

SEMESTRÁLNA PRÁCA – ÚLOHA OBCHODNÉHO CESTUJÚCEHO - HEURISTKY

Optimalizácia sietí

Benjamín Koša

Žilinská univerzita v Žiline | Fakulta riadenia a informatiky

6.1.2023

Obsah

Zadanie úlohy.....	2
Popis spôsobu riešenia.....	3
Tvorenie základnej cesty	3
Simulated annealing SA	3
Popis funkcií	4
Zaver	6
Výsledky	6
Základná trasa	6
Výpis ako sa postupne trasa zlepšovala pri rôznych seedoch.....	6
Najlepšia trasa	7

Zadanie úlohy

Na vybranej testovacej matici vzdialeností riešte úlohu **obchodného cestujúceho** prideleným **heuristickým** algoritmom.

Na skonštruovanie východzej prípustnej trasy využite duálnu heuristiku podľa Vášho zadania:

1. Algoritmus zväčšovania o najbližší uzol

Algoritmus vychádza zo základnej neprípustnej trasy $i_1 - i_2 - i_3 - i_4$, ktorú v každom kroku zväčší vsunutím spracovávaného uzla medzi dva už zaradené uzly, ktoré nasledujú po sebe v súčasnej trase. Spracovávaný uzol sa vyberie z množiny doposiaľ nezaradených uzlov ako uzol, ktorý je najmenej vzdialený od už zaradených uzlov v trase (podľa **súčtového kritéria** – súčet vzdialeností kandidáta na spracovanie od každého uzla zaradeného v trase). Miesto, na ktoré je spracovávaný uzol zaradený, je určené tak, aby sa súčasná trasa zväčšila o čo najmenej.

Základnú neprípustnú trasu $i_1 - i_2 - i_3 - i_4$ určíme tak, že i_1 bude prvý uzol v zozname v zadanej sieti, i_2 bude doposiaľ nezaradený uzol, ktorý je najviac vzdialený od uzla i_1 a i_3 bude doposiaľ nezaradený uzol, ktorý je najviac vzdialený od uzla i_2 .

Takto nájdenú východziu trasu obchodného cestujúceho **zlepšite pomocou metaheuristiky Simulated Annealing** s určeným spôsobom nájdenie okolia aktuálneho riešenia (podľa zadania):

C. Zmena polohy vrchola v trase (výmena reťazcov dĺžky 1 typu neprázdny-prázdny)

Pre metaheuristiku Simulated Annealing nastavte parametre takto: teplotu nastavte na počiatočnú hodnotu $t_{max} = 10000$, maximálny počet preskúmaných prechodov od prechodu k súčasnému riešeniu $u = 40$, maximálny počet preskúmaných prechodov od poslednej zmeny teploty $q = 50$. (Popis algoritmu je v knihe „Optimalizace na dopravních sítích“, str.95-96)

Dáta BB

Popis spôsobu riešenia

Riešenie spočíva z dvoch väčších častí, prvá časť je vytvorenie základnej cesty. V druhej časti sa základnú cestu snažíme vylepšiť pomocou algoritmu simulated annealing.

Tvorenie základnej cesty

Vytvoríme si pole nezaradených vrcholov, z neho podľa zadania vyberáme vrcholy a vkladáme do vytvárajúcej cesty až kým toto pole neje prázdne.

Simulated annealing SA

Základu cestu dáme ako vstupný parameter algoritmu SA, v algoritme náhodne vyberieme vrchol z riešenia, presunieme ho na iné náhodné miesto. Spočítame zmenu dĺžky, ak nastalo zlepšenie zapíšeme si nové lepšie riešenie, ak nenastalo zlepšenie spočítame si akceptačné kritérium a ak je splnená podmienka prijmem aj horšie riešenie.

Dôvodom prečo akceptujeme aj horšie riešenie je umožniť algoritmu uniknúť lokálnym minimám a dôkladnejšie preskúmať priestor riešenia. Algoritmus sa tak môže vyhnúť uviaznutiu v suboptimálnom riešení.

Popis funkcií

Pre úlohu sme sa rozhodli použiť model funkcionálneho programovania.

```
/**
 * search for base way
 * @return
 */
public static ArrayList<Integer> getBaseWay()
```

```
/**
 * count way extension before node insert
 * @param i
 * @param j
 * @param newNode
 * @return
 */
public static int countExtension(int i, int j, int newNode)
```

```
/**
 * count improvement before change of node position
 * @param bestSolution
 * @param cityIndex
 * @param moveToIndex
 * @return
 */
public static int countImprovement(ArrayList<Integer> bestSolution, int
cityIndex, int moveToIndex)
```

```
/**
 * move node in solution
 * @param solution
 * @param cityIndex
 * @param moveToIndex
 */
public static void moveNode(ArrayList<Integer> solution, int cityIndex, int
moveToIndex)
```

```
/**
 * simulatedAnnealing algorithm
 * @param bestSolution solution
 * @param START_TEMPERATURE
 * @param COOLING_RATE
 * @param NUM_ITERATIONS
 * @param MAX_PASSES
 * @param SEED
 */
public static void simulatedAnnealing(
    ArrayList<Integer> bestSolution,
    final int START_TEMPERATURE,
    final int COOLING_RATE,
    final int NUM_ITERATIONS,
    final int MAX_PASSES,
    final int SEED)
```

```
/**
 * count acceptance probability
 * @param bestDistance
 * @param newDistance
 * @param temperature
 * @return
 */
private static double acceptanceProbability(int bestDistance, int
newDistance, double temperature)
```

```
/**
 * count whole solution distance
 * @param way
 * @return
 */
private static int countDistance(ArrayList<Integer> way)
```

Záver

Základná trasa ma dĺžku **3240**. Aby sme trasu zlepšili museli sme SA. vykonať v replikáciách. Použili sme 10 000 000 replikácií, každú replikáciu s rôznym seedom. Zistili sme že po 865369 replikáciách sme už lepšie riešenie nenašli. Po nájdení lepšej trasy sme ďalej hľadali lepšie riešenie na už vylepšenej trase. Najkratšia trasa ku ktorej sme sa dostali má dĺžku **3120**.

Výsledky

Základná trasa

Dĺžka 3240

0;1;5;2;40;41;4;3;7;6;11;8;39;38;37;13;14;142;18;17;16;19;20;139;138;137;108;508;24;21;22;23;109;110;111;112;116;115;114;113;136;143;140;141;126;48;49;171;50;10;9;12;33;44;215;188;154;187;45;504;189;42;43;153;156;155;170;157;158;163;164;165;166;167;162;184;185;169;168;161;160;54;144;60;61;70;72;71;317;316;278;315;314;286;281;280;279;277;276;76;73;102;104;103;101;319;318;320;74;77;75;80;78;82;79;81;83;91;92;99;98;117;85;84;90;106;89;105;107;86;87;94;88;93;95;96;100;97;119;120;122;135;134;133;132;131;130;124;127;118;128;129;125;123;121;58;57;59;56;55;507;52;176;173;175;214;174;64;263;262;261;265;264;258;257;289;288;287;293;307;294;292;306;308;310;309;511;311;401;400;415;416;417;414;354;399;408;406;407;405;404;413;412;411;410;409;403;402;312;305;304;302;301;298;296;295;291;290;249;297;300;303;299;244;245;246;247;248;250;510;252;256;259;251;253;254;255;283;284;274;282;67;68;69;275;272;271;503;285;270;273;269;268;266;267;66;65;63;62;506;217;218;225;230;229;231;232;233;234;216;238;494;356;357;371;372;355;358;370;385;384;359;373;365;364;362;363;512;481;483;361;480;360;369;368;374;375;376;377;397;395;392;391;390;389;444;445;448;449;446;451;450;497;453;496;498;366;367;386;513;514;387;388;437;396;434;431;432;433;435;436;394;393;398;426;425;427;428;429;440;447;441;442;439;443;438;430;452;421;422;501;420;419;418;454;423;424;378;380;382;383;381;240;242;379;243;241;239;237;236;235;228;313;260;226;227;223;224;222;221;220;219;343;344;345;340;342;341;349;347;346;348;350;462;478;476;479;477;482;473;475;474;472;470;471;484;485;489;491;495;492;493;490;486;488;500;487;499;469;468;466;467;464;465;460;463;461;352;351;353;339;337;338;328;456;457;459;458;455;336;335;334;333;332;330;331;329;327;326;325;324;323;322;321;202;201;200;199;204;203;198;197;196;195;194;146;145;180;179;148;178;147;149;152;151;150;183;181;182;193;192;191;190;509;208;206;212;205;209;210;213;211;207;177;51;186;172;53;159;32;47;30;31;34;46;35;36;15;29;505;26;25;502;27;28;0;

Výpis ako sa postupne trasa zlepšovala pri rôznych seedoch

seed: 4738; distance: 3239
seed: 8900; distance: 3238
seed: 18471; distance: 3237
seed: 21635; distance: 3227
seed: 21641; distance: 3226
seed: 29332; distance: 3225
seed: 35026; distance: 3223
seed: 42798; distance: 3214
seed: 58071; distance: 3210

seed: 64840; distance: 3207
seed: 74100; distance: 3205
seed: 87003; distance: 3203
seed: 90857; distance: 3199
seed: 102976; distance: 3195
seed: 106673; distance: 3194
seed: 128599; distance: 3190
seed: 170872; distance: 3184
seed: 182703; distance: 3183
seed: 197185; distance: 3181
seed: 207305; distance: 3180
seed: 214852; distance: 3177
seed: 223414; distance: 3168
seed: 247417; distance: 3166
seed: 256381; distance: 3165
seed: 264282; distance: 3163
seed: 268118; distance: 3158
seed: 314492; distance: 3157
seed: 351205; distance: 3153
seed: 356964; distance: 3150
seed: 363245; distance: 3147
seed: 365620; distance: 3146
seed: 369704; distance: 3145
seed: 388557; distance: 3143
seed: 401629; distance: 3142
seed: 405369; distance: 3139
seed: 413071; distance: 3136
seed: 414592; distance: 3135
seed: 462628; distance: 3133
seed: 500290; distance: 3131
seed: 592219; distance: 3130
seed: 604725; distance: 3127
seed: 614114; distance: 3125
seed: 627808; distance: 3123
seed: 638626; distance: 3121
seed: 865369; distance: 3120

Najlepšia trasa

0;1;5;2;40;41;4;3;7;6;11;8;39;38;37;13;14;15;18;17;16;19;20;139;138;137;108;508;24;21;22;
23;109;110;111;112;116;115;114;113;136;143;140;141;142;126;48;49;171;50;10;9;12;33;4
4;154;188;215;45;504;189;42;43;153;156;155;187;157;158;163;164;165;166;167;162;184;1
85;169;168;161;160;54;144;61;60;70;71;72;278;317;316;315;314;281;280;279;277;276;76;7
3;102;104;103;101;319;318;320;74;77;75;80;78;82;79;81;83;91;92;99;117;85;84;90;106;89;
105;107;86;87;88;94;93;95;96;100;97;98;118;128;119;120;122;135;134;133;132;131;130;12
4;127;129;125;123;121;58;57;59;56;55;507;52;176;173;175;214;263;262;261;265;264;258;2
57;289;286;288;287;293;307;294;292;306;308;310;309;511;311;401;400;416;417;415;414;3
54;399;408;406;407;405;404;413;412;411;410;409;403;402;312;305;304;302;301;298;296;2
95;291;290;249;297;300;303;299;244;245;246;247;248;250;510;252;256;259;251;253;254;2

55;283;284;274;282;67;68;69;275;272;271;503;285;270;273;269;268;266;267;66;65;63;62;64;174;506;217;218;224;225;229;230;231;232;233;234;238;216;494;357;356;355;358;370;372;371;359;373;365;364;362;363;512;481;483;361;480;360;369;368;374;375;376;377;397;395;437;392;391;390;389;444;445;448;449;446;451;450;497;453;496;498;366;367;386;513;514;387;388;434;431;432;433;435;436;394;393;396;398;424;426;425;427;428;429;447;441;442;440;439;443;438;430;452;421;422;501;420;419;418;454;423;379;378;380;382;383;385;384;381;240;242;243;241;239;237;236;235;228;313;260;226;227;223;222;221;220;219;343;344;345;340;342;341;349;347;346;348;350;462;478;476;479;477;482;473;475;474;472;470;471;484;485;489;491;495;492;493;490;486;488;500;487;499;469;468;466;467;464;460;463;461;352;351;339;337;338;328;353;456;457;465;459;458;455;336;335;334;333;332;330;331;329;327;326;325;324;323;322;321;202;201;200;199;204;203;198;197;196;195;194;146;145;180;179;148;178;147;149;152;151;150;183;181;182;190;193;192;191;509;208;206;212;205;209;210;213;211;207;177;51;186;172;53;159;170;32;47;30;31;34;46;35;36;29;505;26;25;502;27;28;0;