$) {$fmin = f$; $x_{\min} = x$; break;}», то получим более общую схему, когда цикл итерированности выполняется при каждом улучшении $x_{\min}$. В строке 7 происходит возмущение решения x_{best}. Будем считать, что возмущаемое решение записано в массив в ячейках $0, \ldots, n-1$, где n — размерность решаемой задачи. Тогда из чисел, записанных в ячейках $0, \ldots, dd[\mu]$, случайно генерируется перестановка, равномерно распределенная на множестве всех таких перестановок и входящая в возмущенное решение.

Опишем теперь предлагаемый повторяемый алгоритм табу ReTS.

```
procedure ReTS(x)
1. x_m = x; istep = step = 0; M = {m_i, i = 1, 2, ..., C_n^2};
2. nfail = 0; Задание nbstep.
3. istep = step; x = x_m; ch(i) = −∞, i = 1, 2, ..., n^2; Задание массива tn[i, k], i, k = 1, ..., n^2;
4. while (nfail < maxnfail){
5.       Δ = ∞;
6.       for k = 1 to |M|{
7.             i = nl1[k]; j = nl2[k]
8.             if ((step − ch[i, x[j]] < tn[i, x[j]] AND step − ch[j, x[i]] < tn[j, x[i]]) OR
               f(x) + g_k(x) < f(x_m)){
9.                   if (g_k(x) < Δ) {Δ = g_k(x); kk = k}
10.            }
11.      }
12.      i = nl1[kk]; j = nl2[kk]
13.      ch[i, x[i]] = step; ch[j, x[j]] = step; Задание величин tn[i, x[i]], ch[j, x[j]].
14.      k = x[i]; x[i] = x[j]; x[j] = k;
15.      if (f(x) < f(x_m)){x_m = x; goto to line2;}
16.      step = step + 1;
17.      if (step − istep > nbstep){nfail = nfail + 1; goto to line3;}
18. }
19. x = x_m;
20. return x;
```

Здесь использованы такие обозначения:

- M — множество ходов алгоритма табу, состоящее из всех пар локаций. В процессе хода объекты, находящиеся в этих локациях, меняются местами;
- $ch[i, r]$ — последний шаг алгоритма табу, когда в локации i находился объект r;
- $tn[i, r]$ — количество шагов алгоритма табу, в течение которых запрещен возврат объекта r в локацию i, $tn[i, r] = tmin + \delta * \xi$, $i, r = 1, \ldots, n^2$, ξ — случайная величина, равномерно распределенная на интервале $[0, 1]$;
- $nbstep$ — максимальное количество шагов алгоритма табу без улучшения решения x_m, задается следующим образом: if $(f(x_m) > f(x_{best}))$ $nbstep = nbstep1$; else $nbstep = nbstep2$, где x_{best} — найденное наилучшее решение;
- $maxnfail$ — максимальное количество попыток улучшения решения x_m, задается следующим образом: if $(f(x_m) > f(x_{best}))$ $maxnfail = nf1$; else $maxnfail = nf2$;
- $nl1[k]$ и $nl2[k]$ — номера локаций в ходе k.

РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

В работе [8] имеется обширный набор тестовых задач QAP, который весьма популярен среди исследователей и широко используется ими.

В оптимизационном плане задачи типа taiXXa наиболее трудные, поэтому именно с ними проведены вычислительные эксперименты.

Т а б л и ц а 1

Задача	BKS	Значение $\bar{\delta}_{avg}$ для алгоритмов						
		RITS	CPTS	ITS	IHGA	BLS	BMA	PILS
tai40a	3139370	0.03(6)	0.148(1)	0.210(1)	0.209(1)	**0.022**(7)	0.059(2)	0.28(0)
tai50a	4938796	**0.04**(6)	0.440(0)	0.373(0)	0.262(0)	0.157(2)	0.131(2)	0.663(0)
tai60a	7205962	**0.142**(1)	0.476(0)	0.330(1)	0.583(0)	0.251(1)	0.144(2)	0.82(0)
tai80a	13499184	**0.41**(0)	0.691(0)	0.494(0)	0.756(0)	0.517(0)	0.426(0)	0.927(0)
tai100a	21052466	**0.33**(0)	0.589(0)	0.427(0)	0.606(0)	0.430(0)	0.405(0)	1.027(0)
Среднее значение	—	**0.231**	0.469	0.367	0.483	0.275	0.233	0.413

Предлагаемый алгоритм реализован на языке C++, среда Microsoft Visual Studio 2012, все вычислительные эксперименты проводились с использованием PC с Intel® Core QUAD CPU Q9550 2.83GHz и 8.0GB оперативной памяти. Каждая задача решалась 10 раз (независимые испытания при различных значениях датчика псевдослучайных чисел), выделенный лимит времени для одной попытки решения составлял 2 часа. Далее приведены значения параметров алгоритма RITS, которые использовались в экспериментах: $tm\_ar[0]=\infty$, $tm\_ar[\mu]=9$, $\mu=1,\ldots,\mu max-1$; $\mu max=10$; $dd[0]=n-1$, $dd[\mu]=\mu*n/10$, $\mu=1,\ldots,\mu max-1$; $tmin=0.07*n$; $\delta=0.15*n$; $nbstep1=27$; $nbstep2=n^2$; $nf1=27$; $nf2=81$.

В работе [10] проведены подобные экспериментальные исследования, поэтому они использованы для сравнения с предлагаемым алгоритмом. Выбрано шесть современных алгоритмов, которые показывают хорошие результаты:

- Cooperative Parallel Tabu Search (CPTS) [13];
- Iterated Tabu Search (ITS) [14];
- Improved Hybrid Genetic Algorithm (IHGA) [15];
- Breakout Local Search (BLS) [16];
- Breakout Memetic Algorithm (BMA) [10];
- Population-based Iterated Local Search (PILS) [17].

Отметим, что алгоритм из [18] не включен для сравнения, так как он исследовался при других условиях.

В табл. 1 приведены результаты сравнительного исследования упомянутых алгоритмов с предложенным алгоритмом RITS, где BKS — известный из работы [14] рекорд (наименьшее значение целевой функции), $\bar{\delta}_{avg}$ — отклонение (в процентах) среднего значения целевой функции, полученного с помощью данного алгоритма из 10 независимых просчетов, от BKS. В скобках рядом со значением $\bar{\delta}_{avg}$ указано число найденных рекордов при 10 попытках решения задачи.

Из анализа результатов экспериментальных расчетов можно сделать вывод о высокой эффективности и конкурентоспособности повторяемого итерированного алгоритма табу для решения квадратичной задачи о назначениях.

СПИСОК ЛИТЕРАТУРЫ

1. Koopmans T.C., Beckmann M.J. Assignment problems and the location of economic activities. *Econometrica*. 1957. Vol. 25. P. 53–76.
2. Burkard R.E. Quadratic assignment problems. *European J. of Oper. Res*. 1984. Vol. 15. P. 283–289.
3. Eiselt H.A., Laporte G. A Combinatorial optimization problem arising in dartboard design. *J. of the Oper. Res. Society*. 1991. Vol. 42. P. 113–118.
4. Elshafei A.E. Hospital layout as a quadratic assignment problem. *Oper. Res. Quarterly*. 1977. Vol. 28. P. 167–179.

5. Nugent Ch.E., Vollmann Th.E., Ruml J. An experimental comparison of techniques for the assignment of facilities to locations. *Oper. Res*. 1968. Vol. 16. P. 150–173.
6. Finke G., Burkard R.E., Rendl F. Quadratic assignment problems. *Annals of Discrete Math*. 1987. Vol. 31. P. 61–82.
7. Steinberg L. The backboard wiring problem: a placement algorithm. *SIAM Review*. 1961. Vol. 3. P. 37–50.
8. Taillard E. Robust taboo search for the quadratic assignment problem. *Parallel Computing*. 1991. Vol. 17. P. 443–455.
9. Podolsky S., Zorin Yu. O(1) delta part computation technique for the quadratic assignment problem. *System Research & Information Technologies*. 2015. Vol. 2. P. 112–121.
10. Benlic U., Hao J.K. Memetic search for the quadratic assignment problem. *Expert Systems with Applications*. 2015. Vol. 42, N 1. P. 584–595.
11. Shylo V.P. The method of global equilibrium search. *Cybernetics and Systems Analysis*. 1999. Vol. 35, N 1. P. 68–74.
12. Шило В.П., Шило О.В., Рощин В.А. Метод глобального равновесного поиска решения задачи о максимальном взвешенном разрезе графа. *Кибернетика и системный анализ*. 2012. № 4. С. 101–105.
13. James T., Rego C., Glover F. A cooperative parallel tabu search algorithm for the quadratic assignment problem. *European J. of Oper. Res*. 2009. Vol. 195, N 3. P. 810–826.
14. Misevicius A., Kilda B. Iterated tabu search: An improvement to standard tabu search. *Information Technology and Control*. 2006. Vol. 35, N 3. P. 187–197.
15. Misevicius A. An improved hybrid genetic algorithm: New results for the quadratic assignment problem. *Knowledge Based Systems*. 2004. Vol. 17, N 2–4. P. 65–73.
16. Benlic U., Hao J.K. Breakout local search for the quadratic assignment problem. *Applied Math. and Computation*. 2013. Vol. 219, N 9. P. 4800–4815.
17. Stutzle T. Iterated local search for the quadratic assignment problem. *European J. of Oper. Res*. 2006. Vol. 174, N 3. P. 1519–1539.
18. Шило В.П., Коренкевич Д.Є., Ляшко В.І. Про оптимізаційну задачу на перестановках. *Наукові записки НаУКМА. Сер.: Комп'ютерні науки*. 2008. Т. 86. С. 21–24.

Надійшла до редакції 22.12.2016

П.В. Шило

ПОВТОРЮВАНИЙ ІТЕРОВАНИЙ АЛГОРИТМ ТАБУ ДЛЯ РОЗВ'ЯЗАННЯ КВАДРАТИЧНОЇ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ

Анотація. Розроблено новий алгоритм повторюваного табу для розв'язання квадратичної задачі про призначення. Проведене порівняльне дослідження цього алгоритму з найкращими на даний час алгоритмами розв'язання цієї задачі показало його конкурентоспроможність як за швидкодією, так і за можливістю отримання кращих розв'язків.

Ключові слова: квадратична задача про призначення, табу, обчислювальний експеримент, порівняльне дослідження алгоритмів.

P.V. Shylo

SOLVING THE QUADRATIC ASSIGNMENT PROBLEM BY THE REPEATED ITERATED TABU SEARCH METHOD

Abstract. A novel Repeated Iterated Tabu Search for quadratic assignment problem is presented. We compare our approach to the state-of-the-art techniques and demonstrate its advantages with respect to run times and solution quality.

Keywords: quadratic assignment problem, tabu search, computing experiment, a comparative study of algorithms.

Шило Петр Владимирович,
младший научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев,
e-mail: petershylo@gmail.com.