

# Vnorená inteligencia

Ing. Marián Podmajerský

## Umelá inteligencia

Umelá inteligencia nie je len o rozprávajúcich počítačoch a robotoch hľadajúcich lásku a smiech. V skutočnosti umelá inteligencia je o veľa dôležitejšia v tej najjednoduchšej forme: účelová schopnosť rozhodovania vo vnorenem systéme.

Navzdory štyrom dekádam výskumu a vývoja, v poslednom desaťročí nič nenaznačovalo aplikácii heuristiky pri programovaní počítačových algoritmov. Dnes má heuristika veľký dopad na výpočty. Napríklad, kontrola pravopisu bola považovaná za úžasne inteligentnú aplikáciu keď bola prvý raz uvedená. Dnes patrí k štandardu.

V skutočnosti, do praxe nasaditeľné programovanie umelej inteligencie je ďaleko pred nami. Samotné programovanie sa v podstate zjednoduší na predpoklad "inteligencie" aplikovaním efektívnych vyhľadávacích metód na sústavu dát. Inteligentné metódy sa môžu miešať s tradičnými precedúrami. Vedomosti získané skúšaním a narábaním s reálnymi objektami a schopnosť ich použiť v správny čas, umožňuje programátorovi dodať takejto aplikácii silu, funkčnosť, účinnosť, efekt a sofistikovanosť.

## Reprezentácia dát

Aby sme mohli navrhovať vyhľadávanie na reálnych dátach, najprv si musíme zostrojiť ich vhodnú reprezentáciu, či už grafickú, alebo matematickú. Často používanou reprezentáciou dát je ich grafické znázornenie grafmi. Ukazuje sa, že grafy sú efektívne pri riešení, v ktorých je požadovaný istý stupeň inteligencie. Grafy a teória grafov poskytuje vizualizačné pomôcky, pravidlá o tvare, forme, a techniky účinné pri rozhodovaniach zameraných na špecifický problém.

Grafy majú vlastnú jedinečnú terminológiu: graf pozostáva z uzlov a úsekov. Uzol je vo všeobecnosti zobrazený ako kruh, úsek je spojnica medzi uzlami. Uzly sú v programoch zväčša implementované ako štruktúry dát s úsekmi a smerníkmi, alebo ako indexy matíc.

K metódam skúmajúcim grafy patria napríklad jedno-úrovňové vyhľadávanie ("depth-first"), a plošné ("breadth") vyhľadávanie. Obe, jedno-úrovňové i plošné vyhľadávacie metódy patria medzi slepé vyhľadávacie metódy. Jedno-úrovňová metóda prechádza uzly a úseky medzi nimi od východiskového uzla po cieľový uzol. Na druhej strane, metóda plošného vyhľadávania skúša všetky uzly vo vzdialenosti jedného úseku od počiatku, potom vo vzdialenosti dvoch úsekov od počiatku atď.

Tieto a im podobné metódy patria do skupiny slepých metód, pretože počas prehľadávacieho procesu nezískavame žiadnu cennú informáciu o polohe alebo smere ku správne riešeniu. Takéto prehľadávacie metódy slepo prechádzajú nablížšie uzly v domnienke, že objavia riešenie. I to, že vieme kedy použiť jeden zo slepých vyhľadávacích algoritmov je veľká devíza ak je zakomponovaná do

vnorenej inteligencie k ostávajúcim metódam.

## Navigácia vozidiel

Povedz me, že navrhujeme systém, ktorý naviguje sieťou ciest. Môže to byť napríklad, automatický zber odpadu, kamera v pohybe, alebo automatické presmerovávanie dopravy. Na Obr. 1 je mapa časti San Francisca, ktorá poslúži na ilustráciu.

Prvým krokom pri návrhu takéhoto riešenia je vytvorenie grafickej reprezentácie dát a zakreslenie uzlov a hrán do grafu. Križovatky ciest je vhodné zvoliť ako uzly. Ich pridaním, získavame čiastočne kompletnú grafickú reprezentáciu dát. Výsledok je bezúčelová, statická reprezentácia mapy.



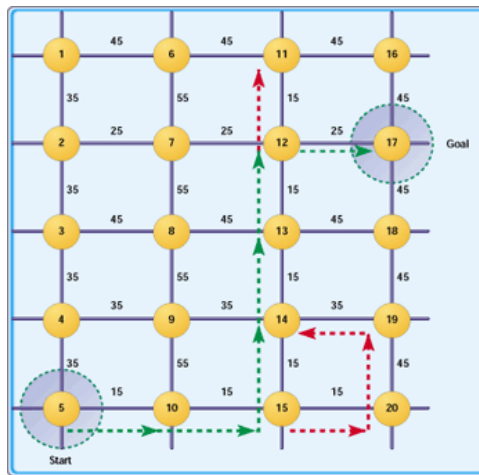
Obr. 1: Ciet' ciest

V ďalšom kroku, hľadáme prídavnú informáciu, na základe ktorej by systém robil inteligentné rozhodnutia. Ak účelom systému bude zvoliť najlepšiu cestu vozidla od jednej križovatky do druhej, je prirodzené charakterizovať alebo ohodnotiť úseky ciest medzi križovatkami. Spojnica jednotlivých úsekov medzi križovatkami môže byť ohodnotená rôzne. V najjednoduchšom prípade predpokladáme, že žiadna z ciest nie je jednosmerná, ale všetky sú obojsmerné. Taktiež rýchlostné obmedzenia a počet pruhov v každom smere je rovnaký. (V prípade, že to nie je tak, je relatívne jednoduché rozšíriť graf o ďalšie doplnkové premenné a ohodnotenia.)

Ohodnotením úsekov medzi jednotlivými križovatkami, umožníme systému navigovať sieťou ciest, teda nájsť najlepšiu cestu vyhovujúca počiatočným podmienkam. V ilustračnom príklade (navigácia ulicami San Francisca), budeme predpokladať, že úseky reprezentujú priemernú hodinovú hustotu premávky. Pre našu potrebu budú zvolené náhodne. Ohodnotením úseku môže byť ľubovoľný opis dynamiky ako napríklad priemerná, maximálna, alebo minimálna rýchlosť, deň v týždni, počasie, havárie, blokády na ceste, sú taktiež možnosti ako ohodnotiť úseky. V tomto prípade, žiadne ďalšie nevlývajú na ďalšiu analýzu.

Z Obr. 2 možno vidieť, že úseky sú navrhnuté tak, aby reprezentovali priemernú dennú premávku na cestách za hodinu (tieto údaje nie sú získané meraním aktuálnych dát, ale primerane zvolené pre

ilustračné účely). Ak chceme aby naše vozidlo bolo navigované napríklad z rohu ulíc Scott-ovej a Jackson-ovej (križovatka 5) ku rohu ulíc Filmore-ovej a Vallejo-ovej (križovatka 17), zohľadnením kritéria o najmenšej hustote premávky na jednotlivých úsekoch, naša navrhnutá navigácia by mal byť schopná vyhľadať cestu ulicami s najmenšou hustotou premávky.



Obr. 2: Navigácia mapou

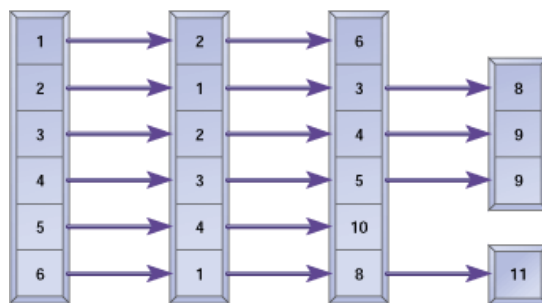
Je jednoduché si načrtnúť grafickú reprezentáciu ručne. Avšak, aby navigačný proces mohol byť automatizovaný je potrebné zapísať ju rečou počítača. Dva najbežnejšie spôsoby zápisu grafov sú pomocou matíc najkratších vzdialeností alebo zoznamov najkratších vzdialeností.

Matica najkratších vzdialeností je mnohorozmerové pole s hodnotami uloženými v matici, ktoré reprezentujú ohodnotenie úsekov z jedného uzla ku ďalšiemu. Obr. 3 ilustruje čiastočnú maticu najkratších vzdialeností obsahujúcu informácie o úseku medzi uzlami 1 až 6 pre náš príklad. Ohodnotenie úsekov medzi uzlami 1 až 6 zapisujeme do matice od pravého horného rohu. Kompletnú maticu najkratších vzdialeností pre 20 uzlov siete cest z Obr. 2 by pozostávalo zo 400 elementov, ale len 31 z nich by obsahovalo plnohodnotnú informáciu.

	1	2	3	4	5	6
1	0	35	0	0	0	45
2	35	0	35	0	0	0
3	0	35	0	35	0	0
4	0	0	35	0	35	0
5	0	0	0	35	0	0
6	45	0	0	0	0	0

Obr. 3: Matica najkratších vzdialeností

Matica najkratších vzdialeností je zvyčajne implementovaná ako prepojený zoznam. Ob. 4 znázorňuje takýto zoznam pre uzly 1 až 6. Ohodnotenie úsekov nie je ďalej reprezentované graficky, ale dátovými štruktúrami.



Obr. 4: Zoznam najkratších vzdialeností

Podľa akých kritérií volíme medzi maticou alebo zoznamom najkratších vzdialeností na konverziu grafu na dátovú štruktúru, je zhrnuté v nasledujúcich bodoch:

1. Ak je graf bohatý na informácie alebo rozmermi malý, lepšia bude popísaný maticou najkratších vzdialeností. Výhoda takejto matice je, že ju možno adresovať priamo, bez ďalšej manipulácie so smerníkmi a komplikovaným listovaním v zoznamoch.
2. Ak je graf chudobný na informácie alebo rozmermi rozsiahly, zoznam najkratších vzdialeností môže zredukovať množstvo spotrebovanej pamäte.
3. Ak sa do budúcnosti počíta s pridávaním nových uzlov a úsekov, alebo ich odstraňovaním za behu, je výhodné použiť zoznam najkratších vzdialeností. Je zrejmé, že pre plnohodnotnú manipuláciu s pamäťou, potrebujeme si ju vedieť dynamicky vyhradiť.

## Heuristika

Ak budeme prehľadávať graf naslepo, tak že budeme prechádzať úseky s najnižším ohodnotením, môže sa stať, že pôjdeme nesprávnou cestou a nikdy neprídeme do cieľa. O veľa inteligentnejšie riešenie môže byť heuristické vyhľadávanie, čo je v podstate sada pravidiel, ktoré platia vo všeobecnosti. Ukážeme si jednoduchý ľudský prístup na príklade: „Je apríl a oblačno vonku, tak si vezmeme dáždnik“. Len preto, že je apríl a vonku oblačno neznamená, že bude pršať, ale pravdepodobnosť je vyššia ako za normálnych podmienok.

Heuristika je dôležitá časť účinného a efektívneho vyhľadávania. Pomocou nej môžeme nájsť riešenie i napriek tomu, že štartovací bod je mimo predpokladaný. Pre navigačné problémy, jedna z heuristík môže byť: „Keď uzol existuje na dvoch alebo viacerom rovnako ohodnotených úsekoch, vykonaj breadth-first vyhľadávanie. Sleduj cestu s najmenším celkovým ohodnotením.“ Ak zoberieme v úvahu uzol číslo 15 v našom príklade, zistíme, že dva úseky s ohodnotením 15 vychádzajú z tohto uzla. Z ďalšej úrovne (z nasledujúcich uzlov) vychádzajú úseky ohodnotené 15 a 45. Ak by sme použili kritérium najkratšieho úseku, tak vyhľadávanie by pokračovalo uzlom 14 a uzol 20 by bol zabudnutý.

Niektoré heuristiky sa môžu zdať na prvý pohľad samorejmé, ale i napriek tomu je treba bližší pohľad na povahu riešenia problému. Veľmi základná heuristika pre navigáciu môže byť stanovená: „Sleduj cestu s najmenším ohodnotením“. Je to jednoduché a zrejmé, avšak mylné.

Metódy zakladajúce na metóde nasledovania najmenej oceneného úseku sa nazývajú žravé

metódy. Žravé metódy sú založené na aktuálnom ohodnotení. Bez predvídania, žravá metóda nasleduje najmenej ohodnotenú cestu po úsekoch. Tento prístup negarantuje výsledok a bežne nájde menej optimálne riešenie. Keď sa rozhoduje medzi navigáciou jednotlivými úsekmi, aj napriek tomu, že sa neskôr zistí, že to bolo nesprávne, žraví algoritmus by odpovedal: „Vyzeral to byť skvelý nápad.“

Čo vyzerať byť zrejme pre človeka nie je ani zďaleka rovnako zrejme pre počítač. Naša heuristika, „keď z uzla vychádzajú dva (alebo viac) rovnako ohodnotenú úseky, vykonaj jedno-úrovňové vyhľadávanie. Nasleduj úseky s najnižším celkovým ohodnotením“, nie je zlá, ale má trhliny. Čo ak takéto riešenie naviguje mimo požadovaný cieľ? V takom prípade si môžeme zvoliť drahšiu cestu. Ukazuje sa, že bez toho aby sme vedeli niečo o smere z aktuálnej pozície ku cieľovej pozícii, nemáme dostatočnú kvalitu navigácie. Zakomponovaním heuristiky takéhoto druhu, je dobrý spôsob ako vylepšiť heuristiku a zefektívniť riešenie problematiky.

Jeden spôsob ako vyriešiť navigačný problém je dynamicky počítať vzdialenosť a pomer medzi aktuálnym a cieľovým uzlom. Potrebujeme na to dáta o zemepisnej šírke a zemepisnej dĺžke, pre každý uzol a pre každý pohyb medzi nimi. Toto je pravdepodobne najnižšie optimálne riešenie, ale nároky na prísun dát, meranie dár a ich kvalitu, sú najvyššie. Tým, že vytvoríme skupinu pravidiel pre rozdiely medzi dvomi zemepisnými šírkami alebo dĺžkami, znížime počet potrebných informácií na minimum.

Najprv určíme vzťah medzi vzdialenosťami, zemepisnými šírkami a dĺžkami. Obr. 5 ukazuje smery odvodené z rozdielov medzi zemepisnou šírkou a dĺžkou.

Keď cestujeme severne, zemepisná šírka sa zvyšuje; ak na západ, dĺžka rastie; a podobne. Týmto jednoduchými vzťahmi, môžeme vylúčiť množstvo operácií v uzloch a medzi nimi. Aplikovaním predchádzajúceho na mapu, počítačová zemepisná šírka a zemepisná dĺžka križovatky Scott-ovej a Jackson-ovej bude N37 47.514 zemepisnej šírky a W122 26.356 zemepisnej dĺžky. Cieľová križovatka na rohu Fillmore-ovej a Vallejo je N37 47.725 zemepisnej šírky a W122 26.002 zemepisnej dĺžky. Aplikovaním kompasového pravidla podľa Obr. 5, je zrejme, že cieľová križovatka je severnejšie od počítačovej križovatky.

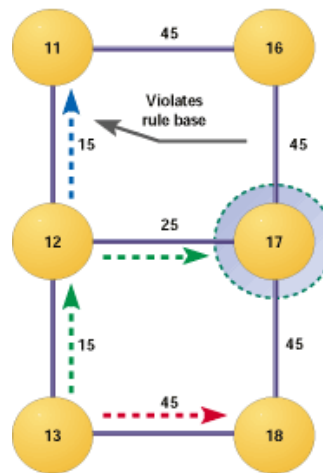
Šesť individuálnych pravidiel môžeme zostrojiť na základe vzťahu medzi smermi:

- Pravidlo 1: Ak Zemepisná\_šírka(cieľ) > Zemepisná\_šírka(aktuálna) potom Cieľ je severne.
- Pravidlo 2: Ak Zemepisná\_šírka(cieľ) < Zemepisná\_šírka(aktuálna) potom Cieľ je južne.
- Pravidlo 3: Ak Zemepisná\_šírka(cieľ) = Zemepisná\_šírka(aktuálna) potom Sme v cieľi.
- Pravidlo 4: Ak Zemepisná\_dĺžka(cieľ) > Zemepisná\_dĺžka(aktuálna) potom Cieľ je západne.
- Pravidlo 5: Ak Zemepisná\_dĺžka(cieľ) < Zemepisná\_dĺžka(aktuálna) potom Cieľ je východne.
- Pravidlo 6: Ak Zemepisná\_dĺžka(cieľ) = Zemepisná\_dĺžka(aktuálna) potom Sme v cieľi.

Kombináciu týchto základných pravidiel môžeme rozvinúť na komplexnejšie smery ako Severovýchod alebo Juho-západ. Logickou kombináciou základných pravidiel operátorom AND získavame ďalšiu skupinu komplexnejších pravidiel:

- Pravidlo 1 & Pravidlo 4 -> cieľ je Severozápadne
- Pravidlo 1 & Pravidlo 5 -> cieľ je Severovýchodne
- Pravidlo 1 & Pravidlo 6 -> cieľ je Severne
- Pravidlo 2 & Pravidlo 4 -> cieľ je Juhozápadne
- Pravidlo 2 & Pravidlo 5 -> cieľ je Juhovýchodne
- Pravidlo 2 & Pravidlo 6 -> cieľ je Južne
- Pravidlo 3 & Pravidlo 4 -> cieľ je Západne
- Pravidlo 3 & Pravidlo 5 -> cieľ je Východne
- Pravidlo 3 & Pravidlo 6 -> sme v cieľi

Výsledkom začlenenenia základných a rozšírených pravidiel je úspešná vyhľadávacia metóda. Ak je cieľom Severozápad od aktuálneho uzla, navigácia Severne a Východne je správna, ale Južne a Západne nie je správna. Keď vyhľadávanie dorazí do uzla číslo 12, logická cesta vedie úsekom z uzla číslo 12 do uzla číslo 11 s ohodnotením 15. Smer je Sever, čo vyzerá byť správne a ohodnotenie úseku je minimálne. Toto je totálne nesprávny smer, pretože navigácia zatočí úplne mimo od cieľa. Sme v situácii keď sada základných pravidiel mala ohraničiť navigáciu. Uzol číslo 12 je rovnobežný s uzlom číslo 17, takže pravidlo 3 je splnené. Zemepisná dĺžka sa znižuje, takže pravidlo 5 je splnené. Ak pravidlá 3 a 5 sú splnené, presúvame sa na Východ. Sada základných pravidiel vykázala lepší výsledok ako navigovanie pomocou slepého vyhľadávania. Vid' Obr. 6.



**Obr. 6: Porušenie základného pravidla**

## Záverom

Ukázali sme, že grafická reprezentácia dát a slepé vyhľadávacie metódy samostatne nie sú dostatočne účinné pri riešení problémov. Použitím kombinácie týchto techník s heuristikou a sadou pravidiel, špecifických pre problematiku, ako bolo ilustrované na príklade, získavame prístup, ktorý úspešne začleňuje umelú inteligenciu. Podobné techniky môžu byť uplatnené na širšie spektrum problémov. I napriek tomu, že náš príklad je zameraný na statické dáta, techniky ukázané v predchádzajúcich riadkoch sú pravdepodobne najúčinnnejšie keď sa úseky a ohodnotenie úsekov mení a žiadne presné pravidlá nie sú možné.

Samozrejme, vnorené systémy trpia niektorými obmedzeniami. Rekurzia nie je zvyčajne povolená pri programovaní vnorených systémov, avšak je to bežný postup v graf-prehľadávacích technikách. I keď množstvo potrebnej dynamickej pamäte a manipulácia s ňou je vo všeobecnosti najslabším článkom zoznamov s najkratšími vzdialenosťami, ale vzhľadom na to, že v praxi je potrebné a pridávať a odoberať uzly za behu, je to najlepšia možnosť. Vzhľadom na predchádzajúce úvahy, môžeme formulovať odporúčania:

- čo najviac operácií nespracovávať za behu, zväčša len niektoré operácie sa obnovujú
- vyhnúť sa rekurzii, ľubovoľná rekurzia sa dá prepísať iteračne
- udržať vyhradenie pamäte na minime, ak veľkosť zoznamu ostáva približne konštantná, pole

je lepšia voľba

- neuvažovať nad zložitou koncepciou inteligencie, ale zamerať sa na tie najzákladnejšie a poučiť sa od jednoduchých foriem života ako sa vyrovnávajú s neočakávanými situáciami a prekážkami

Pri riešení problému vyžadujúceho inteligenciu neexistujú obmedzenia na počet kombinácií metód, či už slepých, žravých alebo inteligentných.