УДК 519.854

Шило В. П., Коренкевич Д. Є., Ляшко В. І.

ПРО ОПТИМІЗАЦІЙНУ ЗАДАЧУ НА ПЕРЕСТАНОВКАХ

У роботі розглядається задача дискретної оптимізації на перестановках натуральних чисел. Пропонується метод її розв'язання, заснований на схемі глобального рівноважного пошуку. Наводяться результати обчислювальних експериментів на відомих тестових задачах.

Задачі дискретної оптимізації посідають важливе місце у сучасній математиці. Велика кількість реальних задач, що виникають останнім часом у різноманітних галузях діяльності людини, приводять до дискретних оптимізаційних задач. Серед цих задач слід виокремити оптимізаційні задачі на перестановках натуральних чисел. Вони відзначаються складністю, що пов'язана з великим обсягом простору розв'язків. Адже навіть для невеликих задач розмірності 30 кількість можливих перестановок становить 30!, а це надзвичайно велике число. Відомі приклади дискретних оптимізаційних задач на перестановках - задача про комівояжера та квадратична задача про призначення. У даній роботі ми більш детально розглянемо другу, як одну з найскладніших для розв'язання задач у цій галузі.

Квадратична задача про призначення (quadratic assignment problem – QAP) може бути сформульована таким чином: нехай $A^{n\times n} = a(i, j)$ та $B^{n\times n} = (b_{i,j})$ – числові матриці, де $a_{i,j}, b_{i,j} \in R^+$, $p:\{1, \ldots, n\} \to \{1, \ldots, n\}$ – деяка перестановка натуральних чисел. Визначимо вартість перестановки як

$$c(p) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j} b_{p(i)p(j)}.$$

Необхідно знайти перестановку $p \in \prod_n$, що мінімізує вартість призначення, тобто $\min\limits_{p\in\prod_n} c(p)$, де $\prod_n$ – множина усіх перестановок чисел $\{1, \ldots, n\}$. Відомо, що квадратична задача про призначення є NP-повною [1].

Існує багато практичних задач, які можуть бути сформульовані у вигляді квадратичної задачі про призначення. Наприклад, задачі, пов'язані зі складанням розкладів, розміщенням електронних чипів на платі, аналізом статистичних даних, балансуванням турбін, аналізом зображень та ін. [2].

У той час як деякі окремі NP-повні задачі дискретної оптимізації можуть бути розв'язані точно для достатньо великих розмірностей, як це ілюструє задача про комівояжера, приклади квадратичної задачі про призначення розмірностей більших ніж 20 не піддаються точному розв'язанню. На практиці велика кількість реальних задач приводять до задач QAP значних розмірностей, які не можуть бути розв'язані точно [2]. Використання наближених методів є на даний час фактично єдиним способом розв'язання проблеми.

Загальна схема алгоритму

Велика кількість дослідників займалася розв'язанням квадратичної задачі про призначення. Для неї було запропоновано багато евристик та наближених алгоритмів. Серед них слід відзначити методи табу, відпалу, мурашиних колоній, ітеративного локального пошуку, генетичні алгоритми [3; 8].

У даній роботі запропоновано алгоритм розв'язання квадратичної задачі про призначення, що ґрунтується на схемі глобального рівноважного пошуку (ГРП) [4]. Цей алгоритм є суперпозицією методу ГРП та методу, заснованого на локальному пошуку, для поліпшення отриманих розв'язків. Як локальний пошуковий метод ми використовували метод табу, як один із найбільш ефективних методів розв'язання QAP [5]. Для прискорення роботи алгоритму використовувалась RESTART-технологія [3]. Розроблений алгоритм показав для великої серії задач результати, кращі за результати найбільш ефективних алгоритмів, що застосовувалися для розв'язання квадратичної задачі про призначення.

У загальній схемі методу ГРП використовується ідея вибору розв'язків на основі Больцманівського розподілу. Розглянемо задачу дискретної оптимізації наступного вигляду: $\min\{f(x) : x \in \prod_n\}$, де $f : \prod_n \to R$ – цільова функція, $\prod_n$ – множина перестановок натуральних чисел $\overline{1,n}$. Розглянемо деяку множину розв'язків $S \subset \prod_n$. Для множини S визначимо статистичну суму $Z(S) = \sum_{x\in S} e^{-\mu f(x)}$, де $\mu \in (0, +\infty)$ – числовий коефіцієнт. Введемо випадковий вектор ξ, розподілений таким чином:

$$P(\xi = x \mid x \in S) = e^{-\mu\cdot f(x)} / Z(S).$$

Розв'язки із множини S будемо вибирати відповідно до значення вектора ξ. Очевидно, що даний розподіл аналогічний класичному розподілу Больцмана.

У запропонованому нами алгоритмі реалізована концепція «інтенсифікація - диверсифікація». При високих значеннях температури (малому параметрі μ) розв'язки вибираються з високою мірою хаотичності, тобто майже рівноврогідно. На цьому етапі алгоритм збирає інформацію про простір розв'язків та локалізує перспективні ділянки. При зменшенні температури (збільшенні параметра μ) збільшується вірогідність вибору добрих розв'язків, і пошук фокусується в околах цих розв'язків.

Визначимо елітну множину розв'язків як множину (її потужність визначається вхідним параметром алгоритму) кращих розв'язків, знайдених алгоритмом з моменту старту. Робота алгоритму є послідовністю температурних циклів. У кожному з них проводиться кілька ітерацій, що складаються з трьох фаз, - робота з множиною елітних розв'язків (EliteSet) на основі схеми ГРП, диверсифікація отриманих розв'язків для більш широкого вивчення околів цих розв'язків і, нарешті, поліпшення отриманих розв'язків за допомогою локального пошукового методу. У нашій реалізації як локальний метод використовувався метод табу.

Детальний опис алгоритму ГРП (GES)

Вхід: *A, B* - матриці відстаней і потоків, *n* - розмірність задачі, *EliteSize* - потужність елітної множини, *Delay* - параметр RESTART-процедури, T - кількість температурних фаз, Q - кількість ітерацій у кожній фазі, κ - параметр розрахунку коефіцієнта μ, a_1, a_2 - параметри списку табу, t - кількість ітерацій процедури пошуку табу.

Вихід: p_{best} - краща знайдена перестановка.

procedure GES:

1. *EliteSet* ← EliteAssign(?);
2. p_{best} ← best solution in *EliteSet*;
3. μ = 0; *m*1 = [*a*1·*n*]; *m*2 = [*a*2·*n*]; *m* = *m*1; *maxn* = *n*/2;
4. **for** *i* = 0 **to** *T* **do**
5. **if** there wasn't solution improvements over last *Delay* iterations **then**
6. Restart (?);
7. **end if**
8. **for** *j* = 0 **to** *maxn* **do**
9. *p** ← ChooseCandidate (*EliteSet*);
10. Mutation (*p**, *m*);
11. TabuSearch (*p**, p_{best}, *t*);
12. EliteAdd (*p**);
13. **if** c(*p**) < c (p_{best}) **then**
14. p_{best} = *p**; *m* = *m*1;
15. **end if**
16. **end for**
17. TempRecalculate (μ*koeff*);
18. **if** (*m* ≥ *m*2) **then**
19. *m* = *m*1;
20. **else** *m* = *m* + 1;
21. **end if**
22. **end for**
23. **return** p_{best}**;**

procedure EliteAssign(.):

1. *c*(p_{best}) = ∞;
2. *Z* = 0;
3. **for** *I* = 0 **to** *EliteSize* **do**
4. *p* ← generate random solution;
5. TabuSearch (*p*, p_{best}, *t*); //improving generated solution with the taboo search//
6. **if** *c*(*p*) < *c*(p_{best}) **then**
7. p_{best} = *p*;
8. **end if**
9. *EliteSet* = *EliteSet* + {*p*}; //adding obtained solution to elite set//
10. *Z* = *Z* + exp (– μ · *c* (*p*)); //updating statistical sum//
11. **end for;**
12. **return** *EliteSet;*

procedure ChooseCandidate (*Z*, EliteSet):

1. *random* = unif(*Z*); //uniformly generating a number from (0; *Z*)//
2. *Z** = 0; //support statistical sum//
3. **for** *i* = 0 **to** *EliteSize* **do**
4. pick solution *p* from *EliteSet*;
5. *Z** = *Z** + exp (– μ · *c* (*p*)); //updating support statistical sum//
6. **if** *random* ≤ Z* **then**
7. **return** *p*;
8. **end if**
9. **end for**

procedure EliteAdd(*p*, *pold*):

1. **If** *EliteSet* ∩ {*p*} = ø **and** *c*(*p*) ≤ *c* (*pold*) **then**
2. *EliteSet* = *EliteSet* / {p_{old}}; *EliteSet* = *EliteSet* + {*p*};
3. Z = Z – exp (– μ · *c* (p_{old})); + exp (– μ · *c* (*p*));

procedure Diversification (*p*, *m*):

1. **for** k = 0 **to** m **do**
2. choose randomly uniformly *i* and *j* from {1, …, *n*};
3. swap *i*-th and *j*-th components in *p*;
4. **end for**
5. **return** *p*;

procedure Restart():

1. randomly uniformly choose a solution *x** ∈ *EliteSet*
2. **for each** solution *x* ∈ *EliteSet*;
3. **if** *c* (*x*) > *c* (*x**) **then** remove *x* from the *EliteSet;*
4. complete the *EliteSet* with new randomly generated solutions;

procedure TempRecalculation():

1. find $\mu_{updated}$ provides $Z(\mu_{updated})$ $k \cdot Z(\mu)$;
2. μ $\mu_{updated}$; *Z* $Z(\mu_{updated})$.

Результати експериментальних досліджень

Нами проведено серію обчислювальних експериментів на відомих задачах з бібліотеки QAPLIB (http://www.opt.math.tu-graz.ac.at/qaplib/). Наш алгоритм порівнювався з алгоритмами робастного пошуку табу (RoTaboo), мурашиної колонії (Ant Colonie) та методом відпалу (SA), коди яких також можна знайти у бібліотеці QAPLIB.

У нашій реалізації використовувалися такі значення параметрів:

EliteSize = 2*n*;
Q = *n*;
Delay = *n*/2;
a_1 = 0,2;
a_2 = 0,4;
k = 0,8;
t = *n*/2.

Ці значення не є оптимальними, і для кожної задачі можна дібрати набір параметрів, який дасть більш високі результати, ніж отримані нами. Ми лише прагнемо продемонструвати роботу алгоритму на широкому колі задач і показати, що запропонована нами модифікація методу ГРП дає кращі результати для більшості прикладів QAP.

Кожен алгоритм для кожної задачі запускався 10 разів. Порівнювалися середні за 10-ма запусками відхилення кращого знайденого алгоритмом розв'язку від найкращого відомого розв'язку:

$$\delta_f = (f_{\text{краще знайдене}} - f_{\text{найкраще відоме}}) / / f_{\text{найкраще відоме}} * 100\ \%.$$

Порівняння на реальних задачах (real-life like problems)

In-stance	*n*	BKS	Ro-Tabu	SA	Ant Colony	GES	*t, s*
			<δf>	<δf>	<δf>	<δf>	
bur26a	26	5 426 670	0,03	0,52	**0,02**	0,03	0,1
bur26b	26	3 817 852	0,09	0,32	**0,03**	0,07	0,1
bur26c	26	5 426 795	0,03	0,29	**0**	**0**	0,1
bur26d	26	3 821 225	0,05	0,10	**0**	**0**	0,1
bur26e	26	5 386 879	0,01	0,18	**0**	**0**	0,1
bur26f	26	3 782 044	0,01	0,06	**0**	**0**	0,1
bur26g	26	10 117 172	0,01	0,30	**0**	**0**	0,1
bur26h	26	7 098 658	0,13	0,13	**0**	**0**	0,1
chr25a	25	3 796	4,23	45,5	1,24	**0**	2
nug30	30	6 124	0,04	5,29	0,15	**0**	2
kra30a	30	88 900	**0**	3,77	0,70	**0**	4
kra30b	30	91 420	0,01	3,59	0,04	**0,01**	4
tai64c	64	1 855 928	0,37	1,28	**0**	**0**	1
tai20b	20	122 455 319	0,05	8,75	0,09	**0**	0,1
tai25b	25	344 355 646	0,02	2,82	**0**	**0**	0,5
tai30b	30	637 117 113	0,04	2,66	**0**	**0**	1
tai35b	35	283 315 445	0,1	3,09	**0**	**0**	2
tai40b	40	637 117 113	0,43	2,12	0,11	**0**	2
tai50b	50	458 821 517	1,58	0,57	0,26	**0**	8
tai60b	60	608 215 054	1,05	0,66	0,32	**0**	20
tai80b	80	818 415 043	0,84	1,43	0,94	**0,12**	40

У наведених таблицях: Instance - назва задачі, *n* - розмірність задачі, BKS - найкращий відомий розв'язок, $<\delta_f>$ - середнє за 10-ма запусками відхилення кращого знайденого розв'язку від найкращого відомого, *t* - час, відведений алгоритмам для роботи.

Порівняння на випадково згенерованих задачах (random problems)

In-stance	*n*	BKS	Ro-Tabu	SA	Ant Colony	GES	*t, s*
				<δf>	<δf>	<δf>	<δf>
tai20a	20	703 482	**0,05**	0,55	0,44	0,06	2,5
tai25a	25	1 167 256	**0**	0,93	1,5	**0**	5
tai30a	30	1 818 146	0,29	0,47	0,93	**0,03**	7,5
tai35a	35	2 422 002	0,63	0,86	1,14	**0,16**	10
tai40a	40	3 139 370	0,78	1,01	1,43	**0,30**	30
tai50a	50	4 941 410	1,04	1,37	1,75	**0,64**	45
tai60a	60	7 205 962	1,17	1,25	1,94	**0,84**	60
tai80a	80	13 546 960	1,28	1,07	1,39	**0,62**	20

Аналіз результатів обчислювальних експериментів дає можливість зробити висновок про високу ефективність запропонованого алгоритму, який майже для кожної задачі показав найкращі результати. Це, в сукупності з іншими успішними застосуваннями схеми ГРП для задач дискретної оптимізації, виконаними за останні роки, свідчить про те, що метод глобального рівноважного пошуку є надзвичайно ефективним інструментом для розв'язання подібних задач. Універсальність і можливість використання у поєднанні з іншими алгоритмами роблять його одним із найбільш потужних методів дискретної оптимізації.

1. *Misevicius A., Blonskis J.* Experiments with tabu search for random quadratic assignment problem // Information technology and control, 2005.
2. *Drezner Z., Hanh P., Taillard E.* Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods // Annals of operations research, 2005.
3. *Сергиенко И. В., Шило В. П.* Задачи дискретной оптимизации. - К.: Наук. думка, 2003.
4. *Шило В. П.* Метод глобального равновесного поиска // Кибернетика и системный анализ. - 1999. - № 1.
5. *Misevicius A., Lenkevicius A., Rubliauskas D.* Iterated tabu search: an improvement to standart tabu search // Information technology and control, 2006.
6. *Misevicius A.* A tabu search algorithm for the quadratic assignment problem // Computational optimization and applications, 2005.
7. *Taillard E.* Robust taboo search for the quadratic assignment problem // Parallel Computing, 1991.
8. *Tseng L., Liang S.* A hybrid metaheuristic for the quadratic assignment problem // Computational optimization and applications, 2005.

V Shylo, D. Korenkewich, V. Lyashko

OPTIMIZATION PROBLEM ON PERMUTATIONS

Discrete optimization problem on permutations of integers is being considered in the paper. The method for it 's solving based on the global equilibrium search method is proposed. Computational experiments results made on well-known instances are provided.