

AZ EMBERI ERŐFORRÁSOK MINISZTERIUMA ÚNKP-16- 3 KÓDSZÁMÚ ÚJ
NEMZETI KIVÁLÓSÁG PROGRAMJÁNAK TÁMOGATÁSÁVAL KÉSZÜLT

Diszkrét Bakteriális Memetikus Evolúciós Algoritmus az Utazó Ügynök Probléma megoldására

TÜŰ-SZABÓ BOLDIZSÁR

FÖLDESI PÉTER

KÓCZY T. LÁSZLÓ

XXXII. MAGYAR OPERÁCIÓKUTATÁS KONFERENCIA

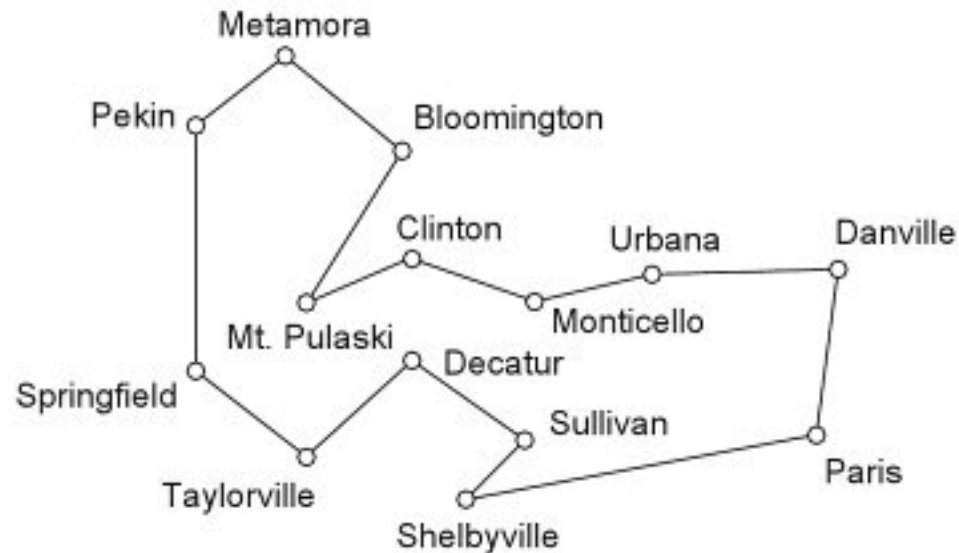
CEGLÉD, 2017. JÚNIUS 14-16.

Tartalom

- ▶ Az Utazó Ügynök Probléma (TSP)
- ▶ Diszkrét Bakteriális Memetikus Evolúciós Algoritmus
- ▶ Futtatási eredmények
- ▶ Következtetések

Az Utazó Ügynök Probléma

- ▶ Az ügynök végiglátogatja a megadott városokat, és végül visszaérkezik a kiinduló helyre → legrövidebb úthosszat keressük
- ▶ Számos alkalmazási terület (logisztika, mikrochipgyártás stb.)



Az Utazó Ügynök Probléma

▶ Gráfkeresési problémaként definiálható élsúlyokkal

$$\begin{aligned} G_{TSP} &= (V_{cities}, E_{conn}) \\ V_{cities} &= \{v_1, v_2, \dots, v_n\}, E_{conn} \subseteq \{(v_i, v_j) | i \neq j\} \\ C : V_{cities} \times V_{cities} &\rightarrow R, C = (c_{ij})_{n \times n} \end{aligned}$$

▶ Cél: megtalálni a csúcsok azon permutációját megtalálni, amelynél az úthossz minimális

$$C(i) = \left(\sum_{i=1}^{n-1} c_{p_i, p_{i+1}} \right) + c_{p_n, p_1}$$

Diszkrét Bakteriális Memetikus Evolúciós Algoritmus

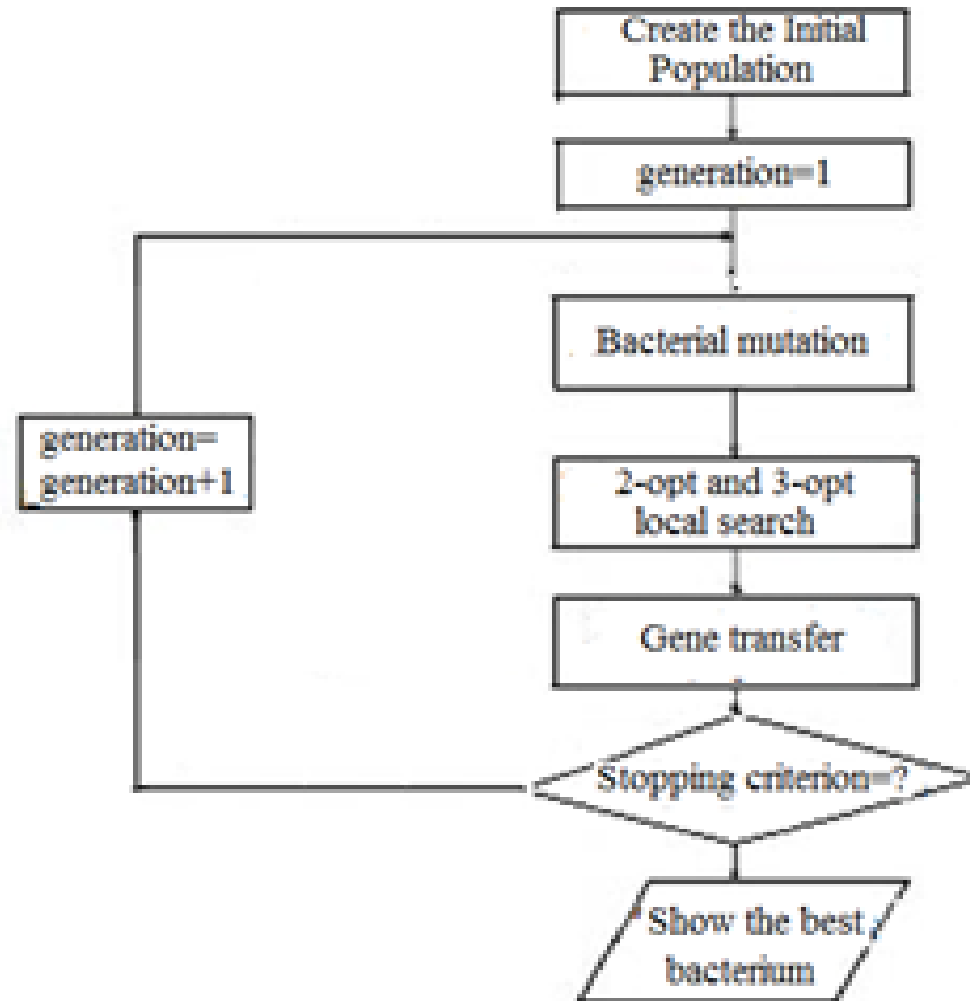
Kóczy, L.T., Földesi, P., Tüő-Szabó, B.: A Discrete Bacterial Memetic Evolutionary Algorithm for the Traveling Salesman Problem, IEEE World Congress on Computational Intelligence (WCCI 2016), Vancouver, Canada, pp. 3261–3267, 2016.

Kóczy, L.T., Földesi, P., Tüő-Szabó, B.: An effective Discrete Bacterial Memetic Evolutionary Algorithm for the Traveling Salesman Problem, In: International Journal of Intelligent Systems, 2017.

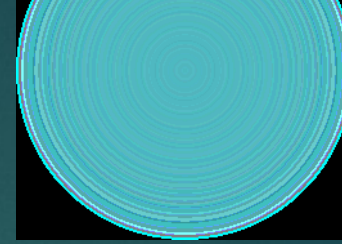
► DBMEA egy memetikus algoritmus, a bakteriális evolúciós algoritmus kibővítése lokális kereséssel (2-opt and 3-opt lokális keresés)

Balázs K, Botzheim J, Kóczy TL. Comparison of various evolutionary and memetic algorithms. Integrated uncertainty management and applications, AISC 68. Berlin, Germany: Springer-Verlag; 2010. pp. 431–442.

DBMEA folyamata

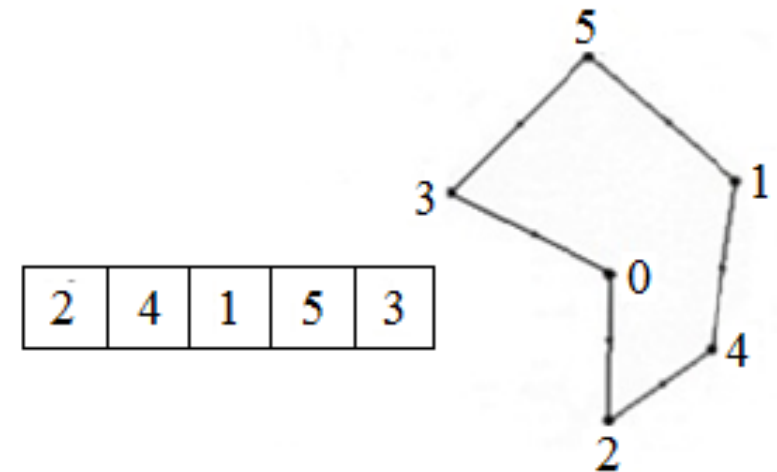


DBMEA folyamata



➤ Egyedek kódolása

Minden egyed egy lehetséges útvonal



DBMEA folyamata

▶ Kezdeti populáció TSP esetén

▶ Random egyedek

▶ Determinisztikus egyedek: 3 determinisztikus

- ▶ Nearest neighbour (NN) heurisztika: mindig a legközelebbi még meg nem látogatott város választja ki következő állomásként
- ▶ Secondary nearest neighbour (SNN) heurisztika: mindig a 2. legközelebbi még meg nem látogatott város választja ki következő állomásként
- ▶ Alternating nearest neighbour (ANN) heurisztika: az előző kettő kombinációja

DBMEA folyamata

► Bakteriális mutáció

Minden egyeden végrehajtásra kerül

Az egyed jobb vagy azonos lesz az eredeti egyeddel a bakteriális mutáció eredményeként

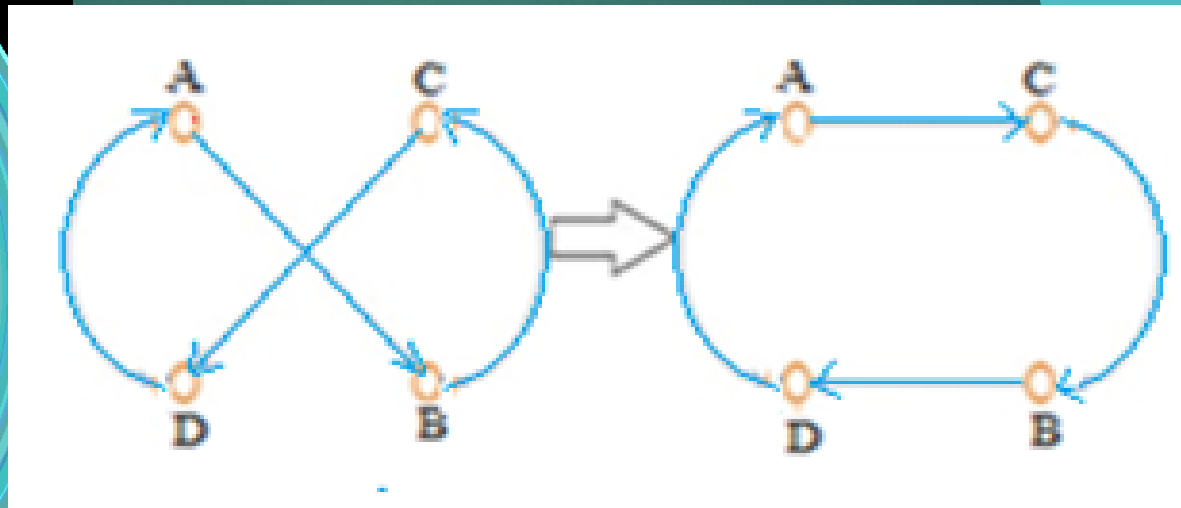
original bacterium:	4	8	3	5	6	2	9	1	7
1. clone (reverse order):	4	8	3	2	6	5	9	1	7
2. clone:	4	8	3	2	5	6	9	1	7
3. clone:	4	8	3	6	5	2	9	1	7
4. clone:	4	8	3	6	2	5	9	1	7

selection of the fittest

DBMEA folyamata

► 2-opt lokális keresés

2 él kicserélésével javítja az útvonalat

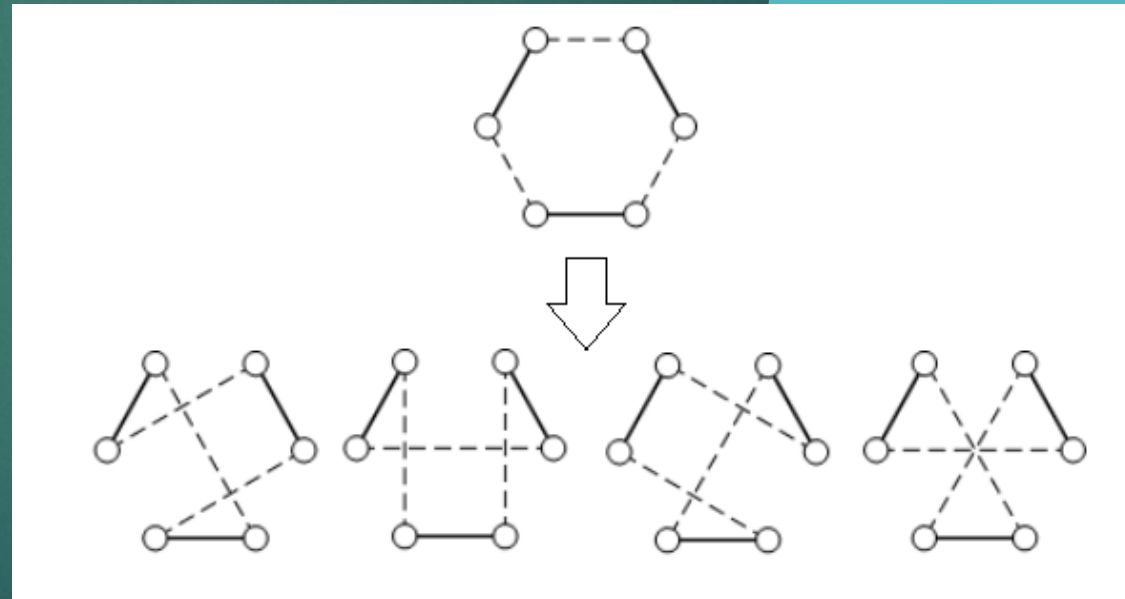


$$|AB| + |CD| > |AC| + |BD|$$

DBMEA folyamata

► 3-opt lokális keresés

3 él cseréje



DBMEA folyamata

lokális keresés gyorsítása TSP esetén:

- ▶ Candidate lists: megvizsgálandó csúcsok listája → nagy élhosszok ritkán szerepelnek az optimális útvonalban
- ▶ Don't look back bits:
 - ▶ minden csúcshoz egy bitet rendelünk hozzá (kezdetben 0 értékű)
 - ▶ 1-re állítjuk, ha az adott csúcshoz nincs olyan élcseré, amely az útvonal javulását eredményezné → következő iterációban ezt a csúcsot nem vizsgáljuk
- ▶ Fix sugarú keresés: minden u_i csúcs esetén egy fix sugarú (az útvonalban a csúcs és a vele összeköttetésben levő egyik u_j csúcs távolsága c_{u_i, u_j}) körön belül keresünk

DBMEA folyamata

► Génátadás

Információ csere az egyedek között a populációban

Source
bacterium:

4	8	3	5	6	2	9	1	7
---	---	---	---	---	---	---	---	---



Destination
bacterium:

2	5	7	3	4	1	8	6	9
---	---	---	---	---	---	---	---	---

Infected destination
bacterium:

5	6	2	9	7	3	4	1	8
---	---	---	---	---	---	---	---	---

Futtatási eredmények (TSP)

- ▶ Concorde algoritmus, Helsgaun's Lin-Kernighan kerekítéseket használ → módosítottuk, hogy valós értékeket számoljon
- ▶ Eredmények összehasonlítása a Concorde-dal futási idő és úthossz tekintetében
- ▶ Intel Pentium Dual CPU T2390 1,86GHz, 2GB RAM

Paraméterek:

- ▶ Populáció nagysága ($N_{ind}=100$)
- ▶ Klónok száma a bakteriális mutáció során ($N_{clones}=n_{cities}/25$)
- ▶ Infekciók száma a génátadás során ($N_{inf}=40$)
- ▶ Szegmensek hossza bakteriális mutációnál ($I_{seg}=n_{cities}/50$)
- ▶ Szegmenshossz a génátadásnál ($I_{trans}=n_{cities}/10$)
- ▶ candidate lista hossza (probléma méret négyzetgyöke)

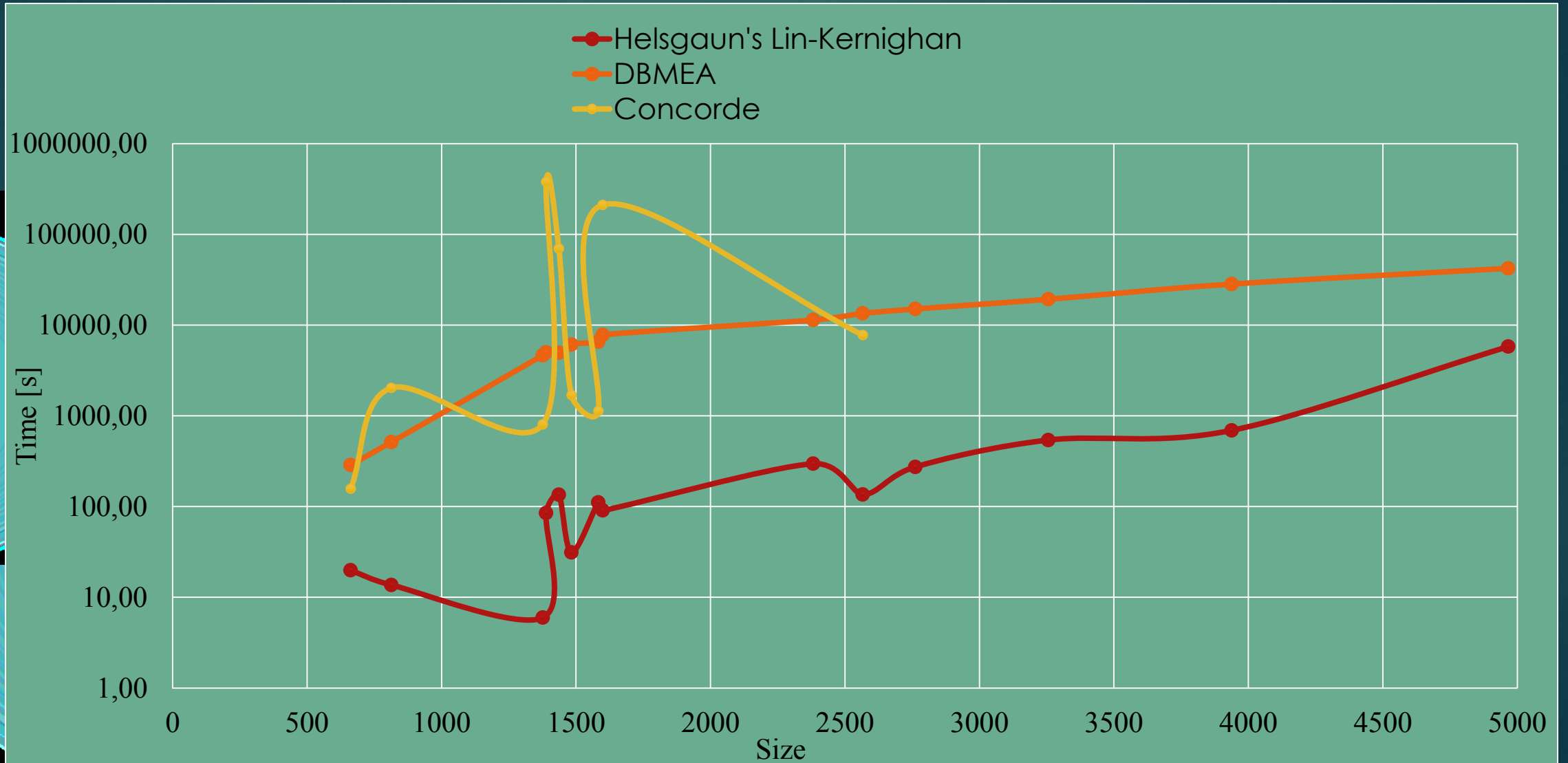
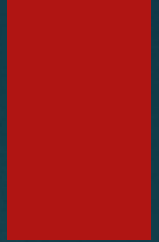
Útvonalhosszak összehasonlítása

Instance	DBMEA		Helgaun's Lin-Kernighan		Concorde	gap DBMEA [%]
	best value	avg. Value	best value	avg. Value	optimal value	
xql662	2 550,84	2 550,84	2 550,84	2 550,92	2 550,84	0
dkg813	3 243,41	3 243,41	3 243,41	3 243,68	3 243,41	0
dka1376	4 738,61	4 743,20	4 736,87	4 736,87	4 736,87	0,04
dca1389	5 156,43	5 158,06	5 152,69	5 153,08	5 152,69	0,07
dja1436	5 332,19	5 335,93	5 328,21	5 329,01	5 328,21	0,07
icw1483	4 467,22	4 472,12	4 465,88	4 465,94	4 465,88	0,03
rbv1583	5 446,09	5 449,99	5 444,67	5 444,91	5 444,67	0,03
rby1599	5 589,13	5 592,02	5 584,16	5 585,20	5 584,16	0,09
dea2382	8 154,34	8 160,93	8 146,94	8 148,87	-	0,09
pds2566	7 774,91	7 781,57	7 766,71	7 766,89	7 766,71	0,1
bch2762	8 376,18	8 380,63	8 368,56	8 370,33	-	0,09
fdp3256	10 104,36	10 108,69	10 091,29	10 092,15	-	0,13
dkc3938	12 715,99	12 723,46	12 704,22	12 704,31	-	0,09
xqd4966	15 554,91	15 560,49	15 530,82	15 531,00	-	0,16

Futási idők összehasonlítása

Instance	DBMEA		Helgaun's Lin-Kernighan		Concorde
	avg. Sec	range sec	avg. sec.	range sec	avg. Sec
xql662	287,35	256,77 - 321,59	19,87	6,83 - 51,13	157,27
dkg813	513,49	474,11 - 534,53	13,68	0,43 – 51,34	2 031,11
dka1376	4 640,81	3 934,12 - 5 494,63	5,96	0,59 - 21,6	807,11
dca1389	5 018,35	4 729,46 - 5 491,89	84,84	8,62 - 161,41	380 155,1
dja1436	4 951,08	4 528,02 - 5 305,59	135,29	105,33 – 171,01	69 811,21
icw1483	6 118,88	5 686,68 - 6 589,64	31,19	9,52 – 53,84	1 690,34
rbv1583	6 553,3	5 993,29 - 6 956,67	110,91	27,32 – 197,70	1 134,22
rby1599	7 810,28	7 177,69 - 8 541,25	90,70	49,78 – 159,14	211 312,32
dea2382	11 385,53	10 456,87 - 13 211,99	296,56	159,84 - 390,87	-
pds2566	13 456,76	12 177,71 - 15 946,74	136,13	36,41 – 251,46	102 797,26
bch2762	15 058,32	13 890,67 - 16 411,52	273,86	134,77 - 560,17	-
fdp3256	19 274,44	17 235,71 - 24 014,76	541,35	322,23 - 1 219,77	-
dkc3938	28 325,26	25 372,49 - 31 871,54	691,99	566,13 - 904,31	-
xqd4966	42 246,95	40 029,88 - 44 691,3	5790,2	3 863,83 - 7 666,99	-

Futási idők összehasonlítása



Jövőbeli tervek

1, DBMEA továbbfejlesztése

- ▶ Más lokális keresés kipróbálása
- ▶ Párhuzamosítás

2, DBMEA tesztelése a TSP különböző változatain

- ▶ Időablakos TSP

Ohlmann, J.W., Thomas, B.W.: A compressed-annealing heuristic for the traveling salesman problem with time windows, *INFORMS Journal on Computing* 19 (1), pp.80–90. 2007.

da Silva, R. F., Urrutia, S.: A General VNS heuristic for the traveling salesman problem with time windows, *Discrete Optimization* 7 (4), pp. 203-211, 2010.

Jövőbeli tervek

▶ Traveling Repairman Problem (Minimum Latency Problem)

Salehipour, A., Sörensen, K., Goos, P., Bräysy, O., 2011. Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem. 4OR: A Quarterly Journal of Operations Research 9 (2), pp. 189–209, 2011.

Silva, M.M., Subramanian, A., Vidal, T., Ochi, L.S.: A simple and effective metaheuristic for the Minimum Latency Problem, European Journal of Operational Research 221 (3), pp. 513-520, 2012.

▶ Időfüggő TSP

▶ Multi TSP és változatai

▶ Stb.

Köszönöm a figyelmet!