DA-OPT 3900: Prosjektrapport for obligatorisk oppgave 1

Amer Sisic, Thor-Stian Follstad og Martin Gran

20. oktober 2016

Sammendrag

This project report describes the implementation of together four different algorithms of the Travelling salesman problem"(TSP). First a symetrical complete graph is implemented with random lengths of 1-10 between a random number of cities. Then a route is chosen on random and the cost is calculated. The next solution is to choose routes at random and updating the best route by the lowest cost. Third a greedy method is implemented. In the implementation the next city in the route is chosen by selecting the path with the lowest value. Fourth a greedy itterative method is chosen. The method works on a complete route, then switches cities at random and updates the route if the total cost is improved. The fourth method works on routes generated by the previous methods. Last the different solutions are run 100 times at the same graph. The best result and the mean result of the methods are then compared.

Innhold

1 Innledning **3**

2 Teoretisk bakgrunn **3**

2.1 Travelling Salesman problemet 3

3 Implementering av TSP-algoritmer **4**

3.1 Regn ut totalkostnad 4

3.2 Random metode 4

3.3 Itterativ Random metode 5

3.4 Grådig metode 5

3.5 Grådig forbedringsmetode 6

4 Testing av metoder **6**

5 Resultater **7**

5.1 TSP: 100 byer 7

5.2 TSP: 500 byer 8

5.3 TSP: 1000 byer 9

6 Konklusjon **10**

Referanser **11**

Denne prosjektrapporten beskriver fire forskjellige implementasjoner av algoritmer for å løse the Travelling Salesman problem"(TSP). Løsningen og rapporten er utarbeidet av Amer Sisic, Thor-Stian Follstad og Martin Gran. Løsningsforslaget implementerer fire forskjellige algoritmer forskjellige algoritmer for å finne den beste løsningen av problemet. De fire algoritmene er tilfeldig metode (random metode), tilfeldig itterativ metode (random itterativ method), grådig metode (greedy method) og grådig itterativ metode (greedy itterativ method). De implementerte algoritmene kjøres deretter 100 ganger på en identisk fullstendig, symetrisk graf som representerer byene med avstander. Deretter sammenlignes resultatene de forskjellige algoritmene produserer. Først sammenlignes det beste resultatet hver enkelt algoritme klarte å produsere. Deretter sammenliknes det gjennomsnittlige resultatet hver enkelt algoritme produserer.

Rapporten vil først drøfte den teoretiske bakgrunnen for TSP. Deretter vil rapporten peke på avveininger gjort i forhold til implementering av algoritmene. Tilslutt vil oppgaven drøfte resultatene med bakgrunn i resultat i form av tid og kostnad.

2 Teoretisk bakgrunn

2.1 Travelling Salesman problemet

Opprinnelsen til Travelling Salesman problemkan spores tilbake til 1800-tallet. Den nevnes blant annet i en bok for omreisende selgere i Sveits og Tyskland. Problemet ble formulert matematisk på 1800-tallet av W.R Hamilton og Thomas Kirkman. Den generelle formen av TSP ble siden studert av matematikere på 1930-tallet, mest kjent av Karl Menger (Biggs, Lloyd & Wilson, 1976). Problemet er populært innen vitenskapelige kretser og har blitt videre studert innenfor matematikk, kjemi, fysikk og datateknikk (Lawler, Lenstra, Kan & Shmoys, 1985).

TSP kan bli modellert som en urettet graf der byene er grafens toppunkter, mens veiene er grafens baner. TSP kan løses eksakt, men vil by på problemer bare med så få som 20 byer. Kompleksiteten for å løse problemet eksakt er blitt anslått til $O(n!)$. Videre er det ikke blitt avgjort hvorvidt det eksisterer en algoritme som løser TSP eksakt i $O(1.9999^n)$ (Woeginger, 2003).

En løsning på problemet er derfor å benytte heuristikker som finner en god nokløsning på problemet. Heuristikker kan i denne sammenheng betraktes som tilnærmingsalgoritmer. Mo-

derne algoritmer kan finne løsninger som med høy sannsynlighet er 2-3% unna den optimale løsningen (Rego, Gamboa, Glover & Osterman, 2011).

3 Implementering av TSP-algoritmer

Algoritmene beskrevet er basert på pseudokode fra forelesninger som omhandler oppgaven og TSP. For enkelhets skyld vil pseudokodene bli forklart, mens kildekoden kan hentes på github.com.

3.1 Regn ut totalkostnad

Metoden er skrevet for å enkelt kunne gjennbruke kode. De resterende metodene velger ut en rute som metoden returnerer kostnaden fra. Metoden er lik i alle implementasjonene av TSP-heurestikkene.

```
Tar imot rute med byer som array og graf med alle byene;
Variabel totalKostnad;
For (< antall byer) inkrementer
        totalKostnad = totalKostnad + avstand til neste by;
totalKostnad = totalKostand + avstand mellom første og siste by;
Returner totalKostnad.
```

Figur 1: Pseudokode av metoden for utregning av kostnad for en rute

3.2 Random metode

I den Random metode"velges det en tilfeldig vei mellom antallet byer. Dette gir algoritmen kompleksitet $O(n)$. Algoritmen er rask å implementere, men gir et lite optimalt resultat.

```
Velg en tilfeldig by;
Marker byen som besøkt;
While(alle byer ikke er besøkt)
        Velg en tilfeldig by;
        Marker byen som besøkt;
        Koble sammen byen til forrige valgte by;
Returner ruten;
Kall metode Regn ut kostnad av ruten;
```

Figur 2: Pseudokode for Random metode

3.3 Itterativ Random metode

Den random itterative metoden likner på random metode. Den skiller seg imidlertid ved at den forsøker å trekke en tilfeldig rute flere ganger. Metoden lagrer den nåværende beste ruten og sammenlikner andre tilfeldige trekk med denne. Skulle en rute med lavere kostnad bli trukket, oppdateres beste rute. Metoden fortsetter frem til et gitt stoppkriterium forekommer. Kompleksiteten til random itterativ metode kan uttrykkes som $O(stoppkriterium) \cdot (n)$.

```
Variabel beste rute;
while(antall itterasjoner)
        Random metode;
        Hvis return Random metode < Beste Rute:
        Antall itterasjoner økes.
                Beste rute = return Random metode;
returner Beste Rute.
```

Figur 3: Pseudokode for Random Itterativ metode

3.4 Grådig metode

Den grådige metoden velger et tilfeldig startpunkt blant byene. Deretter vil den bevege seg til den ubesøkte byen som ligger nærmest nåværende by. Metoden krever en utregning av avstanden mellom nåværende posisjon og antallet gjenstående ubesøkte byer. Den grådige metoden begrenses av at den vil finne det lokale optimum og kan gå glipp av et globalt optimum (Black, 2004). Algoritmen vil ha en kompleksitet på $O(n^2)$

```
Start i tilfeldig by;
Sett tilfeldig by i starten av ruten;
Byen markeres som besøkt;
for(antall byer)
        for(antall byer)
                Finn nærmeste by som ikke er besøkt;
        Hopp til nærmeste ubesøkte by og plasser i ruten;
        Marker byen som besøkt;
returner Rute.
```

Figur 4: Pseudokode for Greedy Method

3.5 Grådig forbedringsmetode

Den grådige forbedringsmetoden baserer seg på å forsøke å forbedre eksisterende ruter. Den er avhengig av en rute for å forsøke å forbedre denne. I denne implemntasjonen får metoden ruter fra de foregående metodene. Dette gjøres ved å velge to tilfeldige byer, bytter deres plasser med hverandre for så å regne ut rutens kostnad. Hvis ruten forbedres så skifter algoritmen den beste ruten. Grådig forbedringsmetode har til forskjell fra grådig metode potensiale til å finne det globale optimum. Grådig forbedringsalgoritme stoppes av et stoppkriterium. Denne implementasjonen benytter et gitt antall itterasjoner. Kompleksiteten vil da være utregning av kostnad ganget med antall itterasjoner. Dette kan skrives som $O(metode\ for\ generering\ av\ rute + stoppkriterium \cdot n)$

```
Få generert rute;
for(antall itterasjoner)
        bytt to tilfeldige byer i ruten;
        kall metode for utregning av kostnad;
        hvis skiftetRute < besteRute;
                bestRute = skiftRute;
        ellers
                reverser byttingen av byer;
returner besteRute.
```

Figur 5: Pseudokode for Greedy Improved Method

4 Testing av metoder

For å teste metodene ble det laget et testprogram for å kjøre alle metodene. Meningen med testene har vært å vise intervallene man kan regne med å få verdier innenfor med gitt datasett. Videre skal testen demonstrere forskjeller mellom de ulike metodene. For å få til dette ble testen gjennomført følgende forutsetninger.

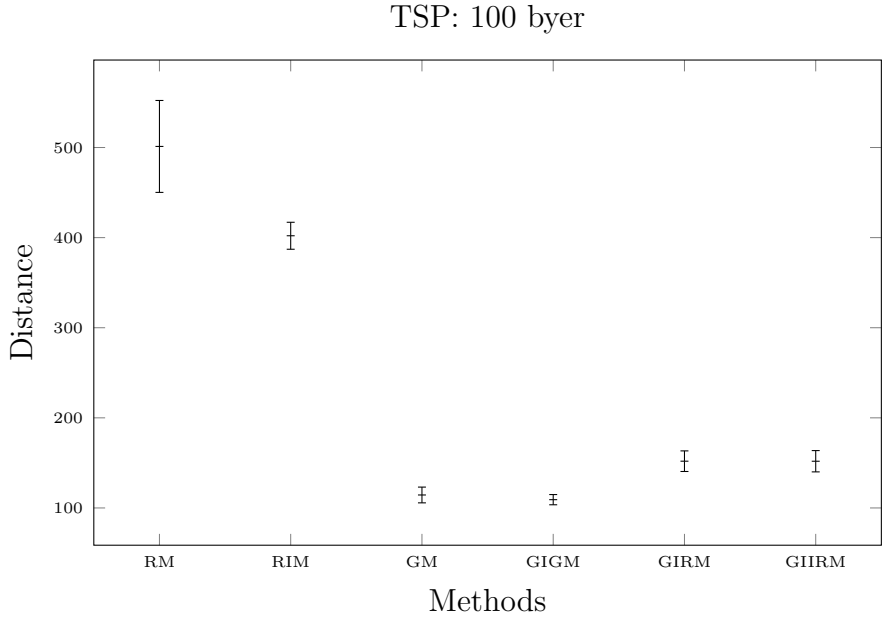
1. Det ble dannet tre datasett, fullstendige grafer, med henholdsvis 100, 500 og 1000 byer
2. Grafene ble fylt symetrisk med tilfeldige verdier mellom 1 og 10
3. De ulike metodene jobber på identiske grafer av de ulike størrelesesordnene
4. Antall triesble satt til 10.000 i Random Itterativ og Greedy Improved metode
5. Hver metode ble kjørt 1000 ganger på hver av de tre grafene

5 Resultater

Resultatene av metodene blir presentert her. Som redgjort i del 4 har hver test blitt kjørt 1000 ganger. Resultatene presenteres først for grafen med 100 byer, deretter 500 og til slutt 1000. Hver metode presenteres med et gjennomsnittlig resultat, konfidensintervall på $\sigma 1.96$ og beste resultat. For ordens skyld vises kun snitt og konfidensintervall i plott, mens beste resultat inkluderes i tabellen.

5.1 TSP: 100 byer

Testen gjennomføres her på en graf med 100 byer og tilfeldige avstander i intervallet 1-10.



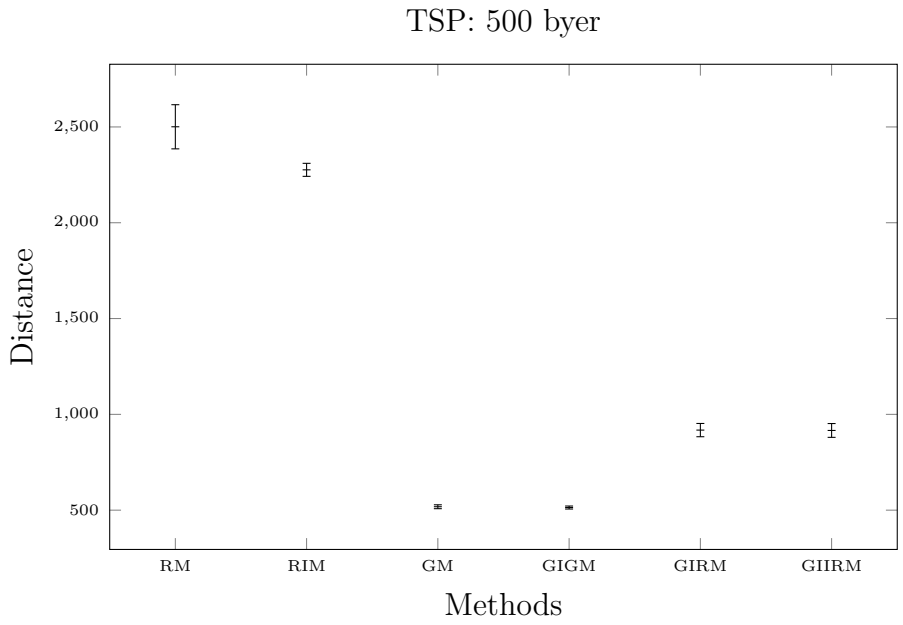
Figur 6: Viser gjennomsnittsverdi og konfidensintervall for Random Method (RD), Random Itterativ Method (RIM), Greedy Method (GM), Greedy Improved Greedy Method (GIGM), Greedy Improved Random Method (GIRM) og Greedy Improved Itterativ Random Method (GIIRM) i graf med 100 byer og tilfeldige avstander mellom 1-10

	Beste Resultat	Gjennomsnitt	SD
Random Method	418	501,20	26,02
Random Itterative	356	402,07	7,64
Greedy Method	106	114,40	4,46
Greedy Improved Greedy	102	109,19	2,89
Greedy Improved Random	134	151,90	5,83
Greedy Improved Itterative Random	135	151,84	6,03

Tabell 1: Viser resultatet av test med 1000 gjennomkjøringer i graf med 100 byer og tilfeldige avstander i inervallet 1-10.

5.2 TSP: 500 byer

Testen gjennomføres her på en graf med 500 byer og tilfeldige avstander i intervallet 1-10.



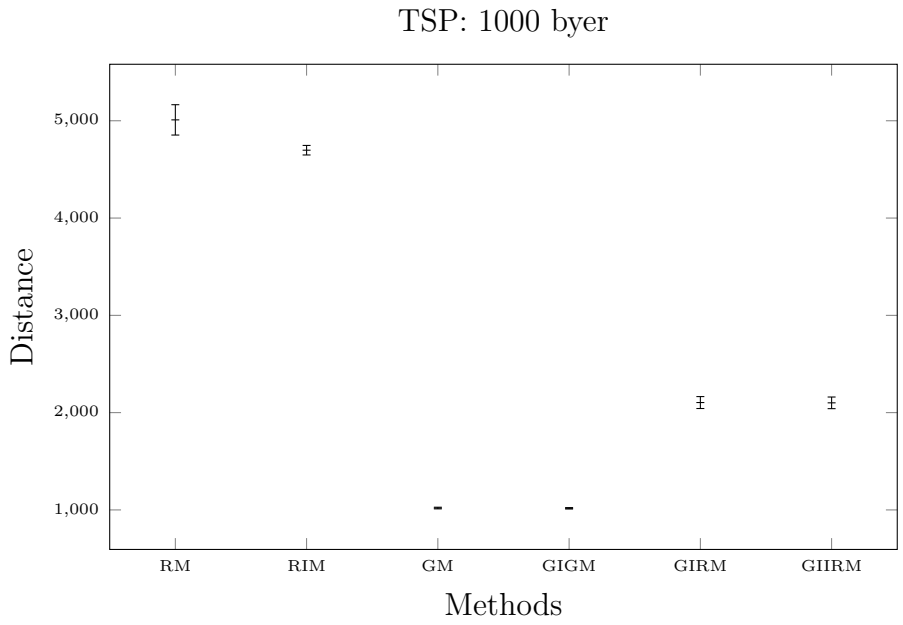
Figur 7: Viser gjennomsnittsverdi og konfidensintervall for Random Method (RD), Random Itterativ Method (RIM), Greedy Method (GM), Greedy Improved Greedy Method (GIGM), Greedy Improved Random Method (GIRM) og Greedy Improved Itterativ Random Method (GIIRM) i graf med 500 byer og tilfeldige avstander mellom 1-10

	Beste Resultat	Gjennomsnitt	SD
Random Method	2315	2500,88	58,78
Random Itterative	2210	2276,16	17,33
Greedy Method	506	517,83	5,18
Greedy Improved Greedy	504	513,73	3,97
Greedy Improved Random	866	917,83	17,67
Greedy Improved Itterative Random	854	916,01	18,38

Tabell 2: Viser resultatet av test med 1000 gjennomkjøringer i graf med 500 byer og tilfeldige avstander i inervallet 1-10.

5.3 TSP: 1000 byer

Testen gjennomføres her på en graf med 1000 byer og tilfeldige avstander i intervallet 1-10.



Figur 8: Viser gjennomsnittsverdi og konfidensintervall for Random Method (RD), Random Itterativ Method (RIM), Greedy Method (GM), Greedy Improved Greedy Method (GIGM), Greedy Improved Random Method (GIRM) og Greedy Improved Itterativ Random Method (GIIRM) i graf med 1000 byer og tilfeldige avstander mellom 1-10

	Beste Resultat	Gjennomsnitt	SD
Random Method	4760	5009,24	79,56
Random Itterative	4569	4697,55	25,02
Greedy Method	1007	1019,33	4,31
Greedy Improved Greedy	1006	1016,59	3,67
Greedy Improved Random	1998	2103,69	31,28
Greedy Improved Itterative Random	2007	2100,75	30,54

Tabell 3: Viser resultatet av test med 1000 gjennomkjøringer i graf med 1000 byer og tilfeldige avstander i inervallet 1-10.

6 Konklusjon

Resultatet av testene viser at Random metode og Random Itterativ metode i snitt gir et resultat som samsvarer med forventningen. Fordi det trekkes tilfeldige avstander mellom 1-10 bør derfor en tilfeldig valgt rute i snitt gi en kostnad på $\frac{10}{2} \cdot byer$. Det kan imidlertid nevnes at Random Itterativ metode i snitt er litt bedre enn Random metode. Grafene viser at konfidensintervallene ikke overlapper noe som indikerer at man med 95% sjanse kan si at metoden vil gi et bedre resultat på dette datasettet.

Det interessante resultatet kan sies å være den marginale forbedringen Greedy Improved metode har i forhold til Greedy Metode. Det er kun når det initialiserende array kommer fra Greedy Metode at følgende datasett gir et bedre resultat. Dette gjelder både snittresultatet og beste resultat. Når Greedy Improved Metode initialiseres med ruten fra Random Metode eller Random Itterativ metode så er Greedy metode i snitt bedre. I ingen av datasettene har Greedy metode overlappende konfidensintervall med de nevnte metoder og man kan med sikkerhet si at metoden i 95% av tilfeller vil klare å fremskaffe et bedre resultat.

Resultatet kan være overraskende og krever noe forklaring. Den første forklaringen kan være at Greedy metode er svært effektiv når antallet byer overgår intervallet som avstander kan trekkes mellom. Fordi mulige avstander mellom byer ligger mellom 1 og 10 så vil i snitt hver by være koblet til 10 andre byer med laveste kostnad i grafen med 100 byer. I grafen med 500 byer vokser antallet til 50 og så til 1000 på tusen byer. Metoden har derfor gode muligheter til å finne en rute bestående av avstender med laveste verdi.

En annen mulighet er også at Greedy Improved metode har et dårlig stoppkriterium. Det er mulig 10.000 byttinger med disse antallene byer er for få og at metoden ville vist seg likeverdig hvis itterasjonene hadde fått fortsette.

En forsiktig konklusjon basert på disse resultatene er at Greedy Metode gir et raskt og svært godt resultat til sammenlikning med Greedy Improved metode. Unntaket er likevel Greedy Improved metode som forbedrer ruten som Greedy metode først skaffer. Likevel viser resultatene at marginale og at konfidensintervallene overlapper. Det er derfor mulig at Greedy Improved metode med Greedy rute ikke klarer å forbedre den ruten den allerede har mottatt.

Referanser

Biggs, N., Lloyd, E.K. & Wilson, R.J. (1976). *Graph theory, 1736-1936*. Oxford University Press.

Black, P.E. (2004). *Dictionary of algorithms and data structures*. National Institute of Standards and Technology.

Lawler, E., Lenstra, J.K., Kan, A.R. & Shmoys, D. (1985). The traveling salesman problem; a guided tour of combinatorial optimization.

Rego, C., Gamboa, D., Glover, F. & Osterman, C. (2011). Traveling salesman problem heuristics: leading methods, implementations and latest advances. *European Journal of Operational Research*, *211*(3), 427–441.

Woeginger, G.J. (2003). Exact algorithms for np-hard problems: A survey. I *Combinatorial optimization—eureka, you shrink!* (s. 185–207). Springer.