TDT4125 Algoritmekonstruksjon

Eksamen, 27. mai 2021, 09:00–13:00

Faglig kontakt Magnus Lie Hetland
Hjelpemiddelkode A

Løsningsforslag

Løsningsforslagene i rødt nedenfor er *eksempler* på svar som vil gi full uttelling. Det vil ofte være helt akseptabelt med mange andre, beslektede svar, spesielt der det bes om en forklaring eller lignende. Om du svarte noe litt annet, betyr ikke det nødvendigvis at du svarte feil!

På grunn av koronapandemien er dette en hjemmeeksamen med alle hjelpemidler tillatt. Følgelig bør ikke eksamenssettet ses som representativt for ordinære eksamener.

1 Ford–Fulkerson-metoden for bipartitt matching går ut på å finne én og én forøkende sti. Din venn Smartnes mener dette går for langsomt, og mener det vil være bedre å finne flere forøkende stier på én gang. Hva tenker du om planen hennes? Forklar og diskuter.

> Her er det i praksis snakk om HOPCROFT-KARP (som finner en maksimal mengde disjunkte forøkende stier i hver iterasjon) eller lignende metoder, og den har en bedre kjøretid ($O(\sqrt{n}m)$) enn FORD-FULKERSON med BFS (altså EDMONDS-KARP, $O(nm)$).
>
> Her er hovedpoenget å koble problemstillingen til HOPCROFT-KARP. Om man gjør det, og konkluderer med at kjøretiden forbedres, gir det full uttelling. (Man trenger ikke eksplisitt oppgi kjøretiden, siden den uansett lett kan finnes i fagstoffet, om man har navnet på algoritmen.)
>
> Noe uttelling kan også gis om man gir et fornuftig resonnement på egen hånd for hvorfor det kan føre til en forbedring, uten å eksplisitt referere til HOPCROFT-KARP.

2 Gi en overordnet forklaring, med egne ord, på hvordan den ungarske metoden (KUHN-MUNKRES) fungerer. Hvorfor gir det mening å kalle den en primal–dual-algoritme? Forklar og diskuter.

Her kan mange forklaringer godtas. Det er essensielt at man har med en diskusjon av dualvariable knyttet til nodene, i motsetning til primalvariable knyttet til kantene, der primalvariablene er binære, og sier om kanten skal være med eller ikke, mens dualvariablene er en form for «betaling», der nodene må betale for (kostnaden til) kantene, så primal- og dual-målverdier blir like.

Man bør også diskutere likhetsgrafen, og at det er snakk om å finne en perfekt matching i den – og at om man ikke finner en perfekt matching, så øker man dualvariable for å utvide likhetsgrafen.

Koblingen til primal–dual-metoden generelt er ideen om at man bygger en stadig «mer gyldig» superoptimal heltallig primalløsning (matchingen) samtidig som man forbedrer en gyldig, suboptimal dualløsning, gjerne for å «betale» for binære primalvariable (som i mengdedekkeproblemet).

3 Gi en overordnet beskrivelse av LP-relaksering og randomisert avrunding. Hvilke egenskaper har løsningen og hvilke utfordringer støter man gjerne på? Forklar og diskuter.

En slakking eller relaksering handler om å utvide mulighetsområdet, dvs., å la flere løsninger være gyldige, mens man beholder målfunksjonen (dvs., de løsningene som opprinnelig var gyldige har fortsatt samme målverdi). Begrepet LP-relaksering brukes primært om å fjerne heltallskravet fra et lineært heltallsprogram (MILP), slik at man ender med et vanlig lineærprogram. Optimum vil da kunne finnes i polynomisk tid, og ha minst like god målverdi som for heltallsprogrammet, men løsningen man finner er ikke nødvendigvis heltallig (og dermed ikke nødvendigvis gyldig for det opprinnelige programmet).

For å kunne bruke LP-løsningen, bruker man f.eks. randomisert avrunding. En vanlig fremgangsmåte er som følger. Dersom en variabel $x_i \in \{0, 1\}$ har blitt relaksert til $0 \leqslant x_i \leqslant 1$, runder man av slik at $\mathrm{P}[x_i = 1] = x_i$. Den forventede målverdien til den avrundede løsningen vil da være lik målverdien til lineærprogrammet, og altså minst like bra som optimum for heltallsløsningen – man har altså en randomisert løsning med (super-)optimal forventningsverdi.

Problemet er at løsningen ikke nødvendigvis er gyldig, siden avrundingen kan bryte restriksjoner. Men siden forventningsverdi er lineært, så vil den *i forventning* oppfylle alle restriksjonene, dvs., for en restriksjon $a_i x_i \leqslant b_i$ så vil den randomiserte avrundingen gi $\mathrm{E}[a_i x_i] \leqslant b_i$.

Man kan generelt ikke forhindre at noen restriksjoner blir brutt (f.eks. ved noen form for derandomisering), for om man klarte det, ville man kunne løse heltallsprogrammer i polynomisk tid, og dermed ha $\mathrm{P} = \mathrm{NP}$.

4 Du har gravd ned en skatt i strandkanten, men er usikker på hvor. Du starter på et vilkårlig punkt og begynner å gå frem og tilbake langs stranden med en metalldetektor. For å minimere hvor langt du må gå totalt, så dobler du hvor langt du går fra startpunktet for hver gang du snur og går tilbake.

Tenk på dette som en form for onlinealgoritme, der offlinevarianten innebærer at du vet om skatten ligger mot nord eller sør, og du går i riktig retning fra starten av, til du finner skatten. Hva blir ytelsesgarantien (*competitive ratio*)? Forklar og diskuter.

Dette tilsvarer det som kalles *the line search problem* (uten at man trenger å vite det).

Offlinevarianten vil bevege seg direkte i riktig retning, og treffe på skatten, med avstand d. I verste tilfelle, er denne avstanden *litt* større enn en toerpotens, så onlinevarianten må kjøre to iterasjoner ekstra (altså gå tilbake til start, så bevege seg en avstand ca. $2d$ til venstre, og deretter være på vei til $4d$ i riktig retning, der man støter på skatten i avstand d). Dvs., $d = 2^i + \epsilon$, for en vilkårlig liten $\epsilon > 0$. Strekningene vekk fra startpunktet blir altså en sum av potenser av 2 opp til 2^{i+1} før man så får det siste strekket på d, og hver av disse må tilbakelegges i begge retninger, så man får

$$2 \cdot (1 + 2 + \cdots + 2^i + 2^{i+1}) + d \leqslant 2 \cdot 2^{i+2} + d \leqslant 8d + d = 9d\,.$$

Man får altså en ytelsesgaranti på $\alpha = 9$.

5 Du har oppgitt følgende lineærprogram:

$$\min\ cx\ :\ Ax \geqslant 1$$

I hver av radene i A er én av verdiene 1, én av verdiene -1, og resten 0. Du er bare interessert i å avgjøre om programmet er *gyldig* (*feasible*), og kan dermed anta at $c = 0$.

Forklar og diskuter hva dualen til programmet blir, hvordan du kan bruke dualen til å finne ut om primalen er gyldig, og hvordan du kan gjøre det med en velkjent kombinatorisk algoritme.

Dette blir i praksis et spørsmål om hvorvidt et sett med ligninger $x < y$ (eller $x + 1 \leqslant y$) er konsistent. Man kan sjekke om primalen er *gyldig* ved å se om dualen er *ubegrenset*.

Her er det mulig å tolke A som en rettet graf, der hver kolonne er en node og hver rad er en kant, som går fra sin -1-kolonne til sin $+1$-kolonne. Vi kan tenke på dualvariablene som en form for flyt (eller sirkulasjon) uten noen kapasitetsbegrensning. Dualrestriksjonene sier da at summen for hver node skal være 0 (med likhet, siden x er fri), og det tilsvarer flytbevaring i nodene. Dvs., om man vil øke én dualvariabel – altså øke flyten i en kant – så må man også øke flyten i en kant inn til startnoden og en kan ut av sluttnoden. Den eneste måten å få til dette på er om man har en sykel. Dvs., om man finner en sykel, så vil dualen være ubegrenset. En slik sykel kan man f.eks. finne med DFS.

(Man kan også se direkte at en sykel i en serie med strenge ulikheter betyr at ulikhetene er inkonsistene, altså at primalen er ugyldig.)

6 Du har oppgitt en vektet rettet asyklisk graf, og skal finne den tyngste delgrafen som er en rettet skog, med kanter fra foreldre til barn.

- Hvor bra vil en grådig algoritme fungere?
- Hvor bra vil en grådig algoritme fungere dersom du skal finne den *letteste* løsningen i stedet?
- Hvor bra vil en grådig algoritme (for den tyngste løsningen) fungere, dersom du har en vilkårlig rettet (ikke nødvendigvis asyklisk) graf?

Forklar og diskuter.

Løsningen karakteriseres ved en hodepartisjon – maks én rettet kant i løsningen ha hodet mot en gitt node. Dette er en matroide, så grådighet vil gi optimum.

Siden alle basiser i en matroide har samme kardinalitet, kan man også bruke grådighet der man velger den *letteste* kanten, til man ikke kan legge til flere.

Om grafen ikke nødvendigvis er asyklisk, kan man ta snittet med den grafiske matroiden for den underliggende urettede grafen; siden man har snittet av to matroider, får vi $\alpha = 2$.

7 Du skal fordele oppgaver på identiske maskiner $i = 1 \ldots m$, der hver jobb $j = 1 \ldots n$ har prosesseringstid p_j. Du vil minimere sluttiden, C_{max}. Din venn Lurvik har et forslag til løsningsstrategi:

> Fordel jobbene utover maskinene, med uniform sannsynlighet. Hver maskin får da forventet sluttid $t = t_j = \sum_j p_j / m$, og den forventede sluttiden for hele problemet blir $\max_j t_j = t$. Du kan så derandomisere for å få en deterministisk løsning med $C_{max} = t$.

Diskuter løsningen til Lurvik. Argumenter generelt for hvorfor du tror det er mulig eller umulig å lage en algoritme med den garantien Lurvik kommer med, og mer spesifikt for hvorfor du mener Lurviks metode er korrekt eller ikke.

Generelt er dette urealistisk, siden vi da er garantert å finne optimum, som medfører $P = NP$. Mer spesifikt er det galt at forventningen til maksimum er maksimum over forventningene.

8 Du har et sett med ansatte, som hver er ekspert på et sett med temaer. Ekspertisen deres er diversifisert, slik at ethvert par med ansatte har maksimalt ett tema til felles. Du skal konstruere en komité som består av maksimalt k eksperter som tilsammen dekker alle temaene, eller rapportere at det er umulig å konstruere en slik komité. Beskriv hvordan du effektivt kan redusere en instans til en problemkjerne (*kernel*). For enkelhets skyld kan du anta at du *ikke* er interessert i å begrense antall ansatte. Hvor mange temaer vil kjernen maksimalt bestå av? Forklar og diskuter.

Dette er ekvivalent med oppg. 2.4 hos Cygan mfl. (om *point–line cover*).

Dersom en ansatt A er ekspert på flere enn k områder, *må* hun bli med; ellers måtte du ha tatt med flere enn k andre, siden hver av dem maksimalt er ekspert på ett av A sine temaer. Ta med A, slett A og A sine temaer, og la $k = k - 1$.

Om du går tom for temaer, men $k \geqslant 0$, har du løst problemet.

Om $k = 0$, og du har igjen temaer, så er problemet umulig å løse.

Ellers sitter du igjen med $k > 0$ og du har igjen m temaer. Du har maksimalt lov til å ta med k personer, og hver av dem dekker maksimalt k av temaene, så om $m > k^2$ temaer igjen, er svaret nei. Ellers sitter du igjen med en problemkjerne med $m \leqslant k^2$ temaer.