P1

Genetiske algoritmer til generering af skoleskemaer i folkeskoler

Fra eksisterende software til modeller

B2-26

Casper Susgaard Nielsen

Kasper Christoffer Andersen

Malthe Søgaard Sørensen

Rasmus Mariegaard

Simon Dam Nielsen

Vejledere

Claus Skaaning

Søren Løkke

Afleveringsdato: 19/12 2016

_____ _____

_____ _____ _____

AALBORG UNIVERSITET

AALBORG UNIVERSITET
STUDENTERRAPPORT

Første Studieår
Software
Strandvejen 12-14
9000 Aalborg
http://tnb.aau.dk

Titel:

Tema:

Projektperiode:

Projektgruppe:

Deltagere:

Vejledere:

Synopsis:

Oplagstal:
Sideantal:
Bilagsantal og -art:
Afsluttet den

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Contents

1 Indledning **3**

2 Problemanalyse **4**

2.1 Skeamets opbygning 4

2.2 Lovmæssige krav til folkeskoleskemaet 5

2.2.1 Skolereformen 7

2.3 Case study - Sofiendalskolen 7

2.4 Interresentanalyse 8

2.5 State of the art 9

2.5.1 Docendo 9

2.5.2 Lantiv 10

2.5.3 Tabulex 11

2.5.4 Untis 12

2.6 Genetiske algoritmer 13

2.6.1 Brute Force 14

2.6.2 Genetiske algoritmer 14

2.6.3 Searchspace 16

3 Delkonklusion **16**

4 Problemafgrænsing **17**

5 Problemformulering **17**

6 Designovervejelser **18**

6.1 Selektion 18

6.1.1 Roulette metoden 18

6.1.2 Rank metoden 18

6.1.3 Tournament metoden 18

6.2 Produktafgrænsning 19

6.3 Kravspecifikationer 20

6.4 Selektion 20

7 Programdokumentation **21**

7.1 Structs 21

7.2 main 22

7.3 Dataindlæsning 24

7.4 Fitness 25

7.5 Selektion 26

7.6 Mutation 27

7.7 Print funktion 28

8 Program test **29**

9 Videreudvikling **30**
9.1 Mangler i løsningen 30
9.2 Graphic user interface 30
9.3 Brugertest 31
9.4 Empiri fra flere skoler 31

10 Diskussion **32**

11 Konklusion **33**

1 Indledning

Skoleskemaet giver struktur til eleverne og lærernes hverdag. Hvis et godt skoleskema bliver planlagt i starten af skoleåret, vil både elever og lærer gavne af det. Et forståeligt og overskueligt skema vil betyde at lærerne kun behøver at tænke på at planlægge deres undervisning, og at eleverne derved får bedre undervisning. I tilfælde af at lærerne synes skemaet er uoverskueligt og ikke passer til deres kriterier, vil det betyde at andre lærer bliver nød til at gå på kompromis og bytte lektioner. Det er derfor en fordel, hvis der bliver lagt et endeligt skema i starten af skoleåret, så der ikke bliver skabt forvirring i løbet af året.[1]

Der er dog mange parametre, der skal tages højde for, når der planlægges skoleskem. Det kan derfor være en vanskelig proces at overskue. Skemaplanlæggerne skal blandt andet tage hensyn til ledige lokaler og gruppearbejde på tværs af parallelle klasser. Der er allerede eksisterende programmer, der kan danne skoleskemaer, men skoler som Sofiendalskolen vælger alligevel at lægge deres i hånden hvert år. Sofiendalskolens proces er langvarig og kostbar, da den kræver at alle 70 lærer på skolen er til stede mens de diskuterer skemaets opbygning. Der kan derfor undersøges, hvilke særparametre skolen stiller til deres skema, siden de mener at de aktuelle software løsninger ikke er tilpas brugervenlige nok eller ikke opfylder deres krav.[2]

Ud fra dette opstilles der et initierende spørgsmål som lyder:

Hvilke parametre tages der højde for når der planlægges skoleskema i folkeskolerne, og hvorledes et program kan hjælpe skolerne i deres skemalægningsproces?

[1] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

[2] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

2 Problemanalyse

I følgende afsnit redegøres for de metoder og valg, der er taget i forhold til problemanalysen. En teknologianalyse i "State of the art"-afsnittet tager udgangspunkt i allerede eksisterende programmer, samt laves en vurdering af potentielle mangler eller problemer med disse programmer.

De konkrete krav, som stilles fra uddannelsesministeriet, er forklaret i et lovgivningsafsnit. Afsnittet vil fremstille de krav, som er nødvendig at tage forhold til, når programmet skal laves. Kravene beskrevet i dette afsnit er nødvendige for forståelsen af opbyggelsen af folkeskoleskemaer. I afsnittet tages der udgangspunkt i de kriterier, som beskrives i "Bekendtgørelse af lov om folkeskolen" med supplerende kilder.

Interessent analysen vurderer de indflydelsesrige og påvirkede medvirkende i forhold til skemalægningen. Afsnittet lægger fokus på vurderingen af interessenternes rolle i skemalægningsprocessen. Analysen udarbejdet i dette afsnit giver grundlag for valget af medvirkende, som er blevet taget kontakt til.

En vital del af problemanalysen er case-study'en, som blev lavet med Søren Kusk fra Sofiendal skole. Interviewet blev semikonstrueret med forberedte spørgsmål, hvorefter der blev stillet uddybende spørgsmål løbende i interviewet. Det var væsentligt at snakke med en, som havde indflydelse på skemalægningen samt forstod og kunne formidle de problemer, som kunne opstå ved denne proces.

Problemanalysens mål er, at forstå skemaets konstruktion samt at finde de mulige problemer, der kan opstå ved selve processen. Afsnittende præsenteret i dette afsnit vil analyseres og udvides med tilstrækkelig information, så en fyldestgørende problemafgrænsning kan konstrueres.

2.1 Skeamets opbygning

I Danmark følger folkeskolerne et fastlagt skema, der giver struktur til eleverne og lærernes dagligdag. Skemaerne for hver folkeskole er unikke, da de er selvstændigt udarbejdet, hvilket betyder det er forskelligt hvilke parametre skolerne prioriter. Prioriteterne kan ændres individuelt mellem skolerne, f.eks. kan nogle skoler prioritere at have flere idræts timer, mens andre prioriterer at have fagene i en hvis rækkefølge. Alle folkeskoleskemaer skal opfylde krav, som er opstillet af regeringen, f.eks. er der et minimumskrav for hvor mange lektioner eleverne på hvert klassetrin skal have i et helt skoleår, se afsnittet "Lovmæssige krav til folkeskoleskemaet".

Kommunerne skolerne ligger i har også indflydelse på skemaplanlægningen. F.eks. har folkeskolerne i Aalborg kommune flere lektioner end folkeskolerne i Rebild kommune.

Skoleskemaet er bygget op således, at eleverne har en række fag hver dag med pauser ind imellem fagene. Lektionerne varer typisk 45 minutter, bemærk at det ikke er det samme som undervisningstimer defineret af uddannelsesministeriet, med pauser på 15 minutter ind imellem lektionerne og en lang middagspause. Nogle skoler vælger dog at afvige fra denne formular ved, f.eks. at have lektioner på 90 minutter med længere pauser ind i mellem. Derudover har lærerne forberedelsestimer, når de ikke underviser. Det vil sige at lærens forberedelsestime potentielt kunne lægge mellem to lektioner, de skal undervise i. Da der er undervisningspligt i 10 år i Danmark, kan folkeskolen være nødsaget til at planlægge 10 til 30 skoleskemaer hvert skoleår.[3] Da der er mange parametre og krav den skemaansvarlige skal tage stilling til, kan skemaplanlægningsprocessen være langvarig.

[3] *Bekendtgørelse af lov om folkeskolen.* 2016. URL: `https://www.retsinformation.dk/forms/r0710.aspx?id=182008`.

2.2 Lovmæssige krav til folkeskoleskemaet

Regeringen stiller en række krav til skolerne under skemaplanlægningsprocessen, der giver rammer, som skemaet bliver nødt til at holde sig indenfor. I "Bekendtgørelsen af lov om folkeskole"' er tre fagblokke beskrevet. De humanistiske fag, praktiske/musiske fag og naturfaglige fag. Det er vigtigt, at produktet kan overholde de lovmæssige krav der stilles. En casestudy vil inddrages senere, som vil bearbejde dette problem yderligere. Her opstilles kravene af de forskellige fag, som eleverne i den 9-årige grundskoleuddannelse skal følge.[4]

Årgang	0	1	2	3	4	5	6	7	8	9	Note
Dansk	X	X	X	X	X	X	X	X	X	X	
Idræt	X	X	X	X	X	X	X	X	X	X	
Matematik	X	X	X	X	X	X	X	X	X	X	
Engelsk		X	X	X	X	X	X	X	X	X	
Historie				X	X	X	X	X	X	X	
Kristendomskundskab	X	X	X	X	X	X	X	X	X	X	Dog ikke på årgangen med konfirmandforberelse
Natur/teknologi		X	X	X	X	X	X				
Billedkunst		X	X	X	X	X					
Musik		X	X	X	X	X	X				
Håndarbejde					X	X	X	X			Skal haves mindst på et af de markede år
Sløjd					X	X	X	X			Skal haves mindst på et af de markede år
Hjemkundskab					X	X	X	X			Skal haves mindst på et af de markede år
Biologi								X	X	X	
Geografi								X	X	X	
Fysik/kemi								X	X	X	
Tysk						X	X	X	X	X	Skal udbydes
Fransk						X	X	X	X	X	Kan udbydes
Valgfag						X	X	X	X	X	Skal vælges

Figure 1: Kravene for hvilke fag den enkelte årgang skal have.

Valgfag består af følgende fag/emner: Tysk, fransk, spansk, medier, billedkunst, filmkundskab, drama, musik, håndarbejde og design, sløjd, madkundskab, motorlære, almindelige indvandrersprog for elever med tilstrækkelig kendskab til det sprog og arbejdskendskab. Valgfag skal være bestående af mindst 120 undervisningstimer årligt.[5]

En undervisningstime er, ifølge førnævnte bekendtgørelse, lig med 60 minutter. Børnehave og fra første til tredje klasse må ikke overskride undervisningstid over seks undervisningstimer om dagen, udover ved særlige arrangementer. Som ses på figur 1 er minimumskravet for første klasse 750 timer på et 200 dags skoleår, hvilket udgør 18,75 undervisningstimer ugentligt.[6]

Medtages undervisningens vejledende timeantal er længden 30 timer ugentligt for en første klasse. For samtlige klasser betyder det følgende antal timer ugentligt:

- Børnehave, første, anden og tredje klasse: 30 timer.
- Fjerde, femte og sjette klasse: 33 undervisningstimer.
- Syvende, ottende og niende klasse: 35 undervisningstimer.

[4] *Bekendtgørelse af lov om folkeskolen.* 2016. URL: https://www.retsinformation.dk/forms/r0710.aspx?id=182008.

[5] *Bekendtgørelse af lov om folkeskolen.* 2016. URL: https://www.retsinformation.dk/forms/r0710.aspx?id=182008.

[6] *Timetal (minimumstimetal og vejledende timetal) for fagene i folkeskolen.* URL: https://www.uvm.dk/-

Timetal (minimumstimetal og vejledende timetal) for fagene i folkeskolen											
Klassetrin	Bh.	1.	2.	3.	4.	5.	6.	7.	8.	9.	Timetal i alt
Humanistiske fag											
Dansk (minimumstimetal)		330	300	270	210	210	210	210	210	210	2.160
Engelsk (vejledende timetal)		30	30	60	60	90	90	90	90	90	630
Tysk eller fransk (vejledende timetal)						30	60	90	90	90	360
Historie (minimumstimetal)				30	60	60	60	60	60	30	360
Kristendomskundskab (vejledende timetal)		60	30	30	30	30	60		30	30	300
Samfundsfag (vejledende timetal)									60	60	120
Naturfag											
Matematik (minimumstimetal)		150	150	150	150	150	150	150	150	150	1.350
Natur/teknik (vejledende timetal)		30	60	60	90	60	60				360
Geografi (vejledende timetal)								60	30	30	120
Biologi (vejledende timetal)								60	60	30	150
Fysik/kemi (vejledende timetal)								60	60	90	210
Praktiske/musiske fag											
Idræt (vejledende timetal)		60	60	60	90	90	90	60	60	60	630
Musik (vejledende timetal)		60	60	60	60	60	30				330
Billedkunst (vejledende timetal)		30	60	60	60	30					240
Håndværk og design samt madkundskab (vejledende timetal)					90	120	120	60			390
Valgfag											
Valgfag (vejledende timetal)								60	60	60	180
Årligt minimumstimetal											
Årligt minimumstimetal pr. klassetrin	600	750	750	780	900	930	930	960	960	930	8.490 inkl. bh.
Undervisningstidens samlede længde											
Undervisningstidens samlede længde (inkl. pauser, understøttende undervisning mv.)	1.200	1.200	1.200	1.200	1.320	1.320	1.320	1.400	1.400	1.400	12.960 inkl. bh.

Figure 2: Skema af kravene af timetal fra folkeskoleklasserne 0.- 9.

Dette stemmer overens med figur 2,[7] dog skal det noteres at for alle fag ekskluderende, dansk

/media/UVM/Filer/Udd/Folke/PDF15/Juli/150710-Timetalsskema_PDF.ashx?la=da.

[7] *Undervisningstimetal i folkeskolen.* Aug. 5, 2016. URL: https://www.uvm.dk/Service/Statistik/Statistik-om-folkeskolen-og-frie-skoler/Statistik-om-elever-i-folkeskolen-og-frie-skoler/Statistik-om-undervisningstimetal-i-folkeskolen?allowCookies=on.

matematik og historie[8] er disse vejledende timeantal. Dog skal de opfylde undervisningstidens samlede længde, der ses nederst på figur 1. Pauser indgår også i denne undervisningstid.[9]

2.2.1 Skolereformen

I 2014 blev den nye skolereform introduceret i folkeskolerne. Reformen opstillede tre klare mål for folkeskolerne:[10]

- Folkeskolen skal udfordre alle elever, så de bliver så dygtige, de kan.
- Folkeskolen skal mindske betydningen af social baggrund for de faglige resultater.
- Tilliden til og trivslen i folkeskolen skal styrkes blandt andet gennem respekt for professionel viden og praksis.

Målene bliver mødt ved længere skoledage, der giver de mindste elever en skoledag, som slutter omkring kl 14, fjerde til sjette klasse er det 14:30 og syvende til niende til 15. Den ekstra skoletid skal bl.a. støtte eleverne i bedre faglig fordybelse ved særligt udfordrende fag ved brug af lektiehjælp. Udover dette bliver flere timer introduceret i form dansk og matematik fra fjerde til niende klasse. Engelsk, andet fremmedsprog og tredje fremmedsprog skal introduceres i henholdsvis første, femte og syvende klasse.[11] Skolereformen har yderligere fokus som hævning af de pædagogiske kompetencer hos lærerne.

2.3 Case study - Sofiendalskolen

For at få et aktuelt indblik i nogle af de problemstillinger, der opstår når en skole lægger et skema, har vi fremstillet et interview med en af skemalæggerne fra Sofiendalskolen.

Sofiendal skolen blev opført i 1911. I dag er Sofiendal skolen en tresporet skole med en speciel adhd klasse, hvilket betyder at der på en årgang befinder sig 3 klasse a, b og c samt speciel klassen. Skolen rummer 70 lærer samt pædagoger, som underviser 650 elever dagligt.[12]

Lørdag den 27 oktober interviewede vi Søren Kusk. Søren Kusk fungerer som matematiklærer samt it-ansvarlig på sofiendalskolen og indgår i et team af 8 lærer, som underviser 3. klasse, hvor to af dem også underviser en anden klasse. På Sofiendalskolen er det ikke en bestemt person, som er ansvarlig for at lægge skemaet, men derimod mødes alle involverede pædagoger og lærer to onsdage i starten af året. Her afholdes to møder med en varighed på tre timer, hvilket vil sige at det tager ca 420 mandetimer for Sofiendalskolen at lægge årets skema. Skemalægningen er et meget simpel koncept men en kompliceret process på Sofiendalskolen. Lærerne og pædagogerne lægger skema uden brug af computerprogrammer. Lærerne starter med at få tildelt hvilke fag, klasser og hvor mange timer, de skal have. Herefter får de hver især farvede brikker, som repræsenterer de timer som lærerne har. brikkerne ligger de på et tomt skema ud fra de klasse, som de skal underviser indtil de ikke har flere brikker.[13]

Når der lægges skema på Sofiendalskolen ønskes det, at lærerne har sammenhængende forberedelses timer, således de ikke er spredt ud over hele ugen, da de mener de ikke får chancen for at forberede sig ordentligt, hvis de kun har en forberedelses time af gangen. Derudover prioriterer

[8] *Den nye folkeskole - en kort guide til reformen.*

[9] *Bekendtgørelse af lov om folkeskolen.* 2016. URL: `https://www.retsinformation.dk/forms/r0710.aspx?id=182008`.

[10] *Den nye folkeskole - en kort guide til reformen.*

[11] *Den nye folkeskole - en kort guide til reformen.*

[12] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

[13] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

de, at eleverne ikke har tunge fag, som f.eks. matematik over middag, da eleverne tit er trætte på daværende tidspunkt, og derfor får begrænset udbytte af undervisningen. Derudover vil lærerne gerne have mulighed for, at lave tværfaglig undervisning på tværs af klasserne, hvilket vil sige at alle tre parallelklasser a, b og c f. eks. skal have mulighed for at have dansk på samme tidspunkt hver mandag. For Skemaplanlæggerne på Sofiendalskolen er det besværligt at opfylde alle disse tre kriterier på en gang. Derfor går de på kompromis med parametrene og vælger hvilke de prioriterer højst. Skemaplanlæggerne bruger på nuværende tidspunkt ikke skemalægningsprogrammer, da de programmer de har afprøvet ikke har kunnet tage højde for flere parametre og præferencer, spredt ud over de forskellige teams og klasser.[14]

Sofiendal skolen har prøvet at lægge skema med programmet Tabulex. Skemaplanlæggerne synes Tabulex var komplekst og avanceret at få sat op for hver klasse, da hver team har forskellige præferencer. Det var derfor svært for Tabulex at lægge skemaet, da Tabulex ikke var fleksibelt nok til deres behov. Derfor har Sofiendalskolen valgt at planlægge skemaer manuelt. Dog vil Sofiendalskolen stadig være åben for at bruge et computerprogram til at lægge skema, hvis programmet kunne hjælpe dem med at gøre processen kortere, uden at gøre det for avanceret at få sat op.[15]

2.4 Interresentanalyse

Der er en række interessenter at tage hensyn til, når det kommer til, hvordan et folkeskoleskema skal planlægges. For at lave et velfungerende skemalægningsprogram til folkeskoler, skal der tages højde for de forskellige interessenter, der bliver påvirket af programmets produkt. I følgende afsnit vil forskellige interessenters påvirkning af en softwareløsning undersøges samt deres indflydelse.

Kommunernes mål er at forbedre elevernes læring, samt overholde undervisningsministeriets krav. Både uddannelsesministeriet og kommunen stiller krav til hvilke fag, som skemaet består af, samt antallet af undervisningstimer, der skal afsættes til de forskellige fag. Kommunernes påvirkning af en softwareløsning ville være minimal, så længe deres opstillede krav overholdes af programmet.[16]

Skolelederen arbejder ud fra et budget, som er tildelt af kommunen, og er derfor interesseret i at optimere mængden af ydelser, dette budget finansierer. En softwareløsning der reducerede timeantallet brugt på skemalægningsprocessen kunne have en stor interesse hos skolelederen, da dette vil tillade budgettet at bruge penge andre steder på skolen, hvor det vil gavne mere. Skolelederen har stor indflydelse på om programmet bliver implementeret, da det er skolelederens beslutning om det er gavnligt at investere i programmet. Det ville også kræve at programmet ikke kræver flere mandetimer end manuel skemalægning, eller producerer en løsning, som ikke er holdbar. Skolelederen har ingen interesse i at investere i en softwareløsning, som skaber flere problemer end den løser. En softwareløsning som producerer et velfungerende skema er for skolelederen vigtigt, da det er skolelederen, der står til ansvar, hvis programmet viser sig at være en dårlig investering.[17]

Lærerne bruger tid og kræfter på skemalægningsprocessen.[18] En softwareløsning vil lette arbejdsbyrden fra lærernes skuldre og ville tillade, at de kan fokusere fuldt ud på undervisningen. Lærerne vil have stor indflydelse på, hvordan en softwareløsning ville komme til at se ud, da det er dem som lægger skemaet. Deres indflydelse påvirker om en softwareløsning vil blive implementeret

[14] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

[15] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

[16] *Bekendtgørelse af lov om folkeskolen.* 2016. URL: `https://www.retsinformation.dk/forms/r0710.aspx?id=182008`.

[17] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

[18] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

på en skole, da softwareløsningen skal kunne opfylde lærernes betingelse for et skoleskema fejlfrit. Da lærerne ikke er interesseret i et program, som skaber flere problemer for dem, end det reelt løser. Lærernes betingelser til en softwareløsning består i at få et optimeret skemaet, så eleverne er fokuseret i undervisningen, og at underviserens forberedelsestimer er samlet. Derudover har Sofiendalskolen fundet ud af, at eleverne har besvær med at koncentrere sig i de tungere fag over middag. Dette giver en præference, hvor de tunge fag placeres før middag. En softwareløsning ville også have en stor påvirkning på lærernes hverdag, da de arbejde ud fra skoleskemaet og problemer, som skoleskemaet skaber, vil have direkte påvirkning på lærerne.[19]

Eleverne har ingen interesse i en softwareløsning og deres indflydelse, på hvordan den endelige løsning kommer til at se ud, er minimal, da de ingen indflydelse har i skemaplanlægning. Eleverne vil dog muligvis blive påvirket af en softwareløsning, skemaet potentielt bliver bedre ved softwareløsningen. Et dårligt planlagt skema vil gøre at de f.eks. ingen energi har til at komme igennem dagen, hvis der bliver lagt tunge fag sidst på dagen.

Forældrenes interesse er baseret ud fra den påvirkning skemaet har for deres børn. Børnenes faglige trivsel i skolen er vigtigt for forældrene, og et optimeret skema vil hjælpe barnet med at få bedst muligt fagligt udbytte af skoledagen. Forældrene har dog ingen påvirkning på, hvordan skemaet bliver lagt og vil derfor ikke opdage, hvis skolen begynder at bruge en softwareløsning til at planlægge deres børns skemaer.[20]

For at overskueliggøre de forskellige interessenter og deres indflydelse på, hvordan skemaet bliver bygget, og hvilken påvirkning skemaet vil have på dem, kan de forskellige interessenter opstilles i et skema som ses på figur 3.

2.5 State of the art

Skemalægningen har længe været et problem, som diverse skoler har haft svært ved at løse, heriblandt er Sofiendalskolen, som vi har interviewet. Selvom der findes mange gode skemalægningsprogrammer er det ikke nødvendigvis ensbetydende med at alle skoler kan benytte disse programmer af den grund, at nogle skoler har svært ved at begrænse sig til så få parametre, som programmerne indeholder. Herudover nævner interviewpersonen, Søren Kusk, at: "[...] selvom det tager højde for mange ting, så er der bare nogle ting som det ikke altid tager højde for."[21] Dette tydeliggør problematikken og pointen i, at skemalægningsprogrammerne ganske enkelt ikke indeholder nok parametre og er præcis nok, til at skoler med forhindringer kan gøre brug af programmerne.

2.5.1 Docendo

Skemalægningsprogrammet Docendo er et brugervenligt samt forholdsvis simpelt program. Programmet går ud på, at der dannes en kalender med uger og dage, hvorefter brugeren har mulighed for at justere diverse parametre alt efter behov. Heriblandt tager programmet bl.a. højde for, at nogle skoler har forskellige fag, og giver derfor brugeren mulighed for at tilføje et eller flere fag. Samtidig har brugeren mulighed for at tilpasse lektionernes længde, hvilket også er en essentiel parameter, da nogle skoler forsøger på at undgå tunge fag om eftermiddagen eksempelvis. Dernæst fastlåser programmet lokaler og lærere som har undervisning på bestemte tidspunkter, så brugeren fortsat har overblik over skemaplanlægningen og så der ikke opstår dobbeltbookninger af et bestemt lokale eller lignende. Hvis et problem skulle opstå, kan lektionerne flyttes med et

[19] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.
[20] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.
[21] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

Påvirkning / indflydelse	Lille indflydelse	Stor indflydelse
Stor påvirkning	Elever Eleverne bliver påvirket meget af skemaet, men har meget lidt påvirkning.	Lærer Det er lærerne der skal undervise i de timer der bliver lagt. Det er også lærerne der skal lægge skoleskemaet. Derudover har flere lærer præferencer til måden hvorpå timerne vil komme til at lægge. Blandt andet deres egne forberedelse timer.
Minimal påvirkning	Forældre Forældrene er interesseret i, at deres børn bliver undervist og får bedst muligt ud af skolen. Dog vil de ikke opleve en stor påvirkning på ændring af skemaet. Kommunen Kommunen har visse krav der skal overholdes i forbindelse med antal timer der skal uddeles, dog er den ikke med i processen.	Skoleleder Det er skolelederens ansvar at skemaet fungerer, og at det er så omkostningsfrit som muligt at få det produceret. En ændring af elevernes skema, vil ikke have en stor forskel på skolelederens hverdag.

Figure 3: Model over de indblandede interessenter.

simpelt klik, og de nye skemaer bliver genereret i ét, hvilket igen gør at der er fortsat overblik over skemalægningen.[22]

2.5.2 Lantiv

Lantiv Timetabling Turbo 7 er et planlægningsprogram der med hensyn til nogle begrænsninger angivet af brugeren, kan udvikle et skema til blandt andet folkeskoler. For hver begrænsning, kan brugeren angive en minimum, maksimum og ønsket værdi. En variation fra den ønskede værdi, bliver af programmet registreret som en mindre overtrædelse, mens en værdi der ligger under minimumsværdien eller over maksimumsværdien, bliver registreret som en alvorlig overtrædelse. Programmet starter med at afsætte kort tid til løsning af overtrædelser, hvorefter den afsatte tid

[22] *Docendo*. Sept. 2014. URL: `https://www.youtube.com/watch?v=F-heCUy4bZ4`.

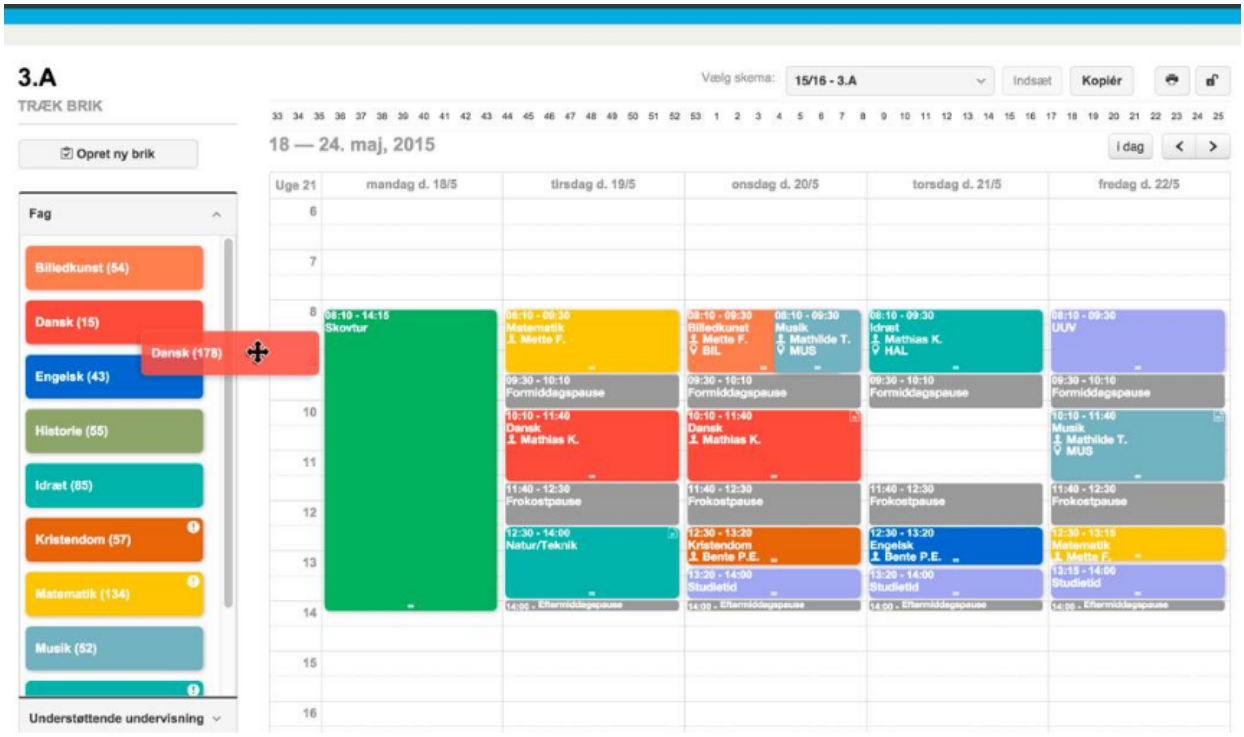


Figure 4: Eksempel på skemplanlægning i Docendo.[23]

stiger for at løse de sværere overtrædelser. Denne proces stopper, når programmet enten har løst alle overtrædelser, eller den maksimale afsatte tid er nået. Hvis overtrædelser af brugerens begrænsninger ikke kan undgås, bliver der lavet et kompromis, hvor programmet hovedsageligt forsøger, at overholde de begrænsninger, brugeren har angivet med høj prioritet, mens overtrædelser af begrænsninger med lav prioritet bliver accepteret. Begrænsningerne kan tilpasses af brugeren, efter skemaet er genereret, og programmet vil levere nogle tilpassede løsninger som forslag. Det oprindelige skema vil kun blive slettet, hvis en af disse løsninger, accepteres af brugeren. Under processen kan brugeren bestemme, hvor meget de tilpassede skemaer må variere fra det oprindelige. Der kan f.eks. stilles et krav, om at programmet kun ændrer lektionerne for en enkelt lærer. I dette tilfælde, vil ingen af de tilpassede forslag, have ændret i andre dele af skemaet. Når skemaet er genereret, er det også muligt for brugeren selv, at tage fat i en lektion og flytte den. Her vil programmet vise, hvor lektionen kan placeres uden at forårsage dobbeltbookninger af lokaler, lærere eller klasser. Hvis brugeren placerer en lektion der forårsager en konflikt, bliver problemet forklaret i detaljer af programmet, og hvis det er en dobbeltbookning, er der muligheden for at slette en af lektionerne eller accepterer dobbeltbookningen.[24]

2.5.3 Tabulex

Skemalægningsprogrammerne er dog ikke problemfri. Selvom programmet opfylder de mest væsentlige krav omkring, hvorvidt et skema bør lægges for at få optimalt udbytte, er programmet for upræcist i forhold til hvilke parametre der tages stilling til, og hvilken af parametre prioriteres højest. Typisk vil sådan et program virke for en skole, hvor lærere ikke har problemer med hensyn til opdeling i teams mm., men dette er ikke tilfældet nogle steder. Heriblandt er Sofiendalskolen, som er af skolerne, hvor lærernes teams ikke fungerer optimalt på grund af, nogle af lærerne er

[24] URL: `http://timetablingturbo.com/advantages.html#1clicktimetabling`.

Figure 5: Eksempel på skemplanlægning i Lantiv.[25]

medlemmer i flere teams. Dette er en essentiel parameter som der ikke tages højde for i skemalægningsprogrammerne, som forårsager forringet udbytte af programmet og i værste fald en helt anden alternativ metode at lægge skemaet. Dog er der samtidig andre faktorer, som gør det en hel del svære at benytte skemalægningsprogrammerne, da programmerne ellers skal være skræddersyet for en specifik skole, før det kan lade sig gøre. Derfor vil en mulig forbedring af de nuværende skemalægningsprogrammer være at tage stilling til mindre faktorer. Programmet skal derfor ikke kunne udlevere et endeligt skema, men til gengæld skal det kunne give en klar formular eller en retningslinje, som skolen herefter kan følge og tilpasse, alt afhængigt af hvilke parametre og faktorer skolen prioriterer højest.[26]

2.5.4 Untis

Untis er et fleksibelt skemalægningsprogram, som både er nemt at bruge, og giver mange muligheder for brugeren. Untis giver bl.a. mulighed for at brugeren kan bestemme, hvordan skemaet

[26] *tabulex 2015 SKEMALÆGNING*. Tabulex. 2015. URL: `http://docplayer.dk/3713728-Tabulex-2015-skemalaegning.html`.

skal se ud uge for uge. Brugeren kan vælge om ugerne skal være A- og B-uger, altså eksempelvis, hvis der gennemsnitligt skal være 5 matematiklektioner på en uge, kan brugeren via A- og B-uger-funktionen gør således at der kommer 4 matematiklektioner i en uge, og 6 i en anden. Desuden har brugeren mulighed for at ændre skemaet, hvis nødvendigt i udvalgte uger, og bibeholde de resterende uger som faste og ensartede. Untis giver brugeren flere muligheder for at lægge skema, heriblandt manuel skemalægning, automatisk skemalægning samt optimering og en blanding af de nævnte. Dette sker ved at brugeren udfylder udvalgte brikker, og herefter benytter Untis til at udfylde resten for at opnå det bedste mulige resultat. Efter de manuelle indtastninger er udført, tager programmet stilling til de angivne parametre og forsøger at levere det bedste skema. Hvis der opstår konflikter, vil Untis informere brugeren og og vise hvori problemet ligger, så brugeren har nemt ved at rette fejlene og justere parametrene. Desuden tager programmet hensyn til indtastede data før programmet begynder at generere mulige skemaer, for at sikre at brugeren har indtastet korrekte eller realistiske antal af eksempelvis lokaler, lærer, klasser mm.[27]

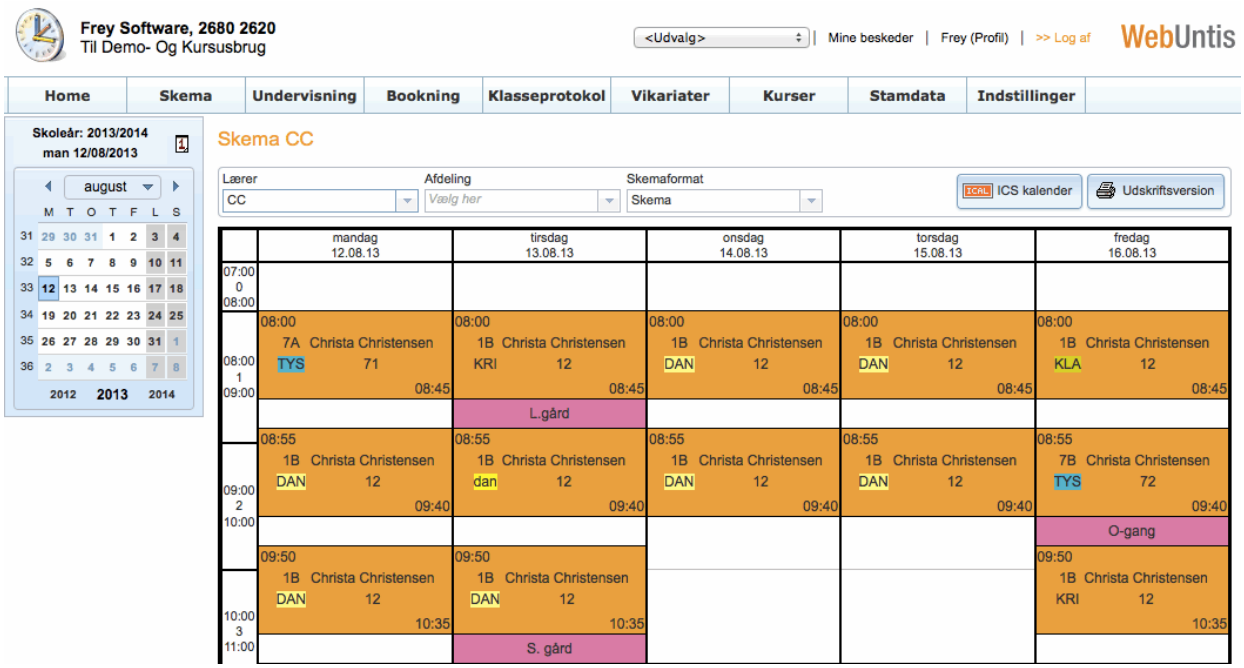


Figure 6: Eksempel på skemplanlægning i Untis.[28]

2.6 Genetiske algoritmer

Et skoleskema består af grundlæggende elementer. Som eksempel kan lektioner være et sådanne element, mens tidspunktet for lektionen kan være et andet. Disse grundlæggende elementer er forholdsvist nemme at beskrive, da de er konkrete, og som sagt før, gundlæggende for skoleskemaets opbygning. Der er også mere abstrakte elemtenter i et skoleskema, som hvornår hvilke fag skal lægge tidsmæssigt, både i forhold til hinanden og i forhold til tiden. Det er disse elementer, som kræver mere individuel input, hvis de skal opfylde de krav, som den enkelte bruger skal bruge.

Genetiske algoritmer giver derfor mulighed for let at generere et skoleskema. Som nævnt tidligere bliver mange generationer genereret ud fra parametrer og deres vigtighed overfor 'fitness'-værdien. Dette betyder, at specifikke parametrer kan tages højde for, og det giver større mulighed

[27] URL: `http://untis.dk/`.

for fleksibilitet uden at kræve en omvæltning at selve programmet.

Det er dette koncept, som bærer grundlaget for brugen af genetiske algoritmer til genereringen af skoleskeamer, da programmet helt naturligt speciferer sit resultat til at tilfredsstille de krav, som bliver stillet af brugeren. Dette, samlet med princippet om at kravenes vigtig kan justeres nemt, giver stærkt grundlag for brugen af genetiske algoritmer.

For at lave algoritmen til at planlægge et skema, er der forskellige metoder. Hver metode har fordele og ulemper og er derfor egnet til forskellige problemer.

2.6.1 Brute Force

Brute-force algoritmen går i sin simple form ud på, at tjekke alle mulige kombinationer af problemet, og ligesom i den genetiske algoritme, beregne en fitness for hver mulig løsning. Da alle mulige løsninger bliver afprøvet, er det sikkert, at den bedste løsning bliver fundet i forhold til den valgte fitness.

Hvis der f.eks. skal findes en divisor for et heltal n, ville brute-force metoden gå ud på at gå gennem alle tal mellem 1 og n, og tjekke om tallet kan divideres uden, at producere en rest.

Selvom denne metode er sikker på den optimale løsning, vil et problem med mange parametre kræve længere tid at køre, sammenlignet med andre metoder som genetisk algoritme, da vi i genetisk algoritme hurtigt sorterer mange løsninger fra, uden at gå igennem dem enkeltvis.

Der er visse applikationer hvor brute-force kan være meget brugbart, hvis man F.eks. vil cracke et pass-word. Når man skal cracke et password kan man ikke bruge genetiske algoritmer, for du får ikke et output der fortæller om ens løsning er tæt på at være rigtig. Med brute-force er det sikkert, at man får den rigtige løsning, da brute-force tester alle løsninger.[29]

2.6.2 Genetiske algoritmer

Genetiske algoritmer er en metode til at optimere en løsning til et givent problem. Genetiske algoritmer er en række retningslinjer som udefra disse danner flere løsninger til et problem og finde den bedste løsning til problemet.

Genetiske algoritmer er en metode, som er inspireret af Charles Darwins teori om naturlig selektion, som omhandler hvordan de biologiske arter udvikler sig over tid, ved at tilpasse sig miljøet og derved bliver bedre egnet til, at overleve og formere sig i miljøet. Teorien danner grundlag i, at populationen af en given art har forskellige kromosomer og at der over tid vil ske små ændringer i kromosomerne hos individerne. Disse små ændringer i kromosomerne vil over længere tid, føre til større ændringer hos individerne. På daværende tidspunkt var Darwins teori meget kontroversiel, dette er dog ikke tilfældet for softwarebrug af genetiske algoritmer, da en algoritme er noget lettere at forklare, end en biologisk ændring i en art over længere tid. En genetisk algoritme består af følgende dele.[30]

1. Individer som er mulige løsninger til problemet.
2. Fitness som er en egenskab hos individerne, som bliver udregnet i forhold til, hvor god en løsning individerne er til problemet.
3. Et individ består af flere kromosomer.
4. En population som består af en mængde af individer.

[29] *Brute force cracking.* 2016. URL: `http://searchsecurity.techtarget.com/definition/brute-force-cracking`.

[30] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

5. Generationer som indikere hvor lang tid algoritmer forløber over.

En population bliver muteret ved følgende operationer:

Selektion Operatoren vælger individer til reproduktionen, jo større fitness individerne har, jo større sandsynlighed er der for, at de bliver valgt til reproduktion. Reproduktion kombiner 2 individer og danner ud fra deres kromosomer et nyt individ.[31]

Crossover Operatoren vælger 2 tilfældige kromosomer og blander dem så der bliver dannet 2 nye kromosomer som er en kombination af de 2 kromosomer fx strengene 1110100 1011111 bliver krydset og danner de 2 nye strenge 1010100 1111111.[32]

Mutation Operationen flipper tilfældigt nogle stykker af kromosomet fx strengen 1001000 bliver muteret i dens tredje position til 1011000. Mutation kan ske i alle positioner i strengen med en hvis sandsynlighed. Mutationsprocessen er vigtig for den genetiske algoritme, da man ved at muterer kromosomet udforsker flere muglie løsninger til det givende problem.[33]

Mangfoldighed Når den optimale løsning skal findes, er det vigtigt at algoritmen ikke bliver fanget på et lokalt højde-punkt, men at man kan komme over disse til det globale højdepunkt. Eksempelvis, hvis et problem havde 5 parametre, og man udregnede fitness heraf. Heri vil der forekomme selektion, crossover og mutation. Dette betyder at endnu en fitness kan udregnes for den givne population. Men hvad nu, hvis algoritmen hænger fast.

Hvis parametrene kun ændres en lille smule for hver generation, gør at programmet nu hænger fast, da tallene omkring disse parametre ikke gør at der kommer en højere fitness. Ud fra parametrene muligheden er det mligt at udregne den højeste fitness alt afhængigt af, hvor meget parametrene skal justeres. Minimale ændringer vil ikke nødvendigvis resultere i den højeste fitness, hvorimod store ændringer i parametrene vil muligvis udregne det højeste fitness. Programmet skal i starten gerne finde løsninger, som lægger spredt ud over hele problemet, og så derefter fjerne denne mangfoldighed igen for at 'zoome' ind på den endelige optimale løsning. Det er gavnligt at have er et individ med, ikke bare stor fitness, men et som heller ikke ligner de andre. På figur XXses, at der forsøges at finde det individ, der ligger på linjerne.

Både ved brug af brute force og generisk algoritme er målet at finde den optimale løsning. Denne beregnes ud fra et fitness niveau, hvor der på hver løsning laves en udregning ud fra valgte parametre. Denne returnerer et fitnessniveau. Herefter gemmes løsningen med det bedste fitnessniveau. Det bedste fitnessniveau kan være defineret ved den højest mulige eller lavest mulige værdi.[34]

løsning til problemet.

1. Der bliver tilfældet generet en population af n-individer med l-kromosomer.

2. De genetiske algoritme operatorer udsætter populationen for mutation og derved vil der fore-komme ændringer hos kromosomerne i individerne eller danne nye individer.

3. Der bliver beregnet fitness for hvert n-individ i populationen. Fitness bestemmer sandsynligheden for individet overlever i populationen.

[31] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

[32] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

[33] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

[34] Patrick Winston. *13. Learning: Genetic Algorithms.* 2014. URL: `https://www.youtube.com/watch?v=kHyNqSnzP8Y`.

4. Processen gentages fra trin 2 i x antal givende generationer eller indtil et givet kriterium er opfyldt.

Algoritmen er færdig når x antal generationer er kørt igennem eller et givet kriterium er opfyldt og der vil være en population med potentielle løsninger til problemet, hvor der herudfra vil blive valgt en løsning til problemet som typisk er løsningen med højst fitness. Genetiske algoritmer kan bruges til at optimere problemet med skemalægning, udefra en række retningslinjer som bliver bestemt. Det vil udmunde i en optimal løsning til skemalægningsproblemet, hvis der bliver givet de korrekte retningslinjer.[35]

2.6.3 Searchspace

"Seach space" referer til en gruppe af kandidat løsning til et problem, hvor der er en "distance" i mellem kandidaterne. For eksempel lad os tage vigtigt problem indenfor bioengeering, hvordan man designer et protein. Antaget at man vil søge efter et protein som er en sekvens af aminosyre som kan blive brugt til at bekæmpe en virus. "Seach space" vil være en kollektion af alle mulige proteiner. Dette vil give os uendelige mange muligheder derfor begrænser vi længden af proteinet til længden 50 som stadig vil være et stort "seach space" siden der er 20 mulige aminosyre i hver position i proteinet. Hvis vi repræsenter aminosyrerne i form af alfabetet vil et muligt protein se ledes ud. ASDKEGHB.... Vi definer distance mellem proteinerne som forskellen i alfabetet på den tilsvarende position i et andet protein fx ASDKEGHB og BSDKEGHB er distance 1 og distance mellem ASDKEGHB og GCCHAKAA er 8.[36]

3 Delkonklusion

Problemanalysen har givet et overblik over de forskellige aspekter af skemalægningen. Det er tydeligt, at variablerne som indgår i processen skal overvejes for at et godt produkt kan laves. De lovmæssige krav til skoleskemaet er en nødvendig del at konkretisere, da de sætter rammerne produktet skal arbejde indenfor. Interviewet gav indblik i, hvordan skemalæggerne vurderede processen, hvilke hensyn og præferencer der tages i brug, samt hvilke problemer der opstår. De øvrige afsnit vurderer, hvordan processen allerede er forsøgt behandlet, state of the art, samt hvilke interessenter der er medvirkende og påvirker processen.

Afsnittet viser, at problemer opstår i selve processen. I Sofiendalskolens situation er det klart, at det er mange variabler der indgår. At være i stand til at vægte disse variabler mod hinanden giver dog en hvis kompleksitet. Det er derfor vigtigt at forstå denne kompleksitet i processen. Skulle programmet kunne løse alle problemerne, samt stadig overholde de lovmæssige krav, ville projektet hurtigt blive uoverskueligt. En afgrænsning af projektets fokus redegøres for i følgende afsnit for at indsnævre fokusset til en konkret og håndterbar problemstilling.

[35] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).
[36] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

4 Problemafgrænsing

Ved vores interview med Søren Kusk fra Sofiendalskolen, har vi fået information om, hvilke krav de stiller, når der ligges skema på Sofiendalskolen. Samt hvilke problemer der opstår under skolens skemalægning. Derudover er det blevet undersøgt hvorfor der bruges manuel skemalægning fremfor at bruge en af et allerede eksisterende softwareløsninger på markedet.[37]

Ud fra vores empiri fra interviewet samt egen research af state of art, har vi fundet ud af, at de eksisterende skemaplanlægningsprogrammerne på markedet har svært ved at tage hensyn til folkeskoleskema, sådan lærere har tidsmæssigt samlede forberedelsestimer.

På sofiendalskolen føler de at det vil være nødvendigt, at indgå kompromiser, og det er svært at standardiserer vigtigheden af de enkelte parametre der skal tages højde for i skemaplanlægningsprocessen. Derudover er det en tidskrævende proces for lærerne, at sætte sig ind og få forståelse for et af de allerede eksisterende skemaplanlægningsprogram.

Ud fra denne viden, har vi besluttet at lave et program der kan lave et grundskema, der stemmer overens med de lovgivningsmæssige krav for elevernes timetal. Vores skema skal tage højde for at hver klasse og lærer ikke kan have mere end en lektion af gangen, samt et begrænset antal faglokaler. Vi vil ved brug af generisk algoritme lave et fit niveau der tilpasses af, i hvor høj grad det lykkedes at lægge lærernes forberedelsestimer i forlængelse af hinanden, så de har 2-3 timers forberedelse af gangen, fremfor mange små forberedelsestimer, da dette vægter højt for Sofiendalskolen.[38] Så vores program vil løse en del af det tidsmæssige problem for lærerne samt problemet med fordelte forberedelsestimer.

5 Problemformulering

Hvordan kan problemerne relateret til skemalægningsprocessen, løses ved hjælp af et program der automatisere processen ved brug af genetiske algoritmer. Hvilke parametre skal der tages højde for i programmet og hvordan skal de rangeres i programmet?

[37] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.
[38] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

6 Designovervejelser

6.1 Selektion

Som tidligere diskuteret er processen for genetiske algoritmer følgende:

- Forældre individerne bliver valgt.
- Forældrene parres.
- Populationen vokser.
- Individerne bliver muteret, eller krydset.
- Sandsynligheden for at en crossover eller mutation finder sted bliver bestemt ud fra en selektionsmetode.

I det følgende afsnit beskrives nogle af de selektionsmetoder der kan bruges. Det vil endvidere også blive diskuteret, hvilken af disse metoder der bedst kan anvendes til at producere skoleskemaer.[39]

6.1.1 Roulette metoden

Man kan forestille sig at hvert individ er tildelt et stykke på en roulette og størrelsen på stykket er proportional med individets fitness. Roulette bliver spinnet n antal gange, det vil tage for at vælge forældrene til den næste generation. Under hvert spin bliver individet under roulettens markør valgt til, at være en del af en gruppe af forældre til den næste generation. En kandidat kan godt blive valgt til, at være forældre flere gange, da det er forældrene til næste generation og ikke selve individerne i generationen, der bliver valgt, gør det dog ingen forskel. Formålet med denne metode er, at få valgt de forældre med den største fitness til næste generation, da de har større sandsynlig for, at skabe individer med større fitness. Problemet med denne metode er dog, at den genetiske algoritme hurtigt vil stå fast i den ene del af fitness rummet, da det er muligt at vælge den samme forældre flere gange, og derved kan der blive skabt en meget ensartet population, som gør at der kun vil blive udforsket et bestemt område af rummet i stedet for at udforske hele rummet.[40][41] Roulette metoden kan illustreres på følgende måde:

6.1.2 Rank metoden

I rang metoden bliver individerne sorteret efter størrelsen på deres fitness. Derefter bliver individerne tildelt lodder efter den plads de har fået efter sorteringen. Det vil sige at den med den laveste fitness får tildelt et lod, den med den anden mindste får tildelt to lodder osv. Antallet af lodder et individ er blevet tildelt forstørre chancen for at individet bliver valgt som forældre til en fremtidig generation. I modsætning til roulette metoden, har fitnessen altså i rang metoden en indirekte påvirkning på individernes chance for at bliver valgt.[43]

6.1.3 Tournament metoden

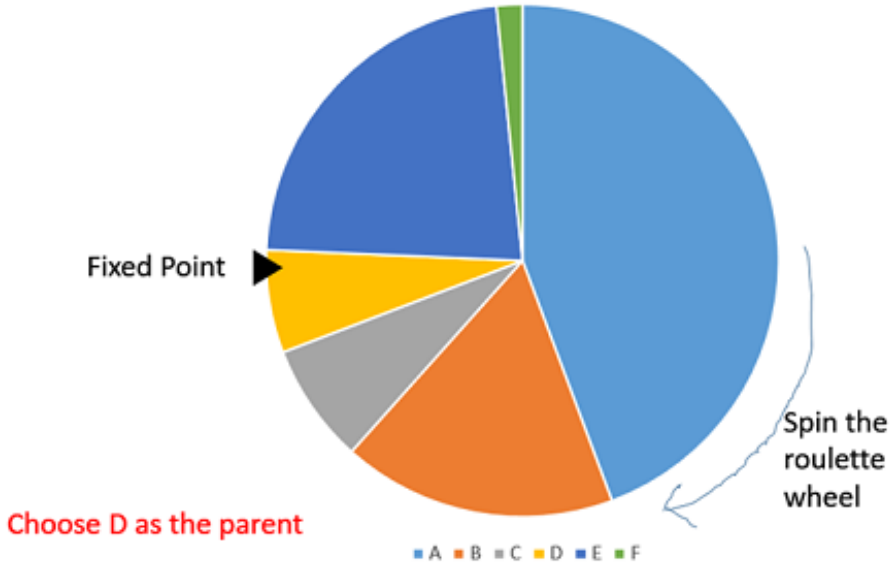
2 tilfældige individer bliver valgt fra populationen. Man generer en tilfældig værdi fra 0-1 for at sammenligne den med valgte sandsynlighedsværdi. Hvis værdien er mindre eller lige med

[39] Patrick Winston. *13. Learning: Genetic Algorithms*. 2014. URL: `https://www.youtube.com/watch?v=kHyNqSnzP8Y`.

[40] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

[41] *Selection*. URL: `http://www.obitko.com/tutorials/genetic-algorithms/selection.php`.

[43] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Figure 7: Model over roulette metoden[42].

sandsynlighedsværdien bliver det individ med højst fitness valgt ellers bliver individet med den lavere fitness valgt. Sandsynlighedsværdien bliver altid sat højere end 0.5 for at favorisere individet med den højeste fitness[44].

I det følgende program benyttes roulette metoden. Ved brug af roulette metoden sikres det at der er større sandsynlighed for at individerne med den højeste fitness bliver brugt til at skabe de fremtidige generationer.

6.2 Produktafgrænsning

Ved vores interview med Søren Kusk fra Sofiendalskolen, har vi fået information om, hvilke krav de stiller, når der ligges skema på Sofiendalskolen. Samt hvilke problemer der opstår under skolens skemalægning. Derudover er det blevet undersøgt hvorfor der bruges manuel skemalægning fremfor at bruge en af et allerede eksisterende softwareløsninger på markedet.[45]

Ud fra vores empiri fra interviewet samt egen research af state of art, har vi fundet ud af, at de eksisterende skemaplanlægningsprogrammerne på markedet har svært ved at tage hensyn til folkeskoleskema, sådan lærere har tidsmæssigt samlede forberedelsestimer.

På sofiendalskolen føler de at det vil være nødvendigt, at indgå kompromiser, og det er svært at standardiserer vigtigheden af de enkelte parametre der skal tages højde for i skemaplanlægningsprocessen. Derudover er det en tidskrævende proces for lærerne, at sætte sig ind og få forståelse for et af de allerede eksisterende skemaplanlægningsprogram.

Ud fra denne viden, har vi besluttet at lave et program der kan lave et grundskema, der stemmer overens med de lovgivningsmæssige krav for elevernes timetal. Vores skema skal tage højde for at hver klasse og lærer ikke kan have mere end en lektion af gangen, samt et begrænset antal faglokaler. Vi vil ved brug af generisk algoritme lave et fit niveau der tilpasses af, i hvor høj grad det lykkedes at lægge lærernes forberedelsestimer i forlængelse af hinanden, så de har 2-3 timers forberedelse af gangen, fremfor mange små forberedelsestimer, da dette vægter højt for

[44]Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).
[45]swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

Sofiendalskolen.[46] Så vores program vil løse en del af det tidsmæssige problem for lærerne samt problemet med fordelte forberedelsestimer.

6.3 Kravspecifikationer

Hvilke krav og bindinger skal vægtes i programmet:

- Programmet skal generere skemaer for 9 klasser, 3 skemaer for henholdsvis 7, 8 og 9 klassetrin.
- Programmet skal overholde minimumskravene for timetal i folkeskolen.
- Programmet skal tage højde for at klassetrinene ikke har det samme antal fag eller samme type fag.
- Det skal ikke være muligt for en lærer at have lektioner i to klasser på en gang.
- Lærerne skal have mere end en forberedelsestime adgangen.
- Der må ikke være tomme lektioner i midten eller starten af skemaet.
- Samarbejde på tværs af parallel klasserne skal være muligt.
- Programmet skal læse lærernes initialer ind via en fil. Filen skal simulere en indstillingsmenu for forbrugeren.
- Der må højest være 8 lektioner på en dag.
- Lektionerne skal helst være ligeligt fordelt over alle ugedagene, således at der ikke er 3 dage med 8 lektioner og 2 dage med 4 lektioner.

6.4 Selektion

I programmet anvendes en række defines i toppen, hvor værdierne kan ændres efter behov. Så er de forskellige skolefag skrevet som en enumeration type, efterfulgt af definitioner på de anvendte structs. Herefter ligger alle anvendte funktioner som prototyper. Selve funktionerne ligger i bunden. Kommentarer skrives over det stykke kode den forklarer. Når der udføres en algoritme i en funktion eller en løkke, er algoritmen rykket ind med et tab af to mellemrum. Tuborgklammer starter inden linjeskift, efter funktioner og løkker, og afsluttes efter et linjeskift.

[46] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

7 Programdokumentation

For at løse problemerne relateret til skemalægningsprocessen er der blevet lavet en softwareløsning i form af et c-program. Det følgende afsnit vil forklare hoveddelene af programmet, samt de vigtigste komponenter. Derudover vil der også være diagrammer der forklarer, hvad der kommer ind og ud af funktionerne, og i hvilken rækkefølge funktionerne bliver kaldt i.

7.1 Structs

Programmet bruger structs, så strukturen af dataene, der produceres, er lettere at håndtere. Fire structs, "lesson", "individual", "teacher" og "requirements" er brugt til henholdsvis de faglige krav, lektionernes -, skemaernes - og lærernes information.

```
struct lesson{
  char teacher_name[TEACHER_NAME_MAX];
  char lesson_name[LESSON_NAME_MAX];
  int number_of_lessons;
  char class_name[TEACHER_NAME_MAX];
};
```

Structen, "lesson", består af to variabler, et navn på læren og et navn på lektionen. Denne struct bruges ved printning af skemaet.

```
struct individual{
  int lesson_num[LESSONS_PER_DAY_MAX][SCHOOL_DAYS_IN_WEEK];
  int fitness;
  int grade;
  int perfection;
  int lessons_with_parallel;
  int lessons_with_both;
  int heavy_lesson_after;
};
```

"Individual" består af et array af arrays af integers, lesson_num, der repræsenterer lektionnummeret (faget) for hver blok på hver dag.

og to integers, fitness og grade. Array'et har to indgange, hvor det første identificerer, hvilken lektion skal tilgås og den anden tilgår ugedagen. Hver indgang i skemaet er givet ved et heltal, som korrelerer med et fag. Fitness er en værdi, der udregnes ud fra, hvor godt skemaet opfylder de krav, der stilles af skemalæggeren. Det er denne værdi, som bestemmer skemaets bæredygtighed i forhold til andre skemaer. Grade er et heltal, som bestemmer hvilken klasse, der er tale om. Siden forskellige krav stilles til forskellige klassetrin, er det vigtigt for programmet, at være i stand til at skelne mellem klassentrinnene.

```
struct teacher{
  char teacher_name[TEACHER_NAME_MAX];
  char lesson_name[LESSON_NAME_MAX];
  int number_of_lessons;
  char class_name[TEACHER_NAME_MAX];
};
```

"teacher" structen indeholder navnet på læren, navnet på faget, antallet af lektioner læren har for det givne fag og klassen læren har til faget. Denne struct skal forstås sådan, at en klasse med et specifikt fag har et specifikt antal undervisningstimer med en specifik lære. Som eksempel, 7.a kunne have 4 timers matematik med læren "Jørgen".

```
struct requirements{
  int Dan_req;
  int Mat_req;
  int Eng_req;
  int Tys_req;
  int Fys_req;
  int His_req;
  int Sam_req;
  int Val_req;
  int Geo_req;
  int Bio_req;
  int Gym_req;
  int Fri_req;
  int Rel_req;
  int Pra_req;
};
```

"requirements" er en struct udelukkende bestående af heltal. Disse heltal viser kravene til de forskellige fag. Det er disse krav, som programmet bl.a. bruger til at tjekke om skemaet er opfyldt, eller om der mangler et specifikt fag, som skal fyldes ind i skemaet.

7.2 main

Følgende kode er main funktionen, der bliver brugt til at danne et skema for udskolingssektorene på en vilkårlig skole. Under koden findes et diagram der kan bruges til at danne overblik over main funktionen, funktionerne der bliver kaldt i main funktionen og hvad de returnere til main funktionen.

```
int main(void){

  int h_classes[NUMBER_OF_HEAVY_LESSONS] = {mat, fys, eng, dan, tys};

  individual individuals[NUMBER_OF_CLASSES][NUMBER_OF_INDIVIDUALS];
  individual chosen_individual[NUMBER_OF_CLASSES][NUMBER_OF_GENERATIONS];
  individual individuals_temp[NUMBER_OF_CLASSES][NUMBER_OF_INDIVIDUALS];
  lesson week[NUMBER_OF_CLASSES][SCHOOL_DAYS_IN_WEEK * LESSONS_PER_DAY_MAX];
  requirements requirements_classes[NUMBER_OF_CLASSES];

  int number_of_teacher = find_number_of_teachers();
  teacher teacher_data[NUMBER_OF_CLASSES][NUMBER_OF_FAG];
  read_teachers_name(teacher_data, number_of_teacher);
  find_req(teacher_data, requirements_classes);
```

```
  srand(time(NULL));

  create_individuals(individuals);
  calculate_fitness_all(individuals, h_classes, teacher_data, requirements_classes);

  int i, j;
  for (i = 0; i < NUMBER_OF_GENERATIONS; i++){
    calculate_fitness_all(individuals, h_classes, teacher_data, requirements_classes

    selektion(individuals);

    mutation(individuals);

    crossover(individuals, individuals_temp);

    for(j = 0; j < (NUMBER_OF_CLASSES); j += 3){
      choose_individual(individuals, chosen_individual, j, i);
    }
    printf("%d \t %d \t %d \t %d \t %d \t %d \t %d \t %d \n", chosen_individua
                                                              chosen_individua
                                                              chosen_individua

  }

  for(i = 0; i < NUMBER_OF_CLASSES; i++){
    printf("  The fitness is: %d    8.A's Skoleschedule: \n", chosen_individual[i][NU
    create_schedule(week, chosen_individual[i][NUMBER_OF_GENERATIONS -1], i);
    print_schedule(week, i);

    if (i \% 3 == 0){
      printf("\n\n");
    }
  }

  return 0;
}
```

Først initialiseres de arrays der bliver brugt i programmets funktioner. De arrays der bliver dannet har to dimensioner. Dernæst forberedes der til brugen af den genetiske algoritme. Srand funktionen klargøres ved at sætte time til at være NULL. Individerne bliver givet tilfædige værdier i lesson_num (der får indsat tilfældige lektioner). Herefter udregnes individernes fitness og de bliver sendt ind i for loopet der kører en gang per generation. I loopet udføres en selektion af alle individer ved hjælp af roulette metoden. Herefter er der i hvert skema sandsynlighed for mutationer. Efter mutationen bliver skemaerne lavet om til crossovers mellem to tilfældige forældre. Efter denne process bliver skemaernes fitness igen regnet ud, inden for loopet begynder forfra. Der kan dannes et overblik over main med følgende diagram:

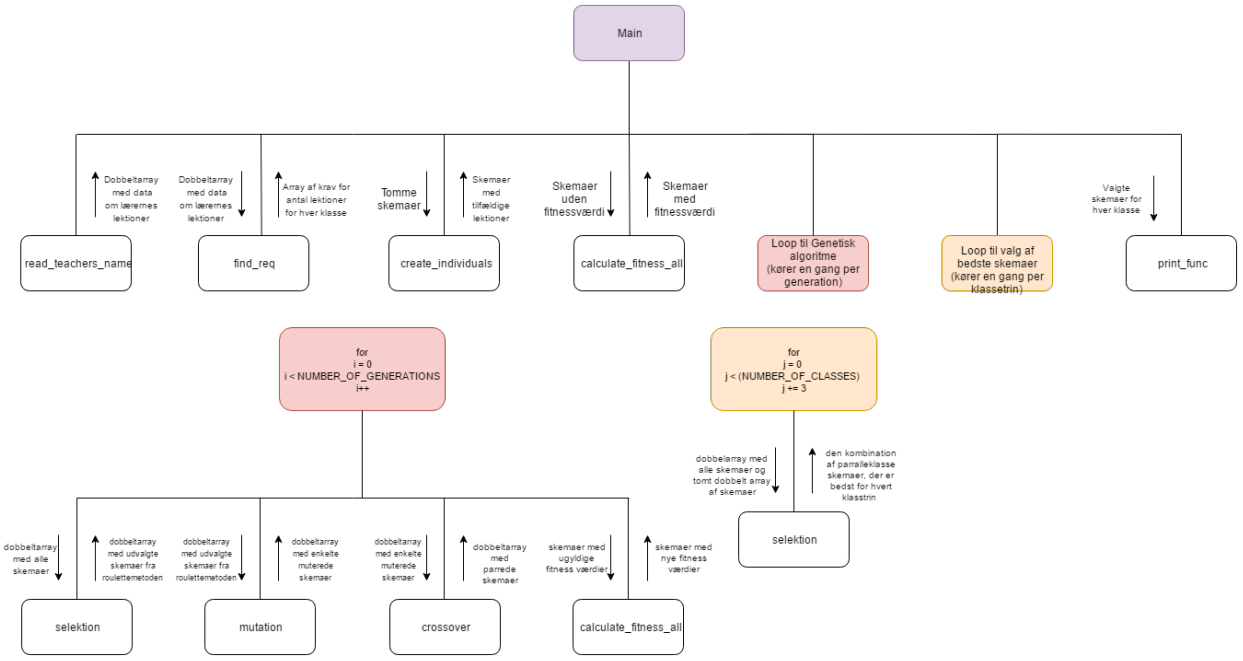


Figure 8: Diagram over main.

7.3 Dataindlæsning

For at nemmere lave ændringer i antallet af lærer, de timer de kan tage og det antal timer de forskellige fag skal have, er der opstillet et tekstdokument hvor dette står. Elementerne i dokumentet står som følgende: 'Lærer forkortelse' 'fag' 'antal timer' 'klasse'

Et eksempel på dette kunne være: CA Dan 6 7a JA Mat 4 7a HA Eng 3 7a RA Tys 3 7a MA Fys 2 7a MA His 2 7a KA Sam 2 7a UA Val 2 7a RA Geo 2 7a Dette bliver indlæst fra filen én gang i starten af programmet, hvorefter data'en fra dokumentet bliver gemt over i en struct. Dette er implementeret på følgende vis:

```
void read_teacher_data(teacher teacher_data[][NUMBER_OF_SUBJECTS]){
 FILE *teacherinfo = fopen("teacherinfo.txt", "r");
  if(teacherinfo == NULL){
    perror("Error the file is empty");
    fclose(teacherinfo);
    exit(1);
  }

  teacher local_teacher_data;
  int i = 0, j = 0;
  for(j = 0; j < NUMBER_OF_CLASSES; j++){
    for(i = 0;i < NUMBER_OF_SUBJECTS; i++){
      fscanf(teacherinfo,
      " %s %s %d %s ",
      local_teacher_data.teacher_name,
      local_teacher_data.lesson_name,
      &local_teacher_data.number_of_lessons,
      local_teacher_data.class_name);
```

```
        teacher_data[j][i] = local_teacher_data;

      }
    }
    fclose(teacherinfo);
}
```

Først åbnes filen, hvorefter det tjekkes om filen er tom. Hvis dette er sandt, bliver der printet en fejl-besked ud, og programmet lukker igen. Dernæst bliver der lavet Local_teacher_data af typen 'teacher', som er et struct. Denne bliver brugt i den efterfølgende løkke, som læser filen linje for linje, hvor den gemmer informationerne for hver linje, ind i local_teacher_data. Efter dette bliver det overført til 'teacher_data' som er det endelige struct, med alt informationen omkring lærerne, deres dag, antal timer og deres klasse. Efter denne løkke bilver filen lukket igen, og alle informationer er kopieret over i 'teacher_data'.

7.4 Fitness

Fitnessen bliver beregnet ved at tage de enkelte individer, samt deres parallelle klasser, og så gå igennem disse enkeltvis, hvor det undersøges om de overholder nogle specifikke krav, eller overtræder nogle andre. Hvis en af disse parametre er sande, får individet så en bonus eller en straf afhængigt af hvilken parametre der går i opfyldelse, og hvor vigtig denne er. Denne straf, eller bonus, bliver så lagt ind på fitness-variablen i det enkelte individ. Hvis individet har en negativ fitness-værdi, bliver dette ændret til 1, da negative værdier ville udgå fra selektion, eller gør processen mere kompliseret. med en fitness værdi på 1 er det stadig meget usandsynligt, men muligt hvergang selektionen kører, at et sådant skema vil blive valgt og påvirke udviklingen. De forskellige krav der bliver tjekket for, som giver bonus, er som følgende:

- Hvis timerne lægger i træk. Hvis der F.eks. ligger to matematiktimer i streg.
- Hvis parallel klasserne har de samme timer på det samme tidspunkt.
- Hvis en lærer har to forberedelses-timer i streg.
- Hvis der er en fri time i bunden af dagen.
- Hvis et skema overholder kravene for antal timer en klasse skal have.

De krav der giver en straf er:

- Hvis der er tunge fag over middag.
- Hvis der er fri midt på dagen.
- Hvis en lærer er booket til flere timer på samme tid.
- Hvis der er for mange af de samme fag i streg.
- Hvis et skema ikke overholder kravene for antallet af timer en klasse skal have.

Måden hvorpå der tjekkes om parallelklasserne har timer på samme tid kan ses herunder:

```
#define SCHOOL_DAYS_IN_WEEK 5
#define LESSONS_PER_DAY_MAX 8
#define FITNESS_PARALEL_CLASS 50

/* Parallel classes - lessons in a sync */
for (j = 0; j < SCHOOL_DAYS_IN_WEEK; j++){
  for(i = 0; i < LESSONS_PER_DAY_MAX; i++){
    if (individual_master->lesson_num[i][j] == individual_parallel1->lesson_num[i][j]
```

```
      individual_master->fitness += FITNESS_PARALEL_CLASS;
    }
    if (individual_master->lesson_num[i][j] == individual_parallel2->lesson_num[i][j]
      individual_master->fitness += FITNESS_PARALEL_CLASS;
    }
  }
}
```

I fitness bruger vi de forskellige variabler 'individual_master', 'individual_parallel1' og 'individual_parallel1'. Disse variabler er af typen 'individual', som er et struct med informationer omkring et skema. Vi starter med at gå gennem to for-løkker, en variabel 'j' bliver talt op antallet af skoledage på en uge, og en variabel 'i' bliver talt op til antallet af lektioner på en dag. Nu ses der på om 'master' har samme time på samme plads, som en af parallelklasserne. Hvis dette er sandt bliver fitness talt op på 'master'. Det samme tjekkes nu om den anden parallelklasse. De andre bindinger bliver tjekket med lignende stykker kode inde i fitnessfunktionen og påvirker fitnessen med en bestemt værdi, skrevet som symbolske konstanter, så programmet er tilpasseligt.

7.5 Selektion

Ved selektion anvendes roulette metoden i programmet. Dette foregår ved, at fitnessen for alle individer i et bestemt klassetrin først lægges sammen for at finde størrelsen på rouletten. Denne gemmes i variablen "sum", som er en integer. Herefter vælges et tilfældigt punkt på rouletten ved hjælp af rand() modolus summen. Punktet gemmes i field integer'en.

Herefter køres en for-løkke der starter fra bunden af rouletten og lægger summen af fitnessen for tre parallelklasser sammen med summen af den forrige sum, hvorefter der tjekkes om punktet ligger under den nye værdi. Hvis dette er tilfældet, bliver de i'ende individer (tre parralelklasser) valgt og gemt ned i temp_individuals.

```
int pick_individual(individual temp_individuals[][NUMBER_OF_INDIVIDUALS], individual
  int i, sum = 0, j;
  int fitness_test = 0;
  int sum_parrallel[NUMBER_OF_INDIVIDUALS];

  for(i = 0; i < NUMBER_OF_INDIVIDUALS; i++){
    sum_parrallel[i] = temp_individuals[class][i].fitness + temp_individuals[class+1
    sum += sum_parrallel[i];
  }
  int field = rand()\% sum;
  for(i = 0; i < NUMBER_OF_INDIVIDUALS; i++){
    fitness_test += sum_parrallel[i];
    if(field < fitness_test){
      individuals[class][individual_number] = temp_individuals[class][i];
      individuals[class+1][individual_number] = temp_individuals[class+1][i];
      individuals[class+2][individual_number] = temp_individuals[class+2][i];
      return 1;
    }
```

```
  }
}
```

Med denne metode sikres det, at alle individer har en chance for at blive valgt, men at de bedre skemaer, i forhold til fitnessniveauet, har større chance for at blive valgt, og derved få skabt den bedst mulige fremtidige generation. Fitnessen bliver regnet for tre parrallelklasser af gangen og de bliver sendt videre sammen, da deres fitness er afhængig af hinanden. En klasse får f. eks. høj fitness, hvis den har sammenhængende klasser med en parrallelklasse. Derfor hænger parrallelklasser sammen.

7.6 Mutation

Mutation er til for at lave små ændringer i skemaet. Den skal kunne sørge for at der er mangfoldighed, således at skemaet ikke ender i et lokalt maksimum. Måden hvorpå dette foregår er ved at tage et individ, finde to helt tilfældige timer på skemaet og bytte disse ud. Der kommer her to variabler, der kan finpudses for at finde den bedste løsning. Der er en procentvis chance for at en mutation kan ske, og antal mutationer der maksimalt kan ske pr. individ. Her under kan koden til mutations funktionen, der er brugt i programmet, ses.

```
void mutation(individual individuals[]){
  int i = 0, j = 0, ran1Day = 0, ran1Week = 0, ran2Day = 0, ran2Week = 0, chance = 0

  for(i = 0; i < NUMBER_OF_INDIVIDUALS; i++){
    chance = rand()\% 100;
    mutations = rand()\% MAX_MUTATIONS_PER_INDIVIDUAL;
    for (j = 0; j < mutations; j++){
      if (chance > CHANCE_OF_MUTATION){
        do {
          ran1Week = rand()\% SCHOOL_DAYS_IN_WEEK;
          ran1Day = rand()\% LESSONS_PER_DAY_MAX;
          ran2Week = rand()\% SCHOOL_DAYS_IN_WEEK;
          ran2Day = rand()\% LESSONS_PER_DAY_MAX;
        } while ((ran1Week == ran2Week) && (ran1Day == ran2Day));

        temp = individuals[i].individual_num[ran1Day][ran1Week];
        individuals[i].individual_num[ran1Day][ran1Week] =          individuals[i].indi
        individuals[i].individual_num[ran2Day][ran2Week] = temp;
      }
    }
  }
}
```

Der bliver kørt gennem tre for-løkker. Den første tæller klassen op, så der først bliver lavet mutationer på 7.a, så 7.b osv. Dernæst bliver der kørt gennem endnu en for-lykke, som går igennem antallet af individer, hvorefter et tilfældig tal mellem 0 og genereres. Hvis det generede tal er mindre end den valgte chance for mutation, forekommer mutationen. Antallet af ændringer ved en mutation vælges tilfældigt mellem 0 og det højeste antal tilladte mutationer.

7.7 Print funktion

I struct'et 'individual' findes det multidimensionalt integer array 'lesson_num'. Denne indeholder tal som repræsentere de forskellige fag. Det er disse tal der bliver ændret på i de forskellige funktioner.

Når der så skal printes et endeligt skema ud, bliver funktionen 'print_func()' kaldt. Denne funktion tager et array af individer ind som parametre. I dette tilfælde 'chosen_individual'. Funktionen ser ud som følgende:

```
void print_func(individual chosen_individual[][NUMBER_OF_GENERATIONS]){
  int i, j, c;
  for (c = 0; c < NUMBER_OF_CLASSES; c++){
    /* Printing the days */
    printf("  Class: ");
    print_class_name(c);
    printf("  Has a fitness of: \%d   The perfection grade is: \%d \n", chosen_indivi
    printf("  Tidspunkt\t\tMandag\t\tTirsdag\t\tOnsdag\t\tTorsdag\t\tFredag\n");
    printf("  ----------------------------------------------------------------------

    for (i = 0; i < LESSONS_PER_DAY_MAX; i++){
      /* Printing the times */
      print_time_func(i);

      for (j = 0; j < SCHOOL_DAYS_IN_WEEK; j++){
        /* Printing lesson name */
        print_lesson_name(chosen_individual[c][NUMBER_OF_GENERATIONS-1].lesson_num[j
        printf("\t\t");
      }
      /* To signal a break */
      if ((i+1) \% 2 == 0){
        printf("\n");
      }
      printf("\n");
    }
    printf("\n\n\n");
  }
}
```

Funktionen starter med en for-løkke som gentages 'antal klasser'-gange. Nu bliver der printet klassenavnet, dens fitness og antallet af krav den overholder i forhold til antallet af timer den skal have. Efterfølgende bliver der kørt endnu en for-løkke. Denne gentages 'antallet af lektioner'-gange. Nu bliver tidspunktet på dagen printet ud, hvorefter der kommer endnu en for-løkke som printer, de første lektioner af hver dag, ud. Efter dette bliver der printet nye linjer ud, som starter de næste timer. Efter hver anden dag der bliver printet, bliver der printet en ekstra linje ud, som tydeliggøre hvor der er pauser på skemaet.

8 Program test

9 Videreudvikling

På grund af tidspres, eller manglende evner, var der nogle ting der ikke blev implementeret i løsningen. I det følgende afsnit, vil disse mangler blive pointeret.

9.1 Mangler i løsningen

I løsningen er der ingen lærer der har lektioner på mere end et klassetrin. Dette er en mangel, da lærerne på en folkeskole arbejder på tværs af klassetrinene. Dette blev ikke implementeret, da alle klassetrinene skal læses ind på samme tid, for at undersøge om lærerne er sat til at være mere end et sted ad gangen. I løsningen indlæses klassetrinene hver for sig, men parallel klasserne indlæses på samme tid. Det undersøges altså stadig om lærerne er mere end et sted ad gangen, dog ikke på tværs af klassetrinene. Derudover er der kun blevet arbejdet med udskoling. Der mangler derfor at blive generet skemaer, for klasserne i indskoling mellemskoling. For at kunne genere skemaer for indskoling og mellemskoling, skal der indsættes nye 'requirements' for disse klassetrin.

9.2 Graphic user interface

Hvis programmet reelt skulle være et produkt der kunne sælges, ville det kræve en gui, eller graphic user interface. Hvis brugeren skal kunne ændre den fil der bliver læst ind, og de ændringer de laver skal være formateret på samme vis, som det der er i dokumentet nu. Hvis brugere ændrer dokumentet og det ikke står i samme format, vil filen ikke blive læst ind korrekt og programmet vil lukke. En reel gui ville derfor være den bedste måde hvorpå brugeren kan udføre input der af programmet nedskrives i en fil. For at opstille den bedst mulige gui, ville man skulle undersøge brugervenlighed, for at finde det bedst mulige design. I gui'en skulle brugeren have mulighed for at ændre antallet af lærere, deres navn/initialer samt hvor højt de ville prioritere forskellige bindinger. Det aktuelle program er programmeret til Sofiendahlskolens præferencer, og ville derfor ikke kunne bruges på en skole, hvor de vægter f.eks. senere møde timer højt.

Derudover er der heller ikke taget højde for lokale reservering i det aktuelle program. Brugeren vil derfor være nødsaget til selv at uddele lokaler efter genereringen af skemaet. I videreudviklingen af programmet, ville en forbedring kunne være at hver klasse bliver tildelt et lokale samt en lærer, og at lokalerne i nogle tilfælde skulle tilhører de forskellige klasser, eller fag, således at idræt f.eks. kun kan foregå i idrætshallen. På den måde, ville det også kunne bestemmes hvilke klasser der for eksempel kan have idræt sammen. Derudover er antallet af parralelklasser, antallet af fag og antallet af klassetrin ikke fleksibelt, så det kan bruges på andet end en udskoling med tre parrelelklasser. Programmet kan delvist tilpasses til nye krav ved ændring af defines i source koden. Ved videreudvikling vil denne del gøres me fleksibel.

En log ind mulighed for skolerne ville også kunne forbedre programmet. At brugeren kan logge ind, kan muliggøre at skemaerne kan gemmes på en profil. Det vil derfor ikke være nødvendigt generer et nyt skema, hver gang programmet køreres. I det aktuelle program er brugeren nød til at gemme en kopi af det generede skema på computeren manuelt for at kunne gemme det. En log ind mulighed vil endvidere kunne gemme skolens præferencer, således at præferencerne ikke skal skrives ind hver gang der skal genereres et nyt skema.

9.3 Brugertest

Hvis en gui bliver implementeret i programmet, ville det også have været relevant at foretage brugertests. En brugertest ville forsikre at gui's interface er brugervenligt, og overskueligt. I interviewet der blev foretaget med Sofiendahlskolen, blev det pointeret at de førhen havde forsøgt sig med softwareløsninger til planlægningen af deres skema. De programmer de havde brugt syntes de dog var for besværlige at sætte sig ind i, de endte derfor med ikke at fortsætte med softwareløsningerne.[47] Det er derfor vigtigt at programmet er brugervenligt, da skolerne ellers ikke vil fortsætte med at bruge produktet.

9.4 Empiri fra flere skoler

For at kunne programmere en bedre og mere fleksibel softwareløsning til skemalægningsprocessen, skal der samles mere empiri fra flere skoler. Den aktuelle løsning er begrænset, i det den kun løser Sofiendahlskolens problemer, og kun tager højde for deres bindinger. Hvilke bindinger de forskellige skoler, har og hvordan de prioriterer dem er meget forskelligt. Da løsningen kun tager højde for Sofiendahlskolens bindinger, kan løsningen ikke bruges optimalt på andre folkeskoler, medmindre, de har præcis de samme bindinger som Sofiendahlskolen. At indsamle mere empiri ville også hjælpe med at danne et større perspektiv over de problemer, der forekommer i skemalægningsprocessen på andre skoler.

For at skabe mere brugervenlighed og effektiv skemalægning for skolernes personale, kunne programmet også udfra en valgt kommune af brugeren, sørger for hvert år automatisk at opdatere kravene til timeantal osv. udfra kommunernes og undervisningsministeriets krav.

I gui'en kunne det også designes så, brugeren kunne flytte lektioner, ændre lærere og lokaler på lektioner osv. hvorefter programmet ville vise, hvilke konflikter ændringen ville skabe og/eller planlægge mulige ændringer, så kravene stadig opfyldes.

[47] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

10 Diskussion

11 Konklusion

For at få indblik i bindingerne relateret til skemaplanlægningsprocessen er en skemaplanlægger fra Sofiendalskolen blevet interviewet. Interviewet gav indblik i de specifikke problemer, der opstår når de planlægger skemaer på Sofiendalskolen. Derudover gav interviewet også indblik i problemerne relateret til de eksisterende softwareløsninger. Det blev dog hurtigt tydeligt, at en stor mængden af bindinger skulle tages højde for i skemalægningsprocessen. Det blev derfor valgt at begrænse hvilke parametre, der skulle tages højde for i softwareløsningen.

Et af de valgte bindinger, der skulle prioriteres højest, var lærernes forberedelsestid. På Sofiendalskolen synes de, det er vigtigt, at lærerne har to forberedelsestimer før hver lektioner, så det bliver prioriteret i softwareløsningen. Derudover bliver det prioriteret at parallel klasserne har mulighed for at arbejde sammen, og at der er så lidt tunge fag over middag som muligt. Programmet skal også overholde de lovmæssige krav om undervisningstimer for de individuelle fag, samt de forskellige fag hvert klassetrin skal have.

I planlægningen til softwareløsningen blev det valgt at bruge genetisk algoritme grundet kompleksiteten af skemalægningsprocessen. Kompleksiteten var bundet i umuligheden i at finde et idéelt skoleskema, og genetiske algoritmer gav en løsning ved at tilnærme sig et idéelt skema. Derudover blev det vurderet hvilken selektions metode, der skulle bruges i den genetiske algoritme. Her blev roulette metoden valgt. Roulette metoden blev valgt, da det er den metode, hvor fitness har mest indflydelse på hvilke individer, der bliver valgt til reproduktion.

Ud fra den skabte softwareløsning kan det konkluderes, at genetiske algoritmer er en effektiv metode til at generer skemaer. Det kan observeres i test af programmet, at skemaerne i gennemsnit bliver bedre i takt med, at der bliver generet flere generationer. Fitnessen når dog et plateau efter et vis antal generationer. Plateauet kan skyldes, at der ikke er nok variation i generationerne, og der derved ikke bliver genereret skemaer, der er forskellige fra hinanden. Den skabte softwareløsning kræver dog optimering og flere egenskaber, før det ville kunne blive brugt på en reel skole, disse mangler er blevet diskuteret i videreudviklings afsnittet. Den vigtigste af disse mangler er bruger input, for at simulere bruger input i programmet indlæses en datafil. Da de forskellige skolers bindinger og prioriteter er forskellige, skal der indgå bruger input for at programmet skal være fleksibelt nok til at kunne bruges på andre skoler end Sofiendalskolen.

References

[1] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen*. Oct. 2016.

[2] *Bekendtgørelse af lov om folkeskolen*. 2016. URL: https://www.retsinformation.dk/forms/r0710.aspx?id=182008.

[3] *Timetal (minimumstimetal og vejledende timetal) for fagene i folkeskolen*. URL: https://www.uvm.dk/-/media/UVM/Filer/Udd/Folke/PDF15/Juli/150710-Timetalsskema_PDF.ashx?la=da.

[4] *Undervisningstimetal i folkeskolen*. Aug. 5, 2016. URL: https://www.uvm.dk/Service/Statistik/Statistik-om-folkeskolen-og-frie-skoler/Statistik-om-elever-i-folkeskolen-og-frie-skoler/Statistik-om-undervisningstimetal-i-folkeskolen?allowCookies=on.

[5] *Den nye folkeskole - en kort guide til reformen*.

[6] *Docendo*. Sept. 2014. URL: https://www.youtube.com/watch?v=F-heCUy4bZ4.

[7] *Docendo eksempel*. Docendo. URL: https://docendo.dk/folkeskole.html.

[8] URL: http://timetablingturbo.com/advantages.html#1clicktimetabling.

[9] *Lantiv Scheduling Studio 7*. Lantiv. Nov. 2016. URL: https://www.youtube.com/watch?v=Q_SJB0Vr8Ds.

[10] *tabulex 2015 SKEMALÆGNING*. Tabulex. 2015. URL: http://docplayer.dk/3713728-Tabulex-2015-skemalaegning.html.

[11] URL: http://untis.dk/.

[12] *Untis billede*. Untis. July 2013. URL: http://untis.dk/wp-content/uploads/2013/07/WebUntis-l%C3%A6rerskema.gif.

[13] *Brute force cracking*. 2016. URL: http://searchsecurity.techtarget.com/definition/brute-force-cracking.

[14] Khalid Jebari and Madiafi Mohammed. "Selection Methods for Genetic Algorithms". In: (Dec. 1, 2013).

[15] Patrick Winston. *13. Learning: Genetic Algorithms*. 2014. URL: https://www.youtube.com/watch?v=kHyNqSnzP8Y.

[16] *Selection*. URL: http://www.obitko.com/tutorials/genetic-algorithms/selection.php.

[17] *Genetic Algorithms - Parent Selection*. tutorialspoint. URL: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm.