Vplyv veľkosti a štruktúry dát na algoritmickú zložitosť v triedach P a NP

The impact of data size and structure on algorithmic complexity in P and NP classes

Peter Schmidt[1]

Abstrakt
Výpočtová zložitosť a klasifikácia problémov do kategórií P a NP predstavujú kritické aspekty v oblasti algoritmickej zložitosti. Tento článok sa zameriava na interakciu medzi veľkosťou a štruktúrovanosťou dátových setov a ich vplyv na zaradenie problémov do týchto kategórií. Zatiaľ čo problémy v kategórii P sú efektívne riešiteľné, problémy v kategórii NP sú charakterizované rýchlym overením ich riešení. V kontexte Big Data sa objavuje nová úroveň komplexity, ktorá komplikuje klasifikáciu problémov. Článok tiež rozširuje diskusiu na Nos-Nob dátové sety, ktoré sú príliš veľké na bežné počítače, ale príliš malé na distribuované systémy, a často vyžadujú špeciálny prístup. Na základe analýzy v rámci štruktúrovaných, semi-štruktúrovaných a neštruktúrovaných dát v kontexte small, big a nos-nob dát, článok ukazuje, že je možné odhadnúť, do akej kategórie dát úloha spadá, a tým pádom aj najvhodnejšiu technológiu spracovania, na základe jej zaradenia do kategórie P alebo NP.

Kľúčové slová
výpočtová zložitosť, algoritmická zložitosť, small data, big data, NoS-NoB data, kategorizácia (P a NP)

Abstract
Computational complexity and the classification of problems into the categories P and NP represent critical aspects in the field of algorithmic complexity. This article focuses on the interaction between the size and structurality of data sets and their impact on the categorization of problems into these categories. While problems in the P category can be efficiently solved, problems in the NP category are characterized by quick verification of their solutions. In the context of Big Data, a new level of complexity emerges, complicating the classification of problems. The article also extends the discussion to Nos-Nob data sets, which are too large for regular computers but too small for distributed systems, and often require a specialized approach. Based on an analysis within structured, semi-structured, and unstructured data in the context of small, big, and nos-nob data, the article shows that it is possible to estimate into which data category a task falls, and therefore the most suitable processing technology, based on its categorization into P or NP.

Key words
computational complexity, algorithmic complexity, small data, big data, Nos-Nob data, categorization (P and NP)

[1] Ekonomická univerzita, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemská cesta 1, 852 35 Bratislava, peter.schmidt@euba.sk

JEL classification
C8

1 Úvod

Klasifikácia výpočtových problémov do tried P a NP predstavuje základný kameň v štúdiu algoritmickej zložitosti. Zatiaľ čo problémy triedy P môžu byť efektívne vyriešené, problémy triedy NP sú tie, ktorých riešenia môžu byť rýchlo overené. Tu vzniká jedna dôležitá otázka: ovplyvňuje zložitosť týchto algoritmov viac veľkosť dát alebo ich štruktúra?

P a NP sú triedy problémov v teórii výpočtov. "P" označuje triedu problémov, ktoré je možné efektívne riešiť, teda existuje algoritmus, ktorý ich dokáže riešiť v polynomiálnom čase. "NP" označuje triedu problémov, pre ktoré, ak existuje riešenie, je možné ho efektívne overiť, teda overenie riešenia je možné vykonať v polynomiálnom čase. Otázka "P vs. NP" je otázka, či existujú problémy, ktoré je možné efektívne overiť (sú v NP), ale nie efektívne vyriešiť (nie sú v P). Táto otázka zatiaľ ostáva nezodpovedaná. (Cormen et al., 2022)

Matematické vyjadrenie P problému je založené na časovej zložitosti algoritmu, ktorý rieši daný problém. Problém patrí do triedy P, ak existuje algoritmus, ktorý ho môže vyriešiť v polynomiálnom čase. To znamená, že najhoršia možná časová zložitosť algoritmu rastie polynomiálne vzhľadom na veľkosť vstupu. Matematicky by to mohlo byť vyjadrené ako:

$$T(n) = O(n^k)$$

kde:

- $T(n)$ je časová zložitosť algoritmu,
- n je veľkosť vstupu,
- k je konštanta,
- O označuje notáciu, ktorá opisuje horný limit rastu funkcie.

Takže, ak existuje algoritmus, ktorý môže riešiť daný problém v čase, ktorý je zhora ohraničený polynómom veľkosti vstupu, potom je to P problém.

Matematicky je NP problém charakterizovaný tým, že riešenie problému je možné overiť (nie nutne nájsť) v polynomiálnom čase. To znamená, ak je k dispozícii kandidátske riešenie, je možné v polynomiálnom čase zistiť, či je to riešenie správne alebo nie. To môže byť matematicky vyjadrené ako:

$$V(x, y) = O(n^k)$$

kde:

$V(x, y)$ je verifikačný algoritmus, ktorý overuje, či y je platné riešenie problému so vstupom x.

- x je vstupný problém.
- y je kandidátske riešenie problému.
- n je veľkosť vstupu.
- k je konštanta.
- O označuje notáciu, ktorá opisuje horný limit rastu funkcie.

Problémy v triede NP sú tie, kde je overenie riešenia rýchle (v polynomiálnom čase), ale nie je známe, či existuje algoritmus, ktorý by mohol nájsť riešenie v polynomiálnom čase (čo by znamenalo, že P=NP).

P a NP algoritmy

P algoritmy sú tie, ktoré riešia problémy v triede P. Tieto problémy môžu byť riešené a vyriešené v polynomiálnom čase. Bežné príklady P algoritmov zahŕňajú algoritmy na triedenie dát, vyhľadávanie a iné úlohy, ktoré sa dajú efektívne vyriešiť (Arora & Barak, 2016).

NP algoritmy sú asociované s problémami v triede NP. Sú to problémy, pri ktorých, aj keď môže byť náročné nájsť riešenie, akonáhle je riešenie nájdené, je relatívne jednoduché a je možné rýchle overiť jeho správnosť. Klasickým príkladom NP problému je problém obchodného cestujúceho, kde je náročné nájsť najkratšiu možnú cestu cez daný zoznam miest, ale ak máme danú cestu, je jednoduché overiť jej dĺžku.

Otázka, či P=NP alebo P≠NP, znamená, či existujú nejaké problémy, ktoré môžeme rýchlo overiť, ale nemôžeme ich rýchlo vyriešiť. Táto otázka je jednou z najväčších nevyriešených otázok v informatike.

2 Vplyv veľkosti a štruktúry dát na algoritmickú zložitosť P a NP v kontexte Small Data

Veľkosť dát vs. štruktúrovanosť

Obe vlastnosti, veľkosť aj štruktúra dát, majú kľúčový vplyv na efektivitu a výkon algoritmov. Veľkosť dát môže byť obmedzujúcim faktorom, najmä pre algoritmy, ktoré nie sú dobre škálovateľné. Na druhej strane, štruktúra dát môže rovnako ovplyvniť efektivitu algoritmov. Napríklad, algoritmy na triedenie môžu mať rozličný výkon v závislosti od organizácie vstupných dát (Peng & Matsui, 2016).

P a NP pre štruktúrované small data

P úloha:

Jednoduchým príkladom P úlohy pre malé štruktúrované dáta by mohlo byť lineárne vyhľadávanie. Predstavte si, že máte malý zoznam čísel (napr. [1, 2, 3, 4, 5]) a chcete zistiť, či konkrétne číslo (napr. 4) je v tomto zozname. Algoritmus prechádza zoznamom a porovnáva každý prvok s hľadaným číslom. Táto úloha je v triede P, pretože čas potrebný na jej vyriešenie rastie lineárne s veľkosťou vstupu.

NP úloha:

Príkladom NP úlohy pre malé štruktúrované dáta by mohla byť problém obchodného cestujúceho (TSP - Traveling Salesman Problem). Predstavte si, že máte malý počet miest (5 miest) a potrebujete nájsť najkratšiu možnú cestu, ktorá prejde všetkými mestami práve raz a vráti sa späť do východiskového mesta. Pre malý počet miest môže byť riešenie relatívne rýchlo identifikované hrubou silou, ale zložitosť problému rastie exponenciálne s pridaním ďalších miest. TSP je NP-ťažký problém, pretože rýchlo overíme, či dané riešenie je správne, ale nájsť toto riešenie môže byť časovo náročné.

P a NP pre semi-štruktúrované small data

P úloha:

Pre semi-štruktúrované dáta môžeme uviesť príklad vyhľadávania kľúčových slov v súbore JSON. Predpokladajme, že máte malý JSON súbor, ktorý obsahuje zoznam zamestnancov a ich atribútov (napr. meno, vek, oddelenie). Úloha by mohla byť vyhľadať všetkých zamestnancov, ktorí pracujú v určitom oddelení. Algoritmus by prechádzal dátami a zhromažďoval záznamy, ktoré spĺňajú kritériá vyhľadávania. Táto úloha by sa dala vyriešiť v polynomiálnom čase a je to príklad P úlohy.

NP úloha:

Zložitejší príklad so semi-štruktúrované dátami by mohol byť problém klasterizácie. Napríklad, malý súbor JSON s dátami o zákazníckych hodnoteniach rôznych produktov a chceme identifikovať skupiny zákazníkov s podobnými preferenciami. Zložitosť tohto problému môže rásť s počtom zákazníkov a hodnotení, a aj keď sú dáta malé, identifikácia optimálnych klastrov môže byť NP problém, závislý od konkrétneho algoritmu a metriky podobnosti, ktoré sú použité. Riešenie môže byť rýchlo overené (t. j. overenie, či zákazníci v rovnakom klastri majú podobné hodnotenia), ale nájdenie týchto klastrov môže byť náročnejšie.

P a NP pre neštruktúrované small data

P úloha:

Pre neštruktúrované dáta, ako je text, môže byť príkladom úlohy P výskyt konkrétneho slova alebo frázy v texte. Predpokladajme malý textový dokument a úlohou je zistiť, či obsahuje konkrétne slovo alebo frázu. Tento proces by zahŕňal prechádzanie textu a porovnávanie reťazcov, čo je operácia, ktorá môže byť dokončená v polynomiálnom čase v závislosti od veľkosti textu.

NP úloha:

Príklad NP úlohy pre neštruktúrované dáta by mohol byť problém identifikácie tém v neštruktúrovaných textových dátach, ako sú články alebo recenzie. Na identifikáciu tém by bolo možné použiť techniky ako LDA (Latent Dirichlet Allocation) alebo iné metódy zhlukovania. Problém spočíva v identifikácii najvhodnejšej sady tém, ktoré najlepšie reprezentujú dáta. Toto môže byť NP úloha, pretože overenie kvality konkrétnej sady tém (napríklad pomocou koherencie tém) môže byť rýchle, ale priestor možných kombinácií tém je obrovský, a preto môže byť ich identifikácia časovo náročná, najmä pre veľké súbory dát.

3 Vplyv veľkosti a štruktúry dát na algoritmickú zložitosť P a NP v kontexte Big Data

Veľkosť dát vs. štruktúrovanosť

Vo veľkých dátových setoch sa problém škálovania stáva ešte významnejším. Algoritmy, ktoré sa možno v malom meradle javia ako efektívne, často neškálujú dobre v prípade Big Data (Marz & Warren, 2015). Podobne, štruktúra dát, či už je jednoduchá alebo komplexná, môže zásadne zmeniť výpočtovú náročnosť.

P a NP pre Big Data s štruktúrovanými dátami

P úloha:

V prípade Big Data je efektívnym príkladom úlohy triedy P výpočet agregovaných hodnôt, ako je napríklad priemerný vek obyvateľov z obrovského datasetu občianskeho registra.

Predstavme si, že máme dataset s miliónmi záznamov o obyvateľoch jedného štátu. Každý záznam obsahuje rôzne atribúty ako meno, priezvisko, dátum narodenia, adresa atď. Našim cieľom je vypočítať priemerný vek obyvateľov. Toto je typická P úloha, pretože je možné ju vyriešiť v polynomiálnom čase. Distribuované výpočtové rámce ako MapReduce môžu túto úlohu rýchlo a efektívne vyriešiť. V prvom kroku (`map`) by sa jednoducho extrahovali vek každého jednotlivca a v druhom kroku (`reduce`) by sa spočítal celkový súčet a vydelením počtom záznamov získal priemer (Leskovec et al., 2022).

NP úloha:

NP úloha by mohla byť optimalizácia dopravných tokov vo veľkomeste. V tomto scenári by čas potrebný na overenie optimálneho riešenia mohol byť relatívne krátky, ale samotný proces nájdenia tohto optimálneho riešenia by mohol byť výpočtovo náročný.

Predstavme si veľkomesto s komplexnou sieťou ciest, kde každý deň dochádza k dopravným zápcham. Úlohou je optimalizovať dopravné toky tak, aby sa minimalizoval čas strávený v zápchach. To môže zahrňovať nastavenie semaforov, určenie preferovaných ciest pre verejnú dopravu, atď. Tento problém je NP-úloha, pretože aj keď je možné rýchlo overiť efektívnosť jedného konkrétneho riešenia (napr. simuláciou alebo analýzou reálnych dát), nájdenie najlepšieho možného riešenia z obrovského počtu kombinácií je výpočtovo veľmi náročné.

P a NP pre Big Data s semi-štruktúrovanými dátami

P úloha:

Vyhľadávanie vzorov v XML alebo JSON súboroch. Jedným príkladom úlohy triedy P by mohlo byť vyhľadávanie konkrétnych vzorov v obrovských XML alebo JSON súboroch, napríklad pomocou distribuovaných výpočtových rámov ako MapReduce. Máme XML súbor s miliónmi záznamov o používateľoch webovej aplikácie. Úlohou je nájsť všetkých používateľov, ktorí sú starší ako 18 rokov. Toto je úloha triedy P, pretože vyhľadávanie konkrétnych údajov môže byť vykonané v polynomiálnom čase. V rámci distribuovaného výpočtového rámca, ako je MapReduce, by jeden set 'mapperov' čítal dáta a identifikoval záznamy používateľov starších ako 18 rokov, a 'reducer' by tieto záznamy zbieral do výsledného zoznamu.

NP úloha:

NP problémom by mohlo byť rozpoznanie komunikačných vzorov v sociálnych sieťach, kde je štruktúra dát relatívne komplexná a nejednoznačná. Vychádzajme z analýzy sociálnej siete s miliónmi používateľov a ich vzájomných vzťahov. V tejto sieti sa snažíme nájsť 'partie' alebo uzavreté skupiny používateľov, ktorí komunikujú veľmi intenzívne medzi sebou. Rozpoznať tieto vzory môže byť veľmi časovo náročné, pretože by sa muselo prejsť všetkými možnými kombináciami používateľov a ich vzájomnými interakciami. Aj keď by sme rýchlo vedeli overiť, či konkrétna skupina používateľov tvorí 'partiu' alebo nie (napr. skúmaním počtu vzájomných interakcií), samotný proces nájdenia týchto skupín v obrovskom datasete je NP problém. Týmto spôsobom je možné lepšie pochopiť, ako sa úlohy triedy P a NP líšia v kontexte

vyhľadávania vzorov v štruktúrovaných alebo semi-štruktúrovaných dátach, ako sú XML a JSON súbory, alebo v komplexných štruktúrach, ako sú sociálne siete.

P a NP pre Big Data s neštruktúrovanými dátami

P úloha:

Extrakcia informácií z veľmi veľkého korpusu textových dát, ako je identifikácia a počítanie slov alebo fráz, je dobrým príkladom úlohy triedy P. Tento druh úlohy sa často využíva v rámci distribuovaných výpočtových technológií, ako je Hadoop alebo Spark. Vychádzajme z veľmi veľkej databázy novinových článkov a úlohou je zistiť, koľkokrát sa v nich spomína konkrétny termín, napríklad "globálne otepľovanie". Môžeme použiť distribuovaný výpočtový rámec ako MapReduce. V prvom kroku ("map") by každý uzol prešiel svojou časťou datasetu a spočítal výskyty pojmu. V druhom kroku ("reduce") by sa tieto čísla spočítali, čím by sa získal celkový počet výskytov. Toto je úloha triedy P, pretože je možné ju efektívne vyriešiť v polynomiálnom čase.

NP úloha:

Automatizovaná analýza sentimentu vo veľmi veľkých súboroch recenzií je dobrým príkladom NP problému. Overenie kvality analyzovaných dát je síce rýchle, ale nájdenie najlepšieho modelu pre analýzu sentimentu môže byť časovo náročné. Ako príklad je možné uviesť milióny recenzií z rôznych e-shopov a úlohou je zistiť, ako je spokojný zákazník s produktom alebo službou. Na tento účel potrebujeme vytvoriť model strojového učenia. Problém je, že existuje veľké množstvo rôznych modelov a techník, ktoré je možné použiť, napr. SVM-Support Vector Machines, neurónové siete, rozhodovacie stromy atď., a každá z nich má svoje vlastné hyper-parametre, ktoré je potrebné nastaviť. Na nájdenie najlepšieho modelu by bolo potrebné vykonať veľké množstvo výpočtov, ktoré by mohli trvať veľmi dlho. Hoci overenie kvality jedného modelu (napr. pomocou kros-validácie) je rýchle, nájsť ten "najlepší" model v obrovskom priestore možností je NP problém.

4 Vplyv veľkosti a štruktúry dát na algoritmickú zložitosť P a NP v kontexte NoS-NoB Data

V dnešnej dobe sú dáta neoddeliteľnou súčasťou každodenného života a zároveň kľúčovým zdrojom informácií pre podniky, vlády a výskumné inštitúcie. Hovoríme často o extrémoch - malých dátach, ktoré môžeme spracovávať manuálne alebo na jednom počítači, a veľkých dátach, ktoré vyžadujú distribuované výpočtové rámce a sofistikované algoritmy (Lin & Dyer, 2010). Ale čo s dátami, ktoré sú "nie malé, ale ani nie veľké", teda NoS-NoB (Not Small - Not Big) dátami?

NoS-NoB dáta predstavujú unikátnu výzvu, keďže sú príliš veľké na to, aby sa spracovávali na bežných desktopových počítačoch, ale zároveň príliš malé na to, aby ospravedlnili náklady a komplexitu distribuovaného výpočtového systému. Práve v tomto "zlatom strede" sa často stretávame s technologickými a metodologickými otázkami, ktoré vyžadujú zvláštny prístup. Či už ide o analýzu trhových trendov v stredne veľkých spoločnostiach, výskumné projekty s obmedzeným rozpočtom alebo vládne štúdie s citlivými údajmi, NoS-NoB dáta nám kladú otázku: Ako zoptimalizovať výpočtové zdroje a metódy analýzy tak, aby sme z dát získali maximálnu hodnotu bez zbytočného plytvania času a zdrojmi?

Veľkosť dát vs. štruktúrovanosť v kontexte NoS-NoB dát

Keď hovoríme o dátach, často zdôrazňujeme extrémy - malé a veľké dáta. No čo je s NoS-NoB (Not Small, Not Big) dátami, ktoré sú príliš veľké na bežné počítače, ale príliš malé na distribuované systémy? V týchto prípadoch sa stáva rovnako dôležitou nie len veľkosť, ale aj štruktúra dát (Easley & Kleinberg, 2019). Tento článok sa venuje problematike, ako veľkosť a štruktúra týchto stredne veľkých dátových setov ovplyvňujú výkon algoritmov triedy P a NP.

P a NP pre Nos-Nob štruktúrované dáta

P úloha:

V prípade štruktúrovaných NoS-NoB dát by jedným z príkladov úlohy triedy P mohla byť SQL operácia, ktorá vyhľadáva záznamy na základe špecifických kritérií v stredne veľkej relačnej databáze. Keďže tieto operácie sú veľmi dobre optimalizované, môže byť tento dopyt vykonaný v polynomiálnom čase.

NP úloha:

Keď ide o NP problémy, môžeme si predstaviť napríklad optimalizáciu dopravného toku v sieti logistiky strednej veľkosti. Získanie optimálnej cesty podľa viacerých kritérií je výpočtovo náročné, a teda sa radí do triedy NP.

P a NP pre Nos-Nob semi-štruktúrované dáta

P úloha:

Jedným z príkladov triedy P by mohlo byť vyhľadávanie v stredne veľkých XML dokumentoch podľa určitých tagov alebo atribútov. Zatiaľ čo štruktúra nie je tak pevná ako v relačných databázach, optimalizované vyhľadávanie môže byť aj tu efektívne. Ako príklad môže poslúžiť stredne veľký dataset v XML formáte o knihách v knižnici. Úlohou triedy P by mohlo byť vyhľadávanie všetkých kníh, ktoré boli vypožičané aspoň 10-krát.

NP úloha:

V prípade NP úloh by sa dalo hovoriť o analýze komplexných spoločenských sietí, kde je cieľom nájsť optimálne komunity alebo skupiny. NP problémom by mohla byť analýza interakcií na sociálnej sieti strednej veľkosti, kde by cieľom bolo identifikovať "vplyvných" jednotlivcov. Aj keď overenie, či niekto je vplyvný, môže byť jednoduché (pozrieť počet followerov), optimalizácia modelu, na dátových setoch, ktorých veľkosť už presahuje spracovateľské možnosti bežných PC, ale nedosahuje veľkosti databáz efektívne spracovateľné na distribuovaných systémoch, je to výpočtovo náročná úloha.

P a NP pre Nos-Nob neštruktúrované dáta

P úloha:

Ak vezmeme do úvahy neštruktúrované dáta, frekvenčná analýza slov v texte veľkosti NoS-NoB je jednoduchým príkladom P úlohy, ktorý môže byť vykonaný rýchlo a efektívne.

NP úloha:

Analýza sentimentu alebo tematická kategorizácia textu by sa mohla radiť do triedy NP, zvlášť ak sa využívajú komplexné modely na rozpoznávanie sentimentu alebo tém, ktoré vyžadujú viacero vstupných premenných a parametrov. Vychádzajme z kolekcie recenzií na

rôzne produkty, veľkosti ako v predchádzajúcom prípade a úlohou by bolo automaticky určiť, ktoré recenzie sú pozitívne a ktoré negatívne. Na tento účel by bolo vhodné vytvoriť sofistikovaný model strojového učenia, a nájdenie najlepšieho modelu by bola NP úloha.

5 Záver

V dnešnom, rýchlo sa meniacom, svete dátových analýz je nevyhnutné chápať, ako rôzne typy a veľkosti dát ovplyvňujú výpočtovú náročnosť a výkon algoritmov. Ako vidno, veľkosť a štruktúra dát ovplyvňujú výkon algoritmov a výpočtovú zložitosť. Dáta sa môžu líšiť nielen vo veľkosti, ale aj v štruktúrovanosti, a tieto dve dimenzie majú zásadný vplyv na efektivitu algoritmov použitých na ich analýzu. Tento článok sa zameriava na tri kategórie dát: malé dáta (Small Data), veľké dáta (Big Data) a dáta často označované ako dáta strednej veľkosti, ktoré sú príliš veľké na bežné počítače, ale príliš malé na distribuované systémy (NoS-NoB Data).

Small Data - V prípade malých dát je kľúčovým faktorom efektivity algoritmu štruktúra dát. Veľkosť dát je obvykle dostatočne malá na to, aby sa dala efektívne spracovať na jednom počítači. Príklady P úloh v tejto kategórii zahŕňajú lineárne vyhľadávanie v malom zozname a vyhľadávanie kľúčových slov v JSON súbore. Komplexnejšie úlohy, ako je problém obchodného cestujúceho (TSP) alebo klasterizácia zákazníkov podľa hodnotení, sú tiež relevantné, ale sú NP-zložité.

Big Data - V kontexte veľkých dát je škálovateľnosť algoritmov kritická. Veľkosť dát je taká, že distribuované spracovanie je nevyhnutné. P úlohy v tejto kategórii zahŕňajú výpočet agregovaných hodnôt a vyhľadávanie vzorov v XML alebo JSON súboroch. Optimalizácia dopravných tokov a analýza sentimentu v obrovských súboroch recenzií sú príklady NP úloh, ktoré sú relevantné pre veľké dáta.

NoS-NoB Data - V tejto kategórii sú dáta príliš veľké na bežné počítače, ale príliš malé na distribuované systémy. SQL operácie na stredne veľkej databáze a frekvenčná analýza slov v texte sú príklady P úloh. Optimalizácia dopravného toku v sieti logistiky a analýza sentimentu alebo tematická kategorizácia textu sú príklady NP úloh v tejto kategórii.

Jedným z kľúčových zistení tohto článku je, že efektivita algoritmov výrazne závisí od dvoch hlavných faktorov: kategorizácie problému do P alebo NP triedy a štruktúrovanosti príslušného dátového setu. Tieto dve dimenzie spolu s typom úlohy nám umožňujú s vysokou presnosťou odhadnúť, do akej kategórie dát (Small, Big, NoS-NoB) daná úloha spadá. Tento odhad je kritický pre výber najvhodnejšej technológie spracovania dát. V praxi to znamená, že pred zahájením akéhokoľvek projektu spracovania dát je nevyhnutné správne identifikovať tieto parametre. Tým sa otvára cesta k optimalizácii zdrojov a času, a umožňuje nám to využiť najefektívnejšie metódy a nástroje pre konkrétnu úlohu. Tento prístup nie je len teoreticky zaujímavý, ale aj prakticky užitočný, pretože vedie k rýchlejším a efektívnejším riešeniam. V Tab. 1 sme uviedli orientačný odhad, nakoľko kvantifikovanie nie je možné, veľkosti dát pre small, NoS-NoB a big data, na základe štruktúrovanosti dát, príslušnosti k P, NP úlohám a konkrétneho typu úloh. Na základe konkrétneho typu úlohy je spätne možné predpokladať o aké veľkosti dát pôjde a aká informačná technológia by bola potrebná na riešenie takýchto úloh.

Tab. 1: Orientačné veľkosti dát v závislosti od kategórií (P a NP) a typu úlohy

Veľkosť dát	Štruktúrovanosť dát	Kategória (P, NP)	Typ úlohy
Small < 1GB	Štruktúrované	P	Lineárne vyhľadávanie
		NP	Problém obchodného cestujúceho
	Semi-štruktúrované	P	Vyhľadávanie kľúčových slov v JSON
		NP	Klasterizácia zákazníkov
	Neštruktúrované	P	Textová analýza
		NP	Predikcia trendov na sociálnych sieťach
NoS-NoB 1GB - 10GB	Štruktúrované	P	Pokročilé SQL operácie
		NP	Optimalizácia dopravného toku
	Semi-štruktúrované	P	Analýza sieťovej prevádzky
		NP	Predikcia chovania používateľov
	Neštruktúrované	P	Text mining
		NP	Rozpoznávanie hlasu
Big > 10GB	Štruktúrované	P	Výpočet agregovaných hodnôt
		NP	Optimalizácia dopravných tokov
	Semi-štruktúrované	P	Vyhľadávanie vzorov v XML
		NP	Komplexná analýza sociálnych sietí
	Neštruktúrované	P	Hlboké učenie na textových dátach
		NP	Automatické generovanie textu

Zdroj: Vlastné spracovanie

Literatúra

[1] Arora, S., & Barak, B. (2016). *Computational complexity a modern approach*. Cambridge University Press.
[2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. The MIT Press.
[3] Easley, D., & Kleinberg, J. (2019). Networks, crowds and markets: Reasoning about a highly connected world. Cambridge University Press.
[4] Leskovec, J. et al. (2022) *Mining of Massive Datasets*. Cambridge University Press.
[5] Lin, J., & Dyer, C. (2010). *Data-intensive text processing with mapreduce*. Morgan & Claypool.

[6] Marz, N., & Warren, J. (2015). Big data: Principles and best practices of Scalable Realtime Data Systems. O'reilly media.
[7] Peng, R. D., & Matsui, E. (2016). *The art of data science a guide for anyone who works with data*. Skybridge Consulting LLC.