Јозеф Кратица
Математички институт САНУ
Јелена Којић, Александар Савић
Математички факултет у Београду

Вишедимензиони проблем поделе скупа на две партиције

Литература

[1] Kojić J. Integer linear programming model for multidimensional two-way number partitioning problem. Comput. Math. Appl. Vol. 60, pp. 2302-2308 (2010)

[2] Pop P.C., Matei O. A memetic algorithm approach for solving the multidimensional multi-way number partitioning problem. Appl. Math. Modelling Vol. 37, No. 22, pp. 9191-9202 (2013)

[3] Kratica J., Kojić J., Savić A., Two Metaheuristic Approaches For Solving Multidimensional Two-Way Number Partitioning Problem. Comput. & Oper. Res. Vol. 46, pp. 59-68 (2014)

Дефиниција основног проблема

- Уопштење познатог НП-тешког проблема поделе скупа на две партиције
- Минимизује се разлика сума тежина по партицијама
- Основни проблем:
- Дат је $S=\{1, \ldots, n\}$ и $w:S\rightarrow[0,\infty)$. Поделити га на $S_1, S_2 \subset S$, $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = S$

$$\min \left| \sum_{i\in S_1} w(i) - \sum_{i\in S_2} w(i) \right|$$

Дефиниција уопштеног проблема

- Уместо бројева w су вектори
- Минимизује се највећа разлика сума тежина по партицијама
- Уопштени проблем:
- Дат je $S=\{1, \ldots, n\}$ и $w:S\to[0,\infty)^k$. Поделити га на $S_1, S_2 \subset S$, $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = S$

$$\min\left(\max_{1\le l\le k}\left|\sum_{i\in S_1} w(i)_l - \sum_{i\in S_2} w(i)_l\right|\right)$$

Пример

$S = \{(2 \;\; 6),\ (3 \;\; 4),\ (8 \; 10),\ (5 \;\; 6)\}$ For S_1 and S_2 we have several candidates:

- $S_1 = \{(2 \;\; 6)\}$, $S_2 = \{(3 \;\; 4),\ (8 \; 10),\ (5 \;\; 6)\}$, sums are: $(2 \;\; 6), (16 \;\; 20)$, and the difference is $(14 \;\; 14)$, $z = 14$ $(max = 14)$
- $S_1 = \{(3 \;\; 4)\}$, $S_2 = \{(2 \;\; 6),\ (8 \; 10),\ (5 \;\; 6)\}$, sums are: $(3 \;\; 4), (15 \;\; 22)$, and the difference is $(12 \;\; 18)$, $z = 18$
- $S_1 = \{(8 \; 10)\}$, $S_2 = \{(2 \;\; 6),\ (3 \;\; 4),\ (5 \;\; 6)\}$, sums are: $(8 \; 10), (10 \;\; 16)$, and the difference is $(2 \;\; 6)$, $z = 6$
- $S_1 = \{(5 \;\; 6)\}$, $S_2 = \{(2 \;\; 6),\ (3 \;\; 4),\ (8 \; 10)\}$, sums are: $(5 \;\; 6), (13 \;\; 20)$, and the difference is $(8 \;\; 14)$, $z = 14$
- $S_1 = \{(2 \;\; 6),\ (3 \;\; 4)\}$, $S_2 = \{(8 \; 10),\ (5 \;\; 6)\}$, sums are: $(5 \;\; 10), (13 \;\; 16)$, and the difference is $(8 \;\; 6)$, $z = 8$
- $S_1 = \{(2 \;\; 6),\ (8 \; 10)\}$, $S_2 = \{(3 \;\; 4),\ (5 \;\; 6)\}$, sums are: $(10 \;\; 16), (8 \;\; 10)$, and the difference is $(2 \;\; 6)$, $z = 6$
- $S_1 = \{(2 \;\; 6), (5 \;\; 6)\}$, $S_2 = \{(3 \;\; 4),\ (8 \; 10))\}$, sums are: $(7 \;\; 12), (11 \;\; 14)$, and the difference is $(4 \;\; 2)$, $z = 4$

The minimum of the maximal elements is in the seventh case. It means that the optimal solution is: $S_1 = \{(2 \;\; 6),\ (5 \;\; 6)\}$, $S_2 = \{(3 \; 4),\ (8 \;\; 10)\}$, $min\ z = 4$.

Примене

Examples for MDTWNPP:

1. Two travel agents need to have as closely similar revenues from organized tours as possible; the parameters under consideration are the price of the tour, the mileage (due to petrol consumption), special offers (such as restaurants, shopping tours, visits to museums, and so on). This is a three-dimensional, two-way number partitioning problem.

2. Police patrols need to have similar mileage, degree of risk (depending on criminal activities), time for touring their beat.

МИЛП модел

Let w_{ij} be the j-th coordinate of the i-th element in the set S, and

$$s_j = \sum_{i=1}^{n} w_{ij}, \quad j = 1, ..., p. \tag{1}$$

If the variable is

$$y_i = \begin{cases} 1, & i \in S_1 \\ 0, & i \in S_2 \end{cases} \tag{2}$$

and the variable z denotes the greatest difference in the sums per the coordinates, i.e.

$$z = max| \sum_{i \in S_1} w_{ij} - \sum_{i \in S_2} w_{ij}|, \qquad j = 1, ..., p \tag{3}$$

The mathematical expression of the ILP formulation is:

$$\min z \tag{4}$$

subject to:

$$-0.5 \cdot z + \sum_{i=1}^{n} w_{ij} \cdot y_i \le 0.5 \cdot s_j, \qquad j = 1, ..., p \tag{5}$$

$$0.5 \cdot z + \sum_{i=1}^{n} w_{ij} \cdot y_i \ge 0.5 \cdot s_j, \qquad j = 1, ..., p \tag{6}$$

$$z \in [0, +\infty), y_j \in \{0, 1\}, \qquad i = 1, ..., n \tag{7}$$

Метода променљивих околина

•

Algorithm 1: VNS pseudo code

Algorithm `Function VNS(`k_{min}`,` k_{max}`,` t_{max}, p_{move}`)`
$x \leftarrow Random\_Init()$ `(starting point` x `is randomly generated)`
$k \leftarrow k_{min}$
while $t < t_{max}$ **do**
  $x' \leftarrow Shaking(x, k)$
  $x'' \leftarrow Local\_Search(x')$
  if $Compare(x, x'', p_{move})$ **then**
    $x \leftarrow x''$
  else
    if $k < k_{max}$ **then**
      $k \leftarrow k + 1$
    else
      $k \leftarrow k_{min}$

Algorithm 2: Pseudo code of function Compare

Algorithm `Function Compare(`x`,` x''`,` p_{move}`)`
if $Obj(x'') < Obj(x)$ **then**
  `return true`
else
  if $Obj(x'') > Obj(x)$ **then**
    `return false`
  else
    `return` $Random(0, 1) < p_{move}$

Метода промењљивих околина

- Решење је низ од n бинарних цифара: 1 → S_1, 0→ S_2
- Функција циља се рачуна по дефиницији
- Околина N_k је скуп свих низова који се од текућег разликују на к места
- Брза процедура локалног претраживања заснована на 1-замени
  - Све међусуме се памте, па уместо поновног израчунавања само се врши ажурирање које је много брже
- $к_{min}=2$
- Критеријум завршетка 600 секунди (као и ЕМ)

$$k_{max} = \min\{30, \lfloor |S|/4 \rfloor\}.$$

Метода заснована на електромагнетизму

Algorithm 3: EM pseudo code

```
Algorithm EM
Random_Init()
iter ← 0
while iter < max_iter do
    iter ← iter + 1
    foreach point p_k in solution_set do
        Objective_function(p_k)
        Local_search(p_k)
        Scale_solution(p_k)
    Compute_charges()
    Compute_forces()
    Move()
```

Метода заснована на електромагнетизму

- Решење: n реалних бројева: $\geq 0.5 \rightarrow S_1$, $< 0.5 \rightarrow S_2$
- Функција циља се рачуна по дефиницији
- Број решења 10
- Стандардно рачунање наелектрисања
  - Добра решења привлаче остала
  - Лоша решења одбијају остала
- Посебна процедура скалирања
- Иста брза процедура локалног претраживања заснована на 1-замени

EM детаљи

In this process, each sample point is considered as a charged particle. As such, the charge of each sample point is initially calculated. The amount of charge, given in (9), relates to the value of the objective function at that point which also determines the magnitude of attraction or repulsion of the point over the sample population. Label Obj_i indicates objective value for i-th sample point, after the local search.

$$q_i = \exp\left(-n \frac{Obj_i - Obj_{best}}{\sum_{k=1}^{m} Obj_k - Obj_{best}}\right) \tag{9}$$

According to the superposition principle of the electromagnetism theory, the force exerted on a point via another point is inversely proportional to the distance between the points and directly proportional to the product of their charges. Mathematically, the power of attraction or repulsion of charges is calculated as follows:

$$\begin{aligned} F_i &= \sum_{j=1, j\neq i}^{m} F_i^j, \; where \\ F_i^j &= \begin{cases} \left(\frac{q_i q_j}{\|p_j - p_i\|^2}\right) \cdot (p_j - p_i), & Obj_j < Obj_i \\ \left(\frac{q_i q_j}{\|p_j - p_i\|^2}\right) \cdot (p_i - p_j), & Obj_j \geq Obj_i \end{cases} \end{aligned} \tag{10}$$

where $\|p_i - p_j\|$ is the Euclidean distance between EM points p_i and p_j.

ЕМ детаљи

Algorithm 4: Move pseudo code

Function `Move`()

foreach *point* p_k *in solution_set* **do**

  if $p_k \neq p_{best}$ **then**

    $\beta \leftarrow Random[0,1]$

    $F_i \leftarrow F_i / \|F_i\|$

    for $i = 1$ *to* n **do**

      if $F_{ki} > 0$ **then**

        $p_{ki} \leftarrow p_{ki} + \beta \cdot F_{ki} \cdot (1 - p_{ki})$

      else

        $p_{ki} \leftarrow p_{ki} + \beta \cdot F_{ki} \cdot p_{ki}$

Експериментални резултати

Table 1: Comparison of the results on instances with $n = 50$

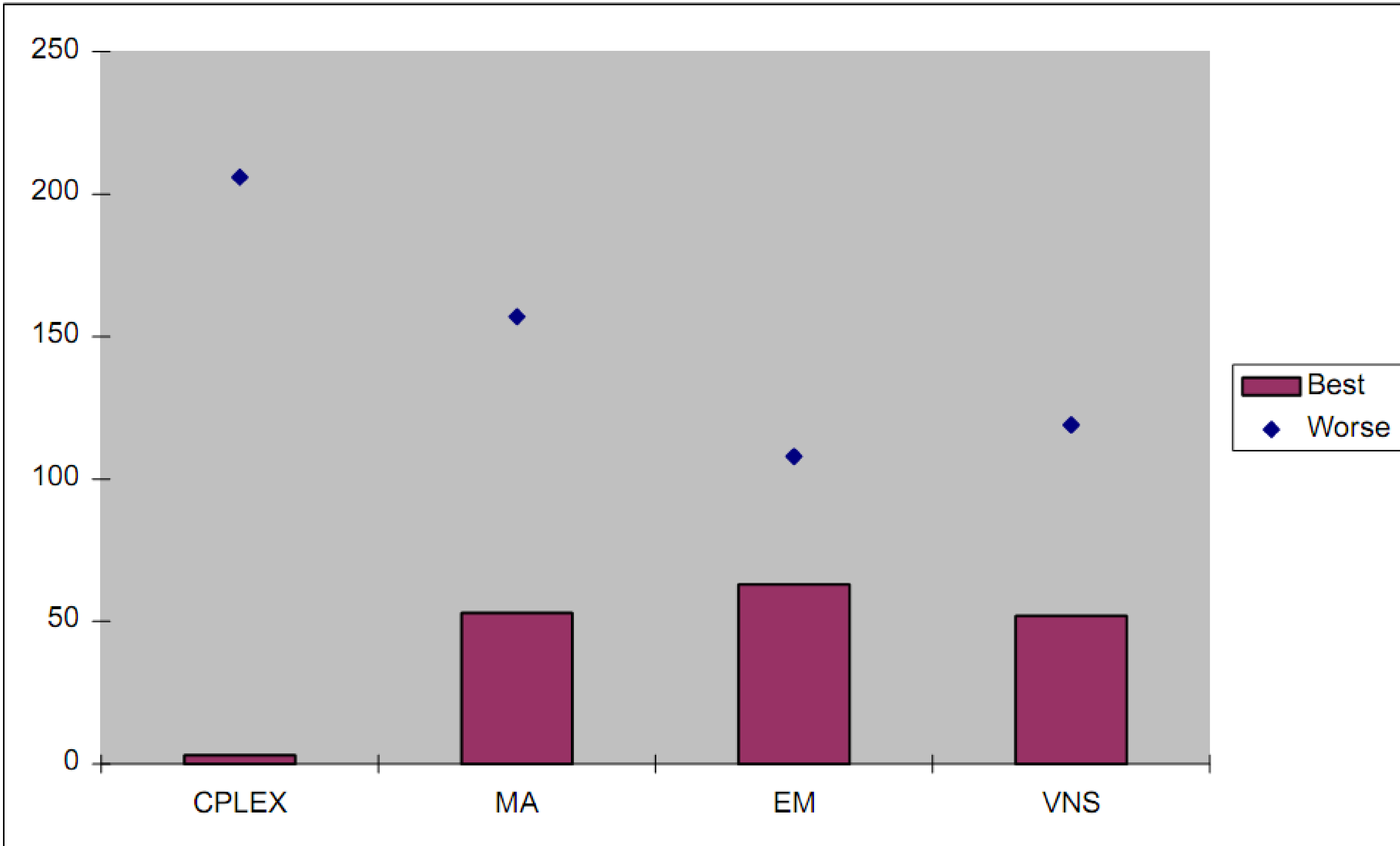
Inst	*CPLEX*		*MA*		*EM*		*VNS*	
	sol	*t*	*sol*	*t*	*sol*	*t*	*sol*	*t*
50_2a	3.204	552.60	5.438	35.73	0.450	357.60	0.450	290.95
50_2b	9.193	63.75	9.193	32.38	3.089	323.33	1.620	408.21
50_2c	9.191	161.90	9.191	38.98	2.483	246.44	3.087	354.15
50_2d	4.753	92.01	4.753	42.65	1.345	339.46	1.347	397.34
50_2e	6.719	1340.84	7.241	44.71	3.541	326.55	3.671	382.88
50_3a	303.581	465.91	283.563	162.34	285.269	357.64	288.317	343.05
50_3b	350.828	430.88	314.392	165.28	305.682	362.10	305.682	426.96
50_3c	152.089	774.38	156.287	168.62	139.883	292.38	139.885	393.20
50_3d	102.516	1378.31	101.283	172.27	96.416	245.45	96.438	184.68
50_3e	217.903	312.16	198.854	171.64	214.847	328.53	211.795	379.72
50_4a	909.765	1663.43	783.046	273.31	906.715	233.54	903.663	368.47
50_4b	1272.224	1581.08	1017.283	261.94	1219.544	257.04	1219.542	347.37
50_4c	461.161	1275.95	456.392	287.36	455.057	389.50	455.057	386.35
50_4d	1024.681	827.70	987.270	279.02	1018.575	298.42	1015.523	326.14
50_4e	1199.574	498.30	1187.273	281.07	1205.676	378.75	1205.676	389.60
50_5a	926.164	281.77	918.278	286.45	921.975	333.68	917.008	408.90
50_5b	3202.875	1192.94	3002.869	290.02	2607.094	298.21	2610.144	379.66
50_5c	2696.703	143.42	2563.904	291.28	1398.771	302.72	1395.725	352.02
50_5d	2275.792	357.38	2183.634	296.43	2272.740	274.23	2272.740	350.41
50_5e	4823.935	197.43	4689.382	296.45	3652.517	312.07	3652.517	285.76
50_10a	16176.578	1432.87	15722.290	452.41	16176.578	306.03	16170.474	315.50
50_10b	19560.318	5.97	19560.318	462.16	19548.104	332.61	19554.210	355.10
50_10c	17757.097	1308.68	15823.832	467.82	14125.537	328.74	14134.693	352.40
50_10d	14925.023	1104.70	14925.023	425.76	14921.973	346.76	14921.979	358.59
50_10e	15369.009	472.44	14527.381	476.45	15362.905	336.90	15359.853	394.05
50_15a	33208.019	1777.83	30728.546	683.21	33211.069	333.33	33214.123	388.46
50_15b	35003.301	1121.87	34362.391	674.57	33240.094	279.94	33240.094	413.45
50_15c	29920.923	1443.76	28736.382	684.65	29456.854	282.72	29456.854	363.16
50_15d	21652.841	1594.98	20356.836	690.25	33085.975	278.11	21655.891	367.04
50_15e	31800.690	332.25	29018.285	692.46	31800.692	288.49	31800.692	353.43
50_20a	52826.340	71.83	50647.836	635.26	52826.340	282.91	52826.340	440.04
50_20b	51917.902	1378.83	50382.384	657.41	51917.902	318.15	51917.902	420.21
50_20c	50560.864	493.27	51829.374	624.67	50560.864	291.38	50560.864	374.02
50_20d	53955.965	166.26	51538.574	635.82	53955.965	282.12	53955.965	347.41
50_20e	48281.499	234.75	47829.865	652.36	48281.499	289.22	48281.499	295.63

Експериментални резултати

Table 6: Comparison of the results on instances with $n = 500$

Inst	*CPLEX*		*MA*		*EM*		*VNS*	
	sol	*t*	*sol*	*t*	*sol*	*t*	*sol*	*t*
500_2a	0.620	1745.39	1.354	563.71	0.520	349.88	0.122	428.14
500_2b	1.879	1797.03	1.879	482.21	0.735	301.00	0.467	362.96
500_2c	2.013	1076.04	2.013	498.28	0.580	287.27	0.666	397.19
500_2d	1.003	1687.53	1.003	421.26	0.190	372.66	0.404	370.34
500_2e	0.913	1726.60	0.913	382.48	0.487	335.77	0.318	377.58
500_3a	213.539	1698.09	124.540	524.42	3.044	354.73	1.517	378.26
500_3b	142.160	806.88	132.878	502.22	57.643	335.54	56.234	438.32
500_3c	270.729	466.64	219.763	522.79	68.875	348.06	72.892	456.76
500_3d	13.736	1134.38	121.321	520.04	4.578	389.52	1.526	399.73
500_3e	7.625	1091.25	12.453	381.33	1.529	432.53	1.525	440.52
500_4a	1562.920	171.29	1232.920	567.08	3.046	428.94	1.534	469.14
500_4b	1186.722	429.41	977.271	465.59	845.267	386.51	895.144	459.93
500_4c	1093.756	63.74	832.244	652.82	605.996	349.19	618.206	414.48
500_4d	4.580	714.37	4.580	552.16	4.582	393.36	1.526	476.25
500_4e	1410.694	106.31	937.823	541.36	534.116	357.58	546.322	467.93
500_5a	3665.484	36.99	1903.802	615.22	7.622	379.31	3.056	445.58
500_5b	4448.741	21.29	2824.579	500.85	1841.422	372.25	1844.490	381.43
500_5c	3511.837	523.39	3839.652	590.72	1882.324	394.22	1879.286	398.99
500_5d	2597.644	81.42	1821.393	621.47	4.582	361.06	7.614	468.71
500_5e	2572.429	955.58	1816.680	579.12	847.689	399.64	817.165	366.75
500_10a	19183.301	1718.29	12892.029	637.01	12920.009	276.84	12913.909	403.32
500_10b	12161.350	128.48	10387.625	622.27	12133.878	357.87	12136.930	316.03
500_10c	16594.760	368.13	10282.295	612.28	12595.464	311.30	12598.518	317.82
500_10d	20284.381	1699.01	16328.752	678.9	12823.522	377.48	12823.524	381.75
500_10e	15548.670	1680.47	14738.344	635.37	14224.406	314.33	14212.198	356.59
500_15a	30316.775	1055.81	20332.662	682.31	26002.365	252.26	25984.057	323.77
500_15b	31878.383	1591.08	28283.569	721.62	29360.731	228.43	22258.131	390.28
500_15c	32792.472	803.77	25461.223	719.81	27134.042	331.98	27131.010	364.76
500_15d	35555.260	881.27	27283.392	724.37	26351.405	305.79	27219.244	382.34
500_15e	30806.719	455.06	21652.005	728.9	28379.732	299.95	24997.925	394.81
500_20a	48281.977	1000.12	32897.388	843.3	42278.177	233.48	38512.711	419.56
500_20b	54921.900	237.63	38467.134	818.39	36626.582	290.30	36654.048	340.80
500_20c	41578.884	1382.98	38763.093	736.9	37845.649	290.35	40581.653	386.12
500_20d	54293.200	1728.58	43637.932	792.38	38774.751	277.61	43814.720	347.85
500_20e	41092.622	1713.03	40341.827	918.22	38578.855	295.65	43064.672	295.71

Експериментални резултати



Закључак

This paper presents efficient VNS and EM approaches for solving MDTWNPP. The neighborhood structures defined for the VNS allow a simple but effective shaking procedure. The new scaling technique combined with movement based on the attraction-repulsion mechanism directs EM toward promising search regions. Local search is implemented very efficiently, which results in excellent overall performances for both VNS and EM approaches.

An extensive experimental comparison on various types of instances indicates that our VNS and EM approaches outperform in most cases the existing methods from the literature designed for solving this problem. From the results it can be concluded that EM produces slightly better results than VNS.

This research can be extended in several ways. It would be interesting to investigate the application of the presented VNS and EM approaches to similar problems. Another direction of research could be a parallelization of the presented methods and testing on a powerful multiprocessor computer.

Хвала на пажњи!!!