ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

Semestrálna práca

Optimalizácia sietí

David Kučera
Akademický rok 2024/25

Obsah

Zadanie práce

Úlohou tejto práce je na vybranej testovacej matici (ZA) vzdialeností riešiť úlohu **obchodného cestujúceho** prideleným **heuristickým** algoritmom a následne toto riešenie zlepšiť pridelenou metaheuristikou.

Popis metódy

Na skonštruovanie východzej prípustnej trasy sa využila duálna heuristika **algoritmu zväčšovania o najvýhodnejší uzol** (2):
Algoritmus vychádza zo základnej neprípustnej trasy $i_1 - i_2 - i_3 - i_1$, ktorú v každom kroku zväčší vsunutím spracovávaného uzla medzi dva už zaradené uzly, ktoré nasledujú po sebe v súčasnej trase. Do súčasnej trasy bude z doposiaľ nezaradených uzlov zaradený ten, vsunutím ktorého sa dĺžka trasy zväčší najmenej, pričom spracovávaný uzol je zaradený medzi také dva po sebe idúce uzly v trase, aby sa dĺžka trasy zväčšila o čo najmenej. (t.j. každý doposiaľ nezaradený uzol je zaradený na také miesto v trase, aby sa trasa zväčšila najmenej. Zo všetkých takýchto trás vyberieme tú, ktorej dĺžka je najkratšia a spracovávaný uzol z tejto trasy sa stane zaradeným uzlom a spracovávané uzly z ostatných takto vzniknutých trás zostanú doposiaľ nezaradené). Základnú neprípustnú trasu $i_1 - i_2 - i_3 - i_1$ 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_2 a i_3 bude doposiaľ nezaradený uzol, ktorý je najviac vzdialený od uzla i_2.

Takto nájdenú východziu trasu obchodného cestujúceho **zlepšíme pomocou metaheuristiky Simulated Annealing** so spôsobom nájdenia okolia aktuálneho riešenia (B): **Inverzia podreťazcov dĺžky 5**

Popis riešenia

Riešenie bolo navrhnuté a implementované v programovacom **jazyku C#** v prostredí **.NET 8.0** ako konzolová aplikácia. Riešenie je možné spustiť v rôznych vývojových prostrediach ako **Microsoft Visual Studio 2022**, alebo JetBrains Rider pomocou súboru *OSSP.sln*. Odovzdaný priečinok obsahuje aj priečinok *run* (*prípadne run_D*), kde sa nachádza spustiteľný súbor vypracovania *.exe*. Zdrojové kódy sa nachádzajú v priečinku OSSP a spravidla končia príponou *.cs*.

Popis premenných, štruktúr a tried použitých v programe

Výsledné riešenie obsahuje 4 triedy – **Program**, **MatrixLoader**, **SalesmanHeuristic** a **SimulatedAnnealing**.

MatrixLoader

Táto trieda sa používa výhradne len na načítanie vstupných dát matice vzdialeností zo súboru a uloženie ich do dvojrozmerného poľa.

Load(string[] lines)

Vstupným parametrom je pole riadkov vstupného súboru. Následne sa tieto riadky prechádzajú a prvky sa ukladajú do dvojrozmerného poľa int s názvom matrix. Po prejdení všetkých riadkov metóda vráti matrix.

SalesmanHeuristic

Trieda, v ktorej je implementovaná duálna heuristika popísaná vyššie v „Popis metódy“. Obsahuje aj pomocné metódy na nájdenie najviac vzdialeného vrcholu od daného vrcholu a na vypočítanie ceny danej trasy.

Atribúty:

- *Dij* – načítaná matica vzdialeností
- *D* – cena aktuálnej trasy
- *Nezaradene* – zoznam aktuálne nezaradených vrcholov
- *Path* – aktuálna trasa

Solve(int[][] dij)

Parametrom je načítaná matica vzdialeností. Z nej sa skonštruuje zoznam nezaradených vrcholov. Východzia trasa sa skonštruuje podľa zadania popísaného vyššie v „Popis metódy“. Následne sa začína algoritmus postupného zväčšovania o najvýhodnejší vrchol. Ten sa dá popísať nasledovne:

1. Kým je nejaký vrchol nezaradený choď na krok 2, inak koniec.
2. Nájdi najvýhodnejší vrchol z nezaradených a jeho pozíciu nasledovne:
    a. pre každý nezaradený vrchol:
        i. pre každú možnú pozíciu medzi dvojicami vrcholov v trase

1. vypočítaj nárast vzdialenosti ktorý vznikne vložením vrcholu na danú pozíciu
2. ak je tento nárast menší ako aktuálny minimálny nárast aktualizuj
    a. najlepší vrchol na aktuálny vrchol
    b. najlepšiu pozíciu na aktuálnu pozíciu
3. Vlož vrchol do trasy na danú pozíciu.
4. Odstráň vrchol zo zoznamu nezaradených vrcholov, vráť sa na krok 1.

GetPathCost(List<int>? path = null)

Metóda sa používa aj v iných triedach na zistenie dĺžky trasy, preto je parameter nepovinný. Ak sa nezadá, použije sa pre aktuálnu Path v tejto triede. Ak sa parameter zadá, vypočíta sa dĺžka pre zadaný zoznam vrcholov na danej matici vzdialeností v metóde Solve.
Prechádza sa celý zoznam danej trasy a len sa postupne pripočítavajú vzdialenosti medzi jednotlivými vrcholmi za sebou.

GetMostDistantNodeFrom(int index)

Metóda vráti index najviac vzdialeného vrcholu od daného vrcholu daným indexom. Teda najskôr sa nájde daný riadok pomocou indexu, v ňom sa nájde maximálna hodnota – najvzdialenejší vrchol a následne sa vráti jeho index v poli.

SimulatedAnnealing

V tejto triede je implementovaná metaheuristika Simulated Annealing. Jej algoritmus je implementovaná podľa knihy „Optimalizace na dopravních sítích", str.95-96.

Atribúty:

- *_rand* – generátor náhodných hodnôt

Konštanty:

- *T_MAX* – počiatočná teplota
- *U* – maximálny počet preskúmaných prechodov od prechodu k súčasnému riešeniu
- *Q* – maximálny počet preskúmaných prechodov od poslednej zmeny teploty
- *DLZKA_INV_RETAZCA* – dĺžka invertovaného reťazca v trase podľa zadania (5)
- *BETA* – hodnota používaná pre výpočet novej teploty v algoritme

Solve(List<int> x_0)

Tu je implementovaný algoritmus tejto metaheuristiky. Je implementovaná podľa knihy spomenutej vyššie. Premenné používané v tejto metóde:

- *xStar* – doposiaľ najlepšie nájdené riešenie
- *t* – teplota
- *v* - počet aktualizácií doposiaľ najlepšieho nájdeného riešenia od posledného zahrievania

- *r* – počet preskúmaných prechodov od prechodu k súčasnému riešeniu
- *w* – celkový počet preskúmaných prechodov od poslednej zmeny teploty

GetOkolie(List<int> xStar)

Metóda na získanie okolia trasy podľa zadaného algoritmu – inverzia reťazcov dĺžky 5.

Program

Trieda obsahujúca metódu Main pre spustenie programu. Postupne volá jednotlivé metódy tried pre vyriešenie problému spolu s informačnými výpismi a výpisom konečného riešenia.

Riešenie nájdené duálnou heuristikou a metaheuristikou

Dĺžka trasy po riešení **duálnou heuristikou**: 1962
Trasa po riešení duálnou heuristikou:
Path: -0-7-4-6-5-3-2-1-21-20-18-19-22-37-23-24-26-25-27-28-128-127-125-124-126-138-135-310-129-130-131-132-133-134-169-168-170-172-171-228-173-174-186-185-183-184-177-308-303-302-179-178-180-182-191-226-225-223-222-221-198-312-224-220-219-240-233-210-232-234-241-239-235-238-236-242-251-258-249-246-244-256-259-257-245-307-306-305-247-253-254-237-250-255-252-248-243-304-313-190-202-204-209-208-203-201-200-199-197-196-194-181-193-192-189-195-205-211-212-188-187-213-44-56-67-66-52-55-54-53-57-58-59-311-51-50-49-47-48-39-40-42-41-43-45-46-215-216-207-206-230-231-77-68-62-64-75-73-74-72-71-70-76-69-65-63-61-217-60-218-214-229-227-175-176-166-165-149-145-143-144-146-162-157-156-161-163-275-276-284-285-164-167-160-278-279-280-301-263-295-294-300-298-299-297-296-260-261-290-289-292-287-288-291-286-271-269-262-314-268-267-265-264-266-277-272-273-270-274-283-281-282-154-153-159-158-155-152-151-148-147-150-140-139-142-141-136-137-98-11-123-35-34-122-33-38-29-30-36-31-9-32-10-12-17-99-13-81-80-82-83-84-309-86-85-94-96-95-104-109-114-115-116-293-118-117-120-113-110-112-111-107-106-108-105-100-101-102-103-97-93-90-91-87-92-89-88-119-79-14-121-78-15-16-8-0-

Dĺžka trasy po zlepšení riešenia **metaheuristikou SA**: 1962
Trasa po riešení metaheuristikou SA:
Path after SA: -0-7-4-6-5-3-2-1-21-20-18-19-22-37-23-24-26-25-27-28-128-127-125-124-126-138-135-310-129-130-131-132-133-134-169-168-170-172-171-228-173-174-186-185-183-184-177-308-303-302-179-178-180-182-191-226-225-223-222-221-198-312-224-220-219-240-233-210-232-234-241-239-235-238-236-242-251-258-249-246-244-256-259-257-245-307-306-305-247-253-254-237-250-255-252-248-243-304-313-190-202-204-209-208-203-201-200-199-197-196-194-181-193-192-189-195-205-211-212-188-187-213-44-56-67-66-52-55-54-53-57-58-59-311-51-50-49-47-48-39-40-42-41-43-45-46-215-216-207-206-230-231-77-68-62-64-75-73-74-72-71-70-76-69-65-63-61-217-60-218-214-229-227-175-176-166-165-149-145-143-144-146-162-157-156-161-163-275-276-284-285-164-167-160-278-279-280-301-263-295-294-300-298-299-297-296-260-261-290-289-292-287-288-291-286-271-269-262-314-268-267-265-264-266-277-272-273-270-274-283-281-282-154-153-159-158-155-152-151-148-147-150-140-139-142-141-136-137-98-11-123-35-34-122-33-38-29-30-36-31-9-32-10-12-17-99-13-81-80-82-83-84-309-86-85-94-96-95-104-109-114-115-116-293-118-117-120-113-110-112-111-107-106-108-105-100-101-102-103-97-93-90-91-87-92-89-88-119-79-14-121-78-15-16-8-0-

Pozn. jednotlivé čísla v trase reprezentujú poradie jednotlivých uzlov z dátovej matice, teda 0 je prvý uzol, 1 je druhý uzol, ap.

Záver

Po implementovaní a vyriešení zadaného problému pomocou duálnej heuristiky na našej testovacej matici sme vytvorili východzie riešenie zadaného problému. Žiadne ďalšie zlepšenie východzej trasy pomocou metaheuristiky Simulated Annealing sme nezískali.