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

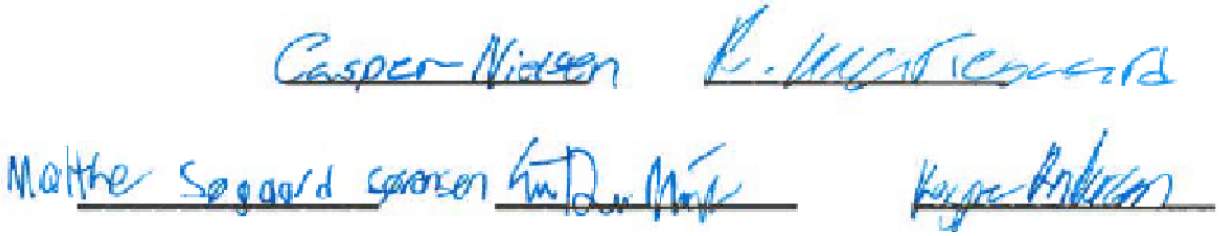
Simon Dam Nielsen

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:
Genetiske algoritmer til generering af skoleskemaer i folkeskoler

Tema:
Fra eksisterende software til modeller

Projektperiode:
10/10-2016
19/12-2016

Projektgruppe:
B2-26

Deltagere:
Casper Nielsen
Kasper Andersen
Malthe Sørensen
Rasmus Mariegaard
Simon Nielsen

Vejledere:
Claus Skaaning
Søren Lykke

Synopsis:

I følgende projekt arbejdes der med folkeskoleskemaplanlægning softwaremæssig problemstilling. Der blev foretaget et interview med en skemaansvarlig fra Sofiendalskolen. Ud fra dette interview blev der fremstillet en case-study, som undersøgte hvilke problematikker Sofiendalskolen har i deres skemalægningsproces.
Derefter blev eksisterende softwareløsninger til skemalægning undersøgt, herunder Tabulex som Sofiendalskolen tidligere havde forsøgt brug af.
Genetiske algoritmer blev undersøgt og deres relevans i forhold til skemalægning blev vurderet positivt.
Det blev forsøgt at fremstille en softwareløsning, som ved brug af genetiske algoritmer genererede skoleskemaer.

Oplagstal: 1
Sideantal: 50
Bilagsantal og -art: 1
Lydfiler af interview
Afsluttet den: 19/12/2016

1 Forord

Vi vil gerne takke Sofiendalskolen for at lade os interviewe Søren Kusk, og specielt tak til Søren Kusk for hans tid.

2 Abstract

The following project examines the use of genetic algorithms for use in the process of planning elementary school schedules. An interview with a schedule planner from Sofiendalskolen was conducted. From this interview a case-study was produced, which examined the problems which occurred in the scheduling process in Sofiendalskolen. Existing scheduling programs were examined afterwards, once of which were Tabulex, a program Sofiendalskolen previously had attempted to use. Genetic algorithms relevance was examined in context of the planning of school schedules. It was concluded to be adequate for generating school schedules. An attempt was made to produce a software solution which used genetic algorithms to generate school schedules. The software solution was concluded to produce an adequate schedule, but with an ineffective genetic algorithm.

Contents

3 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 tillade lærerne at fokuserer på planlægningen af deres undervisning og eleverne får derved får bedre undervisning. I tilfælde af at lærerne synes skemaet er uoverskueligt og ikke passer til deres kriterier, vil det betyde, at andre lærere bliver nødt til at gå på kompromis og bytte lektioner. Det er derfor en fordel, hvis et endeligt skema bliver lagt i starten af skoleåret, så forvirring ikke bliver skabt i løbet af året.[1]

Der er mange parametre, der skal tages højde for, når skoleskemaet planlægges. 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, som kan danne skoleskemaer, men skoler som Sofiendalskolen vælger alligevel at lægge deres skema i hånden hvert år. Sofiendalskolens proces er langvarig og kostelig, da den kræver, at alle 70 lærere og pædagoger på skolen er til stede, mens de diskuterer skemaets opbygning. 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 skoleskemaet planlægges i folkeskolerne, og hvorledes kan et program 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.

4 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 der 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, der er nødvendig at opfylde, når løsningen skal laves. Kravene beskrevet i dette afsnit er nødvendige for forståelsen af opbygningen af folkeskoleskemaer. I afsnittet tages der udgangspunkt i de kriterier, som beskrives i "Bekendtgørelse af lov om folkeskolen" med supplerende kilder.

Interessentanalysen vurderer de påvirkende og påvirkede interresenter i forhold til skemalægningen. Afsnittet lægger fokus på vurderingen af interessenternes rolle i skemalægningsprocessen. Analysen udarbejdet i dette afsnit giver grundlag for valget af kontaktede interresenter.

En vital del af problemanalysen er case-study'en, som blev lavet med Søren Kusk fra Sofiendalskolen. 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. Underafsnittende præsenteret i dette afsnit vil analyseres og udvides med tilstrækkelig information, så en fyldestgørende problemafgrænsning kan konstrueres.

4.1 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 på et helt skoleår, se afsnittet "Lovmæssige krav til folkeskoleskemaet".[3][4]

Kommunerne har også indflydelse på skemaplanlægningen.[5] F.eks. har folkeskolerne i Aalborg Kommune flere lektioner end folkeskolerne i Rebild Kommune.[6]

Skoleskemaet er bygget op således, at eleverne har en række fag hver dag med pauser mellem lektionerne. 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

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

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

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

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

have lektioner på 90 minutter med længere pauser ind i mellem. Derudover har lærerne forberedelsestimer, når de ikke underviser. Det vil sige at lærerens forberedelsestime potentielt kunne ligge 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.[7] Da mange parametre og krav skal tages stilling til af den skemaansvarlige, kan skemaplanlægningsprocessen være langvarig.[8]

4.2 Lovmæssige krav til folkeskoleskemaet

Regeringen stiller en række krav til skolerne under skemaplanlægningsprocessen, der giver rammer 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 løsningen kan overholde de lovmæssige krav, der stilles. En casestudy, som vil inddrages senere, vil bearbejde dette problem yderligere. Her opstilles kravene for de forskellige fag, som eleverne i den 9-årige grundskoleuddannelse skal følge.[9] På figur 1 ses hvilke fag hvert klassetrin skal have undervisning i.

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 de enkelte årgange skal have.[10]

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.[11]

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

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

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

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

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 3 er minimumskravet for første klasse 750 timer på et 200 dags skoleår, hvilket udgør 18,75 undervisningstimer ugentligt.[12]

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.

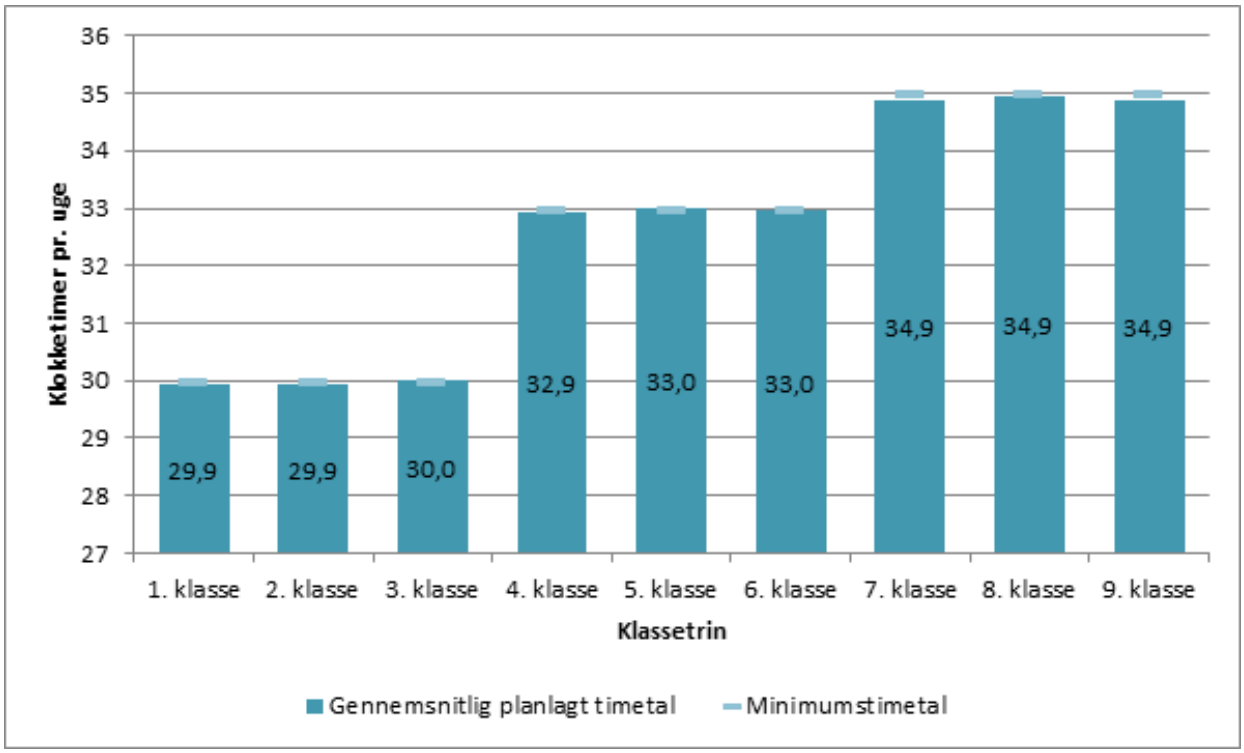


Figure 2: Gennemsnitligt planlagttimetal kontra minimumstallet for 1.-9.-klasse.[13]

Dette stemmer overens med figur 2.[14] Dog skal det noteres at for alle fag ekskluderende, dansk matematik og historie[15] er disse vejledende timeantal. De skal også opfylde undervisningstidens samlede længde, der ses nederst på figur 3. Pauser indgår også i denne undervisningstid.[16]

[12] *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.

[14] *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.

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

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

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 3: Skema af kravene af timetal fra folkeskoleklasserne 0.- 9.[17]

4.2.1 Skolereformen

I 2014 blev den nye skolereform introduceret i folkeskolerne. Reformen opstillede tre klare mål for folkeskolerne:[18]

- 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 i særligt udfordrende fag ved brug af lektiehjælp.

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

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.[19] Skolereformen har yderligere fokus på hævning af de pædagogiske kompetencer hos lærerne.

4.3 Case study - Sofiendalskolen

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

Sofiendalskolen blev opført i 1911. I dag er Sofiendalskolen en tresporet skole med en special ADHD klasse, hvilket betyder at der på en årgang befinder sig 3 klasser a, b og c samt speciel klassen. Skolen rummer 70 lærere og pædagoger, som underviser 650 elever dagligt.[20]

Lørdag den 27. oktober blev Søren Kusk interviewede. Søren Kusk fungerer som matematiklærer samt it-ansvarlig på Sofiendalskolen og indgår i et team af 8 lærer, som underviser 3. klasse. To af disse lærere underviser også i andre klasser. På Sofiendalskolen er det ikke en bestemt person, der er ansvarlig for at lægge skemaet. Derimod mødes alle involverede pædagoger og lærere to onsdage før starten af skoleåret. Her afholdes to møder med en varighed på tre timer hver, hvilket vil sige, at det tager ca. 420 mandetimer for Sofiendalskolen at lægge skoleårets skema. Skemalægningsprocessen 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 antal lektioner, de skal undervise. Herefter får de hver især farvede brikker, som repræsenterer de lektioner som lærerne underviser. Brikkerne lægger de på et tomt skema ud fra de klasser, de skal undervise i, indtil de ikke har flere brikker.[21]

Når skemaet lægges på Sofiendalskolen ønskes det, at lærerne har sammenhængende forberedelses timer, således de ikke er spredt ud over hele ugen. De mener, at de ikke får chancen for at forberede sig ordentligt, hvis de kun har en forberedelsestime af gangen. Derudover prioriterer de, at eleverne ikke har tunge fag, som f.eks. matematik over middag, da eleverne tit er trætte sidst på dagen, 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 ofte på kompromis med parametrene, og vælger hvilke parametre de prioriterer højst. Skemaplanlæggerne bruger på nuværende tidspunkt ikke skemalægningsprogrammer, da programmerne, de har afprøvet, ikke har kunnet tage højde for flere parametre og præferencer, spredt ud over de forskellige teams og klasser.[22]

Sofiendalskolen har prøvet at lægge skema med programmet Tabulex. Skemaplanlæggerne synes Tabulex var komplekst og avanceret at opstille for hver klasse, da hvert team har forskellige præferencer. Det var derfor svært for Tabulex at lægge skemaet, da Tabulex ikke var fleksibelt nok til

[19] *Den nye folkeskole - en kort guide til reformen.*
[20] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.
[21] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.
[22] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

deres behov. Derfor har Sofiendalskolen valgt at planlægge skemaerne manuelt. Dog vil Sofiendalskolen stadig være åben for brugen af et computerprogram til skemalægningsprocessen, hvis programmet kunne hjælpe dem med at gøre processen kortere uden at gøre det for avanceret at få sat op.[23]

4.4 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 eller påvirker en softwareløsnings 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, 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.[24]

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 løsningen 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 softwreløsningen viser sig at være en dårlig investering.[25]

Lærerne bruger tid og kræfter på skemalægningsprocessen.[26] 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 operere, da de er med til at planlægge skemaet. Deres indflydelse påvirker om en softwareløsning vil blive implementeret på en skole, da softwareløsningen skal være i stand til at opfylde lærernes betingelser for at skoleskemaet kan kaldes fejlfrit. Lærerne er ikke interesseret i et program, som skaber flere problemer for dem, end det reelt løser. Lærernes betingelser til en softwareløsning består i at få optimeret skemaet, så eleverne er fokuseret i undervisningen, og at underviserens forberedelsestimer er samlet. Derudover har Sofiendalskolen opdaget, 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. Softwareløsningen ville også have en stor påvirkning på lærernes hverdag, da de arbejder ud fra skoleskemaet, og problemer som skoleskemaet eventuelt skaber, vil have direkte påvirkning på lærerne.[27]

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

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

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

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

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

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 bliver påvirket af en softwareløsning, i tilfælde af at skemaet potentielt bliver bedre ved softwareløsningen. Et dårligt planlagt skema vil betyde, 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 bedste faglige 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.[28]

For at overskueliggøre de forskellige interessenter og deres indflydelse på, hvordan skemaet bliver planlagt, og hvilken påvirkning skemaet vil have på dem, kan de forskellige interessenter opstilles i et skema som ses på figur 4.

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 4: Model over de indblandede interessenter.

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

4.5 State of the art

Skemalægningen har længe været et problem, diverse skoler har haft svært ved at løse, heriblandt er Sofiendalskolen.

Selvom der findes mange skemalægningsprogrammer, er det ikke nødvendigvis ensbetydende med, at alle skoler kan benytte disse programmer. Nogle skoler har svært ved at begrænse sig til så få parametre, som programmerne kan behandle. 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."[29] Dette tydeliggører problematikken i, at skemalægningsprogrammerne ikke er i stand til at behandle specfikke parametre.

4.5.1 Docendo

Skemalægningsprogrammet Docendo er et brugervenligt samt forholdsvis simpelt program. Programmet danner en kalender med uger og dage, hvorefter brugeren har mulighed for at justere diverse parametre alt efter behov. Heriblandt tager programmet bland andet højde for, at nogle skoler har forskellige fag, og giver derfor brugeren mulighed for at tilføje flere fag. Samtidig har brugeren mulighed for at tilpasse lektionernes længde, hvilket giver programmet fleksibilitet. Dernæst fastlåser programmet lokaler og lærere, som har undervisning på bestemte tidspunkter. Det forhindrer dobbeltbookninger i et bestemt lokale eller lignende. Hvis et problem skulle opstå, kan lektionerne flyttes.[30]

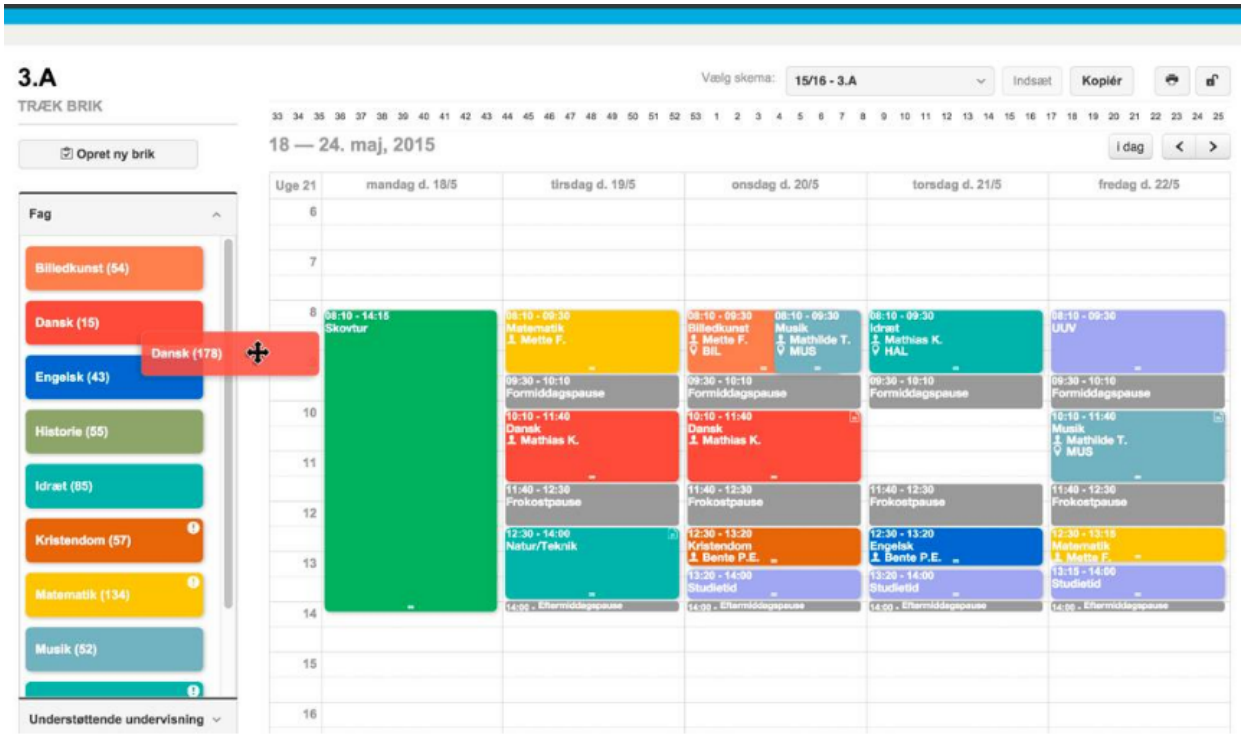


Figure 5: Eksempel på skemplanlægning i Docendo.[31]

Da Docendo ikke genererer et endeligt skema, men et skema, som skal rettes i, betyder det at der stadig skal investeres en mængde tid i skemaplanlægningen.

[29] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.
[30] *Docendo.* Sept. 2014. URL: `https://www.youtube.com/watch?v=F-heCUy4bZ4`.

4.5.2 Lantiv

Lantiv Timetabling Turbo 7 er et skemaplanlægningsprogram, 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 af 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 stiger for at løse de sværere overtrædelser. Denne proces stopper, når programmet enten har løst alle overtrædelser, eller den 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. Hvis det er en dobbeltbookning, er der mulighed for at slette en af lektionerne eller accepterer dobbeltbookningen.[32]

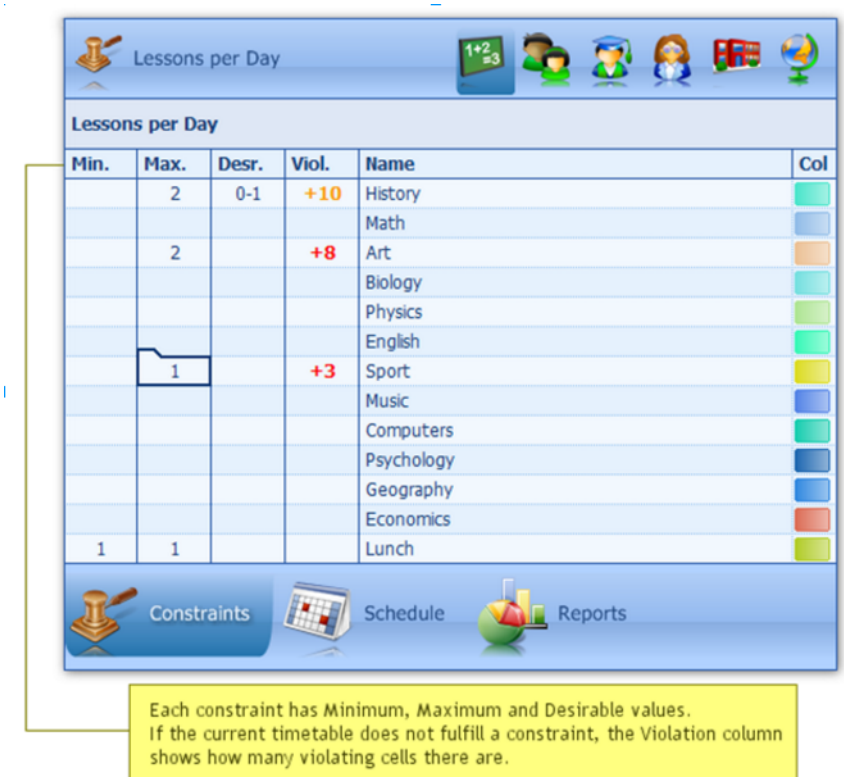


Figure 6: Eksempel på skemplanlægning i Lantiv.[33]

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

4.5.3 Tabulex

Tabulex er et dansk skemalægningsprogram, som fungerer ved at brugeren først indtaster alle sine ressourcer (lærere, lokaler, klasser, fag og placeringsregler).

Under hver enkel lærer sættes maksimum antal mellemtimer, altså timer hvor de ikke underviser. Derudover vælges maksimum og minimum antal lektioner per dag på specifikke ugedage samt blokeringer, der gør at læreren ikke kan have undervisninger på bestemte tidspunkter.

Samme parametre kan sættes for klasserne samt tre yderligere. En parameter angiver hvor sent eleverne må møde hver dag. De kan altså godt møde tidligere nogle dage, men ikke senere. En anden parameter angiver efter samme princip, hvornår eleverne tidligst får fri. Sidste parameter angiver, om det er et krav, at klassen skal møde hver dag. Hvis lærerne har arbejdsopgaver udenfor klasserne, er det nødvendigt at lave fiktive klasser uden elever dertil.

Under hold er det muligt at lave et hold af flere klasser.

Under fag indtastes fagene til klasserne enkeltvis. Der kan sættes et krav til, at en klasse ikke må have fagrepetition, altså to blokke af samme fag på en dag. Som udgangspunkt forsøger Tabulex at undgå fagrepetition, men hvis dette krav ikke er valgt, tillades det i nogen situationer. Fagene kan også lægges i faggrupper, hvor Tabulex også vil forsøge at undgå repetition af fag i samme faggruppe. Lokaler bliver inddelt i grupper, efter hvilke fag de kan bruges til. Derudover har de også en blokeringsmulighed, hvis lokalerne ikke kan bruges på specifikke tidspunkter.

Der findes fire placeringsregler i Tabulex, men flere kan oprettes manuelt. De eksisterende placeringsregler kan bruges til at bestemme, at en binding skal ligge i første lektion, at den skal ligge i sidste lektion eller at den skal ligge i enten første eller sidste lektion. Den sidste placeringsregel bestemmer, at en lektion ikke må være skemalagt to dage i træk.

Efter alle ressourcer er defineret, sættes de sammen med bindinger med lærer, klasse, fag, lokale og eventuelt placeringsregel. Når brugeren laver bindingerne, kan der løbende følges med i hvor mange lektioner, der er afsat til hver lærer, klasse og lokale. Efter disse informationer er indtastet manuelt, kan den automatiske skemalægningsfunktion bruges. Denne funktion leder efter fejl i det manuelle skema og forsøger derudfra at oprette bedre skema. Under denne proces kigges på parametre som antallet af mellemtimer og overholdelse af krav om undervisning hver dag, mødetider, gåtider, antal lektioner på en dag og antallet af blokeringer overtrådt.[34]

4.5.4 Untis

Untis er et fleksibelt skemalægningsprogram, som både er nemt at bruge og giver mange muligheder for brugeren. Untis giver bland andet mulighed for at brugeren kan bestemme, hvordan skemaet 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-uge funktionen gøre således, at der kommer 4 matematiklektioner i en uge og 6 i en anden. Desuden har brugeren mulighed for at ændre skemaet 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 førnævnte. Dette sker ved at brugeren udfylder udvalgte brikker, og herefter benytter Untis til at udfylde resten for at opnå det

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

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 vise hvori problemet ligger, så brugeren nemt kan 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 de korrekte eller realistiske antal af eksempelvis lokaler, lærer, klasser mm.[35]

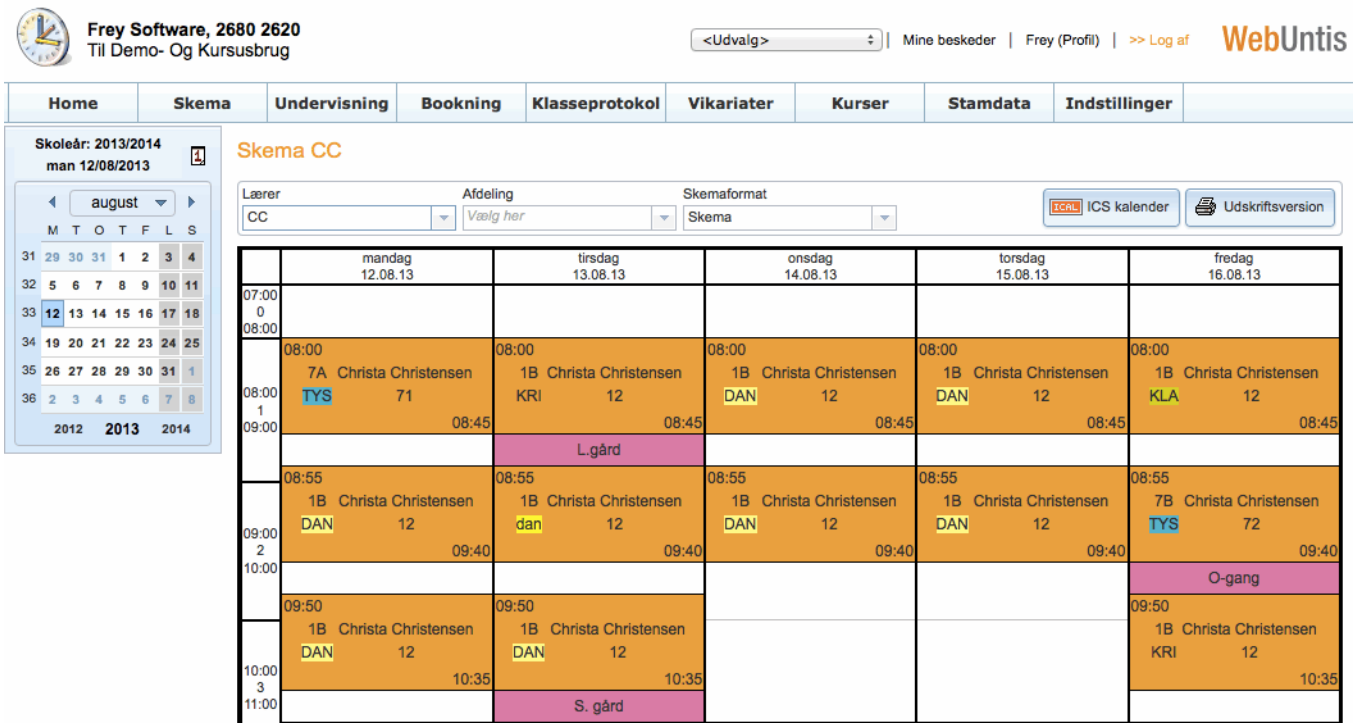


Figure 7: Eksempel på skemplanlægning i Untis.[36]

4.6 Problemerne relateret til eksisterende softwareløsninger

Et problem ved brug af eksisterende skema kan være, at de ikke er designet til en specifik skole, og må være meget fleksible, for at brugeren kan tilpasse programmet til eget behov. Denne tilpasning kan være et problem, da det kræver at brugeren har større forståelse for programmet, samt lægger arbejdstimer i tilpasningen. Derudover prioiterer et program nogle bindinger over andre. Det kan dog være individuelt for, de forskellige lærere hvilke bindinger, de prioiterer højt, og det kan derfor være svært for et softwareprogram, at gøre alle tilfredse.[37]

4.7 Algoritmer

I følgende afsnit vil der redegøres for forskellige algoritmetyper til løsningen af generering folkeskoleskemaer. Til sidst vurderes hvilken algoritmetype er mest hensigtsmæssigt til skoleskemaplanlægning.

4.7.1 Genetiske algoritmer

Genetiske algoritmer er en metode til at optimere en løsning til et givent problem. Genetiske algoritmer bruger en række retningslinjer, som ud fra disse danner flere løsninger til et problem

[35] URL: `http://untis.dk/`.
[37] swb2 26. *Interview med Søren Kusk fra Sofiendahlskolen.* Oct. 2016.

og finde den bedste løsning til problemet.

Genetiske algoritmer er en metode, som er inspireret af Charles Darwins teori om naturlig selektion, der omhandler hvordan de biologiske arter udvikler sig over tid. Teorien danner grundlag for, 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.

En genetisk algoritme består af følgende dele.[38]

1. Individer som er mulige løsninger til problemet.
2. Fitness der 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.
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.[39]

Crossover Operatoren vælger to tilfældige kromosomer og blander dem, så der bliver dannet to nye kromosomer, som er en kombination af de to kromosomer f.eks. strengene 1110100 1011111 bliver krydset og danner de to nye strenge 1010100 1111111.[40]

Mutation Operationen ombytter tilfældigt nogle stykker af kromosomet, f.eks. 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 der ved at muterer kromosomet udforskes flere muglie løsninger til det givende problem.[41]

På figur 8 ses variationer af skemaer hen ad x-aksen og fitness værdien op af y-aksen. Det første toppunkt markeret med "lokalt maksimum" indikerer et punkt, hvor en variation har den bedste fitness værdi i forhold til andre tætliggende variationer. Men et globalt maksimum, som indikerer den bedste fitness værdi et skema kan have, eksisterer også. Et problem opstår, hvis produktet er fanget i det lokale maksimum. Dette kan forekomme, hvis produktets fitnessværdi ender i det lokale maksimum og der ikke bliver lavet store ændringer nok til at få variationen hen til punktet P. Før punktet P vil variationen ikke være nok til at få en høj fitness værdi. Det er derfor vigtigt, at skemaet bliver ændret til en grad, som kan tillade at det kan bryde ud af det lokale maksimum. Det skal bemærkes, at skemaet ikke nødvendigvis bliver bedre med en højere variation, men det gælder om at varierer den nok gange indtil fitnessværdien er høj nok. Yderligere skal det bemærkes, at figur 8 skal ses, som en generalisering af princippet og ikke realistisk afbildning. Det er vigtigt, at forstå, at variation kan være en stor mængde parametre, som ikke kan afbildes i en 2-dimensionel graf.

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

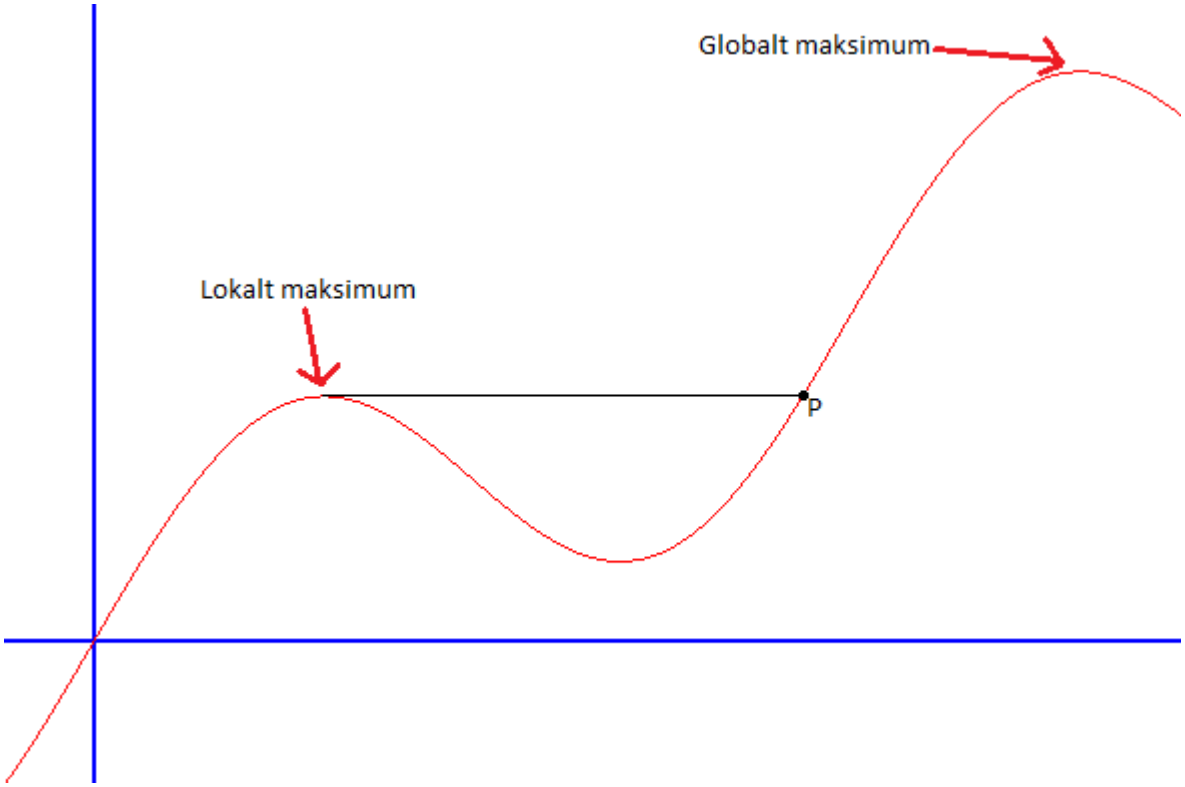


Figure 8: Afbildning af variationer af skemaer på x-aksen og fitness værdi på y-aksen

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 fem 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 det at programmet nu hænger fast, da tallene omkring disse parametre ikke gør at der kommer en højere fitness. Ud fra parametrene er det muligt at udregne den højeste fitness. Minimale ændringer vil ikke nødvendigvis resultere i den højeste fitness, hvorimod store ændringer i parametrene muligvis vil forårsage en højere 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 et individ med, der udover at have god fitness, ikke ligner de andre.

Ved brug af 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.[42]

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.

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

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

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.[43]

4.7.2 Udfaldsrum

Udfaldsrum 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. udfaldsrum 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 udfaldsrum 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.[44]

4.7.3 Brute Force

Brute-force algoritmen går i sin simple form ud på at gennemgå alle mulige kombinationer af problemet, og beregne en fitness for hver 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 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.[45]

4.7.4 Genetiske algoritmer i relation til skoleskemaer

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

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

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

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

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 stiller.

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 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.

5 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, som 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 der er mange variabler som 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.

6 Problemafgrænsing

I interviewet med Søren Kusk fra Sofiendalskolen, er information indsamlet, hvilke krav skolen stiller, når skemaet lægges, 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.[46]

Ud fra empiri fra interviewet samt research af state of art, kan det konkluderes, at de eksisterende skemaplanlægningsprogrammer på markedet har svært ved at tage hensyn til Sofiendalskolens krav.

For Sofiendalskolens lærere er det nødvendigt at indgå kompromisser, 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 skemaplanlægningsprogrammer.

Ud fra denne viden er det blevet besluttet at lave et program, der kan lave et grundskema, som stemmer overens med de lovgivningsmæssige krav for elevernes timetal. Skemaet skal tage højde for, at hver klasse og lærere ikke kan have mere end en lektion af gangen. Ved brug af generisk algoritmer laves et fitness 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, da dette vægter højt for Sofiendalskolen. Derudover skal programmet være i stand til at vurderer fitness på tværs af klasser, så programmet giver mulighed for at vægte skeamer med tværfaglige lektioner højere.[47]

Programmet vil løse en del af det tidsmæssige problem for lærerne samt problemet med fordelte forberedelsestimer.

7 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 prioriteres i programmet?

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

8 Designovervejelser

8.1 Selektion

Som tidligere diskuteret er processen for genetiske algoritmer følgende:

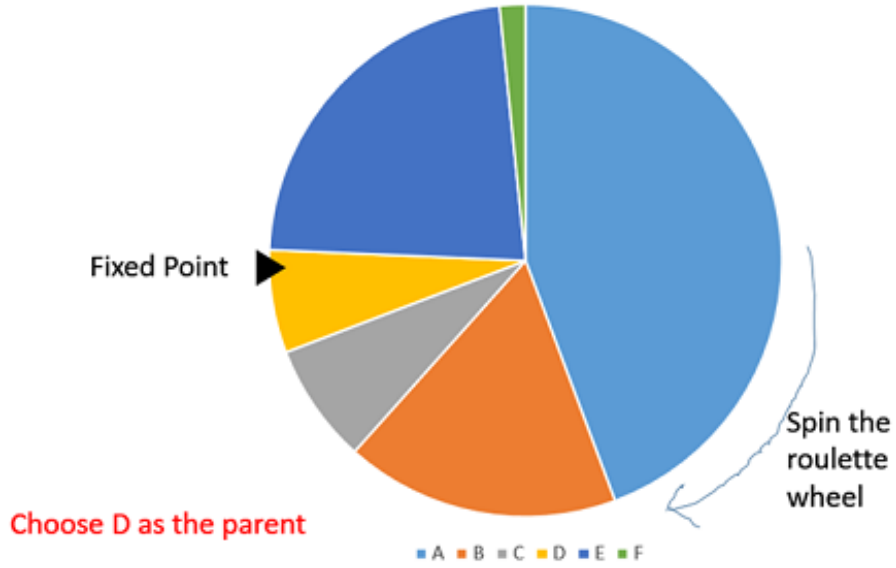
- Forældre individerne bliver valgt.
- Forældrene parres.
- Populationen vokser.
- Individerne bliver muteret, eller krydset.
- Sandsynligheden for at en crossover eller mutation finder sted bliver bestemt ud fra en selektionsmetode.

I det følgende afsnit beskrives nogle af de selektionsmetoder, der kan bruges. Det vil endvidere også blive diskuteret, hvilken af disse metoder, som bedst kan anvendes til at producere skoleskemaer.[48]

8.1.1 Roulette metoden

Roulettemetoden går ud fra et hvert individ i en generation får tildelt et felt på en roulette. Individets fitness niveau afgør størrelsen på individets felt. Et tilfældigt punkt vælges på rouletten. Det felt, det tilfældige punkt ligger på, bliver valgt. På denne måde vil individer med gode fitnessværdier oftest blive gemt, men individer med dårlige fitnessværdier kan også mindre ofte gå videre og derved påvirke genitikken.[49][50]

Roulette metoden er illustreret på figur 9.



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

Figure 9: Model over roulette metoden.[51]

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

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

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

8.1.2 Rang 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 det individ 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.[52]

8.1.3 Tournament metoden

To tilfældige individer bliver valgt fra populationen. En tilfældig værdi fra 0-1 genereres for at sammenligne den med valgte sandsynlighedsværdi. Hvis værdien er mindre eller lige med 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.[53]

I det følgende program benyttes roulette metoden. Ved brug af roulette metoden er der forskel ved sammenligning af to individer, hvor deres fitness ligger tæt på hinanden, og to individer hvor deres fitness ligger langt fra hinanden. Tournemant metoden laver ikke nogle forskel på disse situationer og er derfor ikke optimal til dette problem.

8.2 Produktafgrænsning

At lave et godt skema stiller store krav til et skemalægningsprogram. Det er vigtigt, at programmet er i stand til at møde disse krav, ellers er programmet essentielt ubrugeligt. Det grundlæggende princip i programmet er, at automatiserer skemalægningsprocessen ud fra forudbestemte parametre. Grundet den begrænsede tid, der er afsat til dette projekt, vil følgende afsnit afgrænse hvilke funktionaliteter, som vil blive løst i det endelige program.

Som tidligere nævnt vil programmets selektion være baseret på genetiske algoritmer. Brugen af genetiske algoritmer giver mulighed for at buge programmet gentagende gange med forskellige resultater. En bruger kan køre programmet indtil, der genereres et skema, de anser som tilfredsstillende.

Programmet skal tage højde for prædefinerede krav, både lovmæssige og brugerspecifikke.

De lovmæssige krav er defineret i afsnittet om lovgivning. Der er ikke meget debat om at disse krav er essentielle grundsten, og derfor skal inkluderes for at lave et acceptabelt program.

I interviewet med Søren Kusk blev konkrete problemer understeget. Forberedelsestimer til lærerne, som ligger samlet, tunge fag skal ligge før middag, tværfaglig undervisning på tværs af klasser.[54]

Programmet skal arbejde med at løse tre problemer. Som nævnt i problemafgrænsningen er samlede forberedelsestimer et vigtigt krav at få opfyldt. Det er vigtigt for lærerne at have tilstrækkelig konsekutive forberedelsestimer, så undervisningstimerne er af tilstrækkelig kvalitet for elevernes indlæring. Det er derfor medtaget i programmet, som et at de primære problemer, som skal løses.

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

Et andet problem der er valgt til at løse, er problemet omkring tunge fag efter middag. Dette var et problem, som var blevet gjort til kende af Søren Kusk og som en præference, der ønskes at medtages i skemalægningen.[55] Det er derfor også et krav, som vælges at tages højde for i forhold til programmet, både på grund af at det var et problem, som blev nævnt i interviewet, men ydermere et koncept genetiske algoritmer har potentiale til at løse ved hjælp af selektion og nedprioritering af skemaer med sådanne hændelser.

Sidste prioritet fra interviewet, som vælges at programmet skal tage højde for, er kravet om tværfaglige undervisning på tværs af parallelklasserne. Der stilles krav om, at lærerne kan undervise parallelklasser samtidig. Dette er dog kun muligt, hvis to eller mere parallelklasser har det samme fag samtidig. Derfor skal programmet være i stand til vurdere placeringen af specifikke fag og lærere på tværs af parallelklasser.

Disse krav er lavet ud fra interviewet, men yderligere krav blev opstillet. Kravene er som følgende:

Sofiendalskolen ønsker ikke at have mere end to blokke med samme fag i træk.[56]

Programmet skal være i stand til at lave en mængde skemaer for flere klasser. Grundet omfanget af dette krav og den tidsmæssige begrænsning, der ligger på dette projekt, er valget truffet om at begrænse dette krav til 7. 8. og 9. klasse.

Fritimer skal ligge i slutningen af dagen og bestemt ikke i midten af dagen. Dette er gjort for ikke, at forlænge skoledagen længere end nødvendigt, ydermere ikke have længere pauser, som kan lede i brud af koncentrationen hos eleverne.

Programmet skal bruge en fil, som simulerer brugerinput. Valget er taget som et alternativ til en brugergrænseflade.

8.3 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 parallelklasserne 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.

8.4 Kodestil

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

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

structs. Herefter ligger alle anvendte funktioner som prototyper. Selve funktionerne ligger i bunden. Kommentarer skrives over det stykke kode den forklarer. Når en algoritme udføres i en funktion eller en ikke, er algoritmen rykket ind med et tab af to mellemrum. Tuborgklammer starter inden linjeskift, efter funktioner og løkker, og afsluttes efter et linjeskift. Det følgende er et eksempel på kodestillen i softwareløsningen:

```
/* Free lessons */
  int count_free_lessons = 0;
  for (j = 0; j < SCHOOL_DAYS_IN_WEEK; j++){
    for(i = 0; i < LESSONS_PER_DAY_MAX; i++){
      /* Need to be a free lesson - count op */
      if (individual_master->lesson_num[i][j] == fri){
        count_free_lessons++;
      }
```

9 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, som forklarer, hvad der kommer ind og ud af funktionerne, og i hvilken rækkefølge funktionerne bliver kaldt i.

9.1 Structs

Programmet bruger structs, så strukturen af dataene, der produceres, er lettere at håndtere. Tre structs, "individual", "teacher" og "requirements" er brugt til henholdsvis skemaerne, information om lærerne og -kravene til en klasse.

```
struct individual{
  int lesson_num[LESSONS_PER_DAY_MAX][SCHOOL_DAYS_IN_WEEK];
  int fitness;
  int perfection;
  int lessons_with_parallel;
  int lessons_with_both;
  int heavy_lesson_after;
  int heavy_lesson_before;
  int teacher_overbooked;
  int best_gena7;
  int best_gena8;
  int best_gena9;
};
```

"Individual" består af et array af arrays af integers, lesson_num, der repræsenterer lektionnummeret (faget) for hver blok på hver dag. Fitness indeholder skemaets fitnessværdi. Perfection indeholder antallet af fag, der optræder nok gange i skemaet ud fra kravene, og som ikke har mere end en blok for meget. lessons_with_parallel indeholder antallet af gange, skemaet har samme fag som en parallelklasse. lessons_with_both angiver antallet af gange, skemaet har samme fag som begge parallelklasser. heavy_lesson variabler angiver, hvor mange gange de tunge fag ligger før og efter middag. teacher_overbooked fortæller hvor mange gange, en lærer i skemaet bliver brugt i en anden parallelklasse på samme tid. best_gen variablerne angiver i hvilken generation, de bedste skemaer for hver årgang er opstået.

```
struct class_info{
  int number_of_lessons;
  char padding[2];
  char teacher_name[TEACHER_NAME_MAX];
  char lesson_name[LESSON_NAME_MAX];
  char class_name[TEACHER_NAME_MAX];
};
```

"class_info" 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 "JP".

```
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 blandt andet bruger til at tjekke om skemaet er opfyldt, eller om der mangler et specifikt fag, som skal fyldes ind i skemaet.

9.2 main

På figur 10 ses et diagram over main.

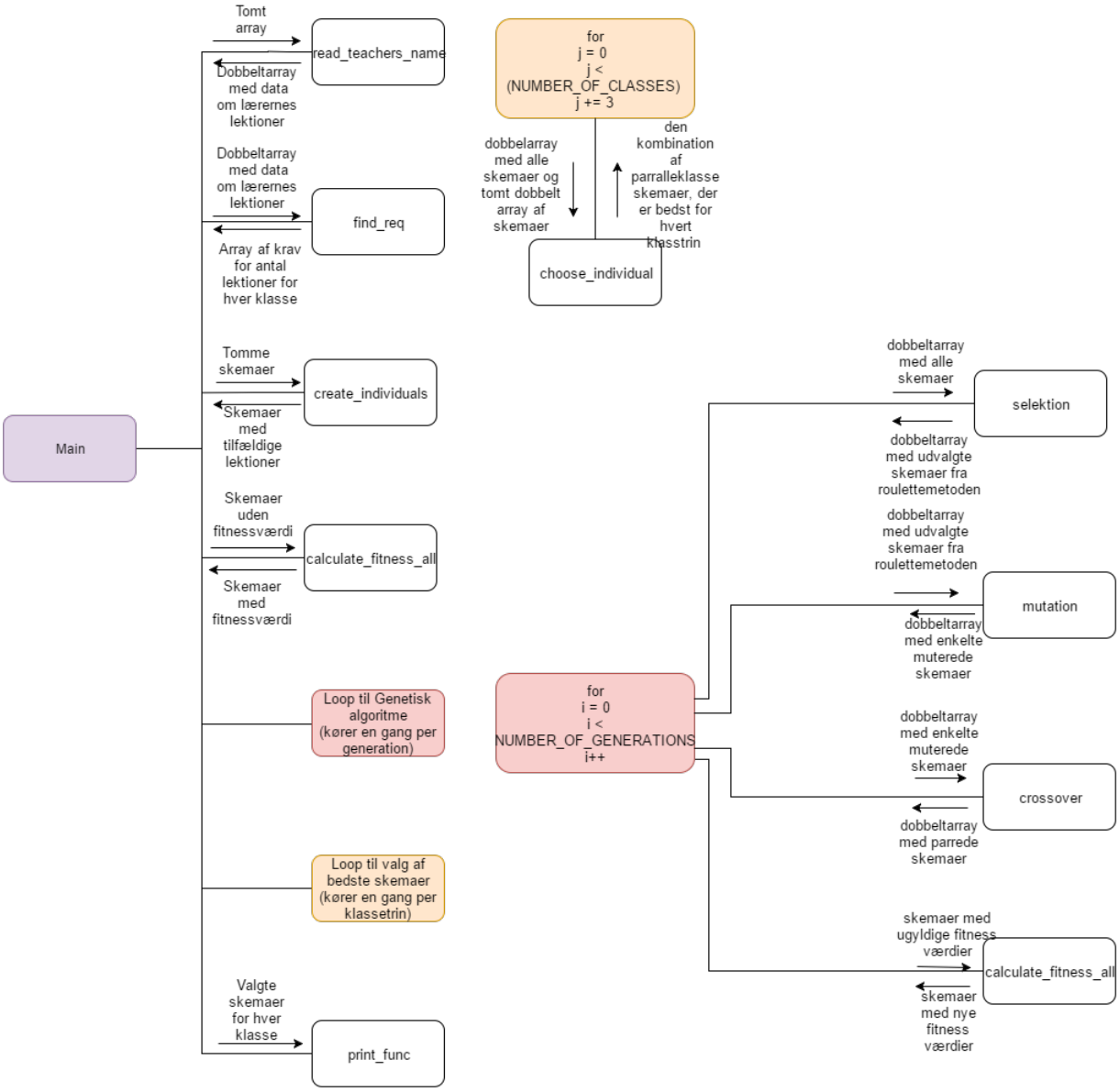


Figure 10: Grafisk diagram over main

Følgende kode er main funktionen, der bliver brugt til at danne et skema for udskolingssektoren 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){
  /*init general stuff*/
  int i, j;
  int h_classes[NUMBER_OF_HEAVY_LESSONS] = {mat, fys, eng, dan, tys};

  individual **population;
  population = (individual **)calloc(NUMBER_OF_CLASSES, sizeof(individual *));
```

```
individual **old_population;
old_population = (individual **)calloc(NUMBER_OF_CLASSES, sizeof(individual *));

individual **chosen_individual;
chosen_individual = (individual **)calloc(NUMBER_OF_CLASSES, sizeof(individual *));

individual *best_of_best;
best_of_best = (individual *)calloc(NUMBER_OF_CLASSES, sizeof(individual ));

requirements *requirements_classes;
requirements_classes = (requirements *)calloc(NUMBER_OF_CLASSES, sizeof(requirement

class_info **class_data;
class_data = (class_info **)calloc(NUMBER_OF_CLASSES, sizeof(class_info *));

for(i = 0; i < NUMBER_OF_CLASSES; i++){
  population[i] = (individual *)calloc(SIZE_OF_POPULATION +3, sizeof(individual));
  old_population[i] = (individual *)calloc(SIZE_OF_POPULATION, sizeof(individual));
  chosen_individual[i] = (individual *)calloc(NUMBER_OF_GENERATIONS, sizeof(indivi
  class_data[i] = (class_info *)calloc(NUMBER_OF_SUBJECTS, sizeof(class_info));
}

read_teachers_name(class_data);
find_req(class_data, requirements_classes);

srand(time(NULL));
create_individuals(population);
calculate_fitness_all(population, h_classes, class_data, requirements_classes);

for(i = 0; i < NUMBER_OF_GENERATIONS; i++){

  selektion(population, old_population);

  mutation(population);

  crossover(population, old_population, requirements_classes);

  calculate_fitness_all(population, h_classes, class_data, requirements_classes);

  for(j = 0; j < (NUMBER_OF_CLASSES); j += 3){
    choose_individual(population, chosen_individual, j, i);
  }

}
```

```
    i--;

  find_best(chosen_individual, best_of_best);
  /* Printing */
  printf("\n\n\n");
  print_func(best_of_best, requirements_classes, class_data);

  for(i = 0; i < NUMBER_OF_CLASSES; i++){
    free(population[i]);
    free(old_population[i]);
    free(class_data[i]);
    free(chosen_individual[i]);
  }
  free(population);
  free(old_population);
  free(chosen_individual);
  free(best_of_best);
  free(class_data);
  free(requirements_classes);

  return 0;
}
```

Først initialiseres de arrays, der bliver brugt i programmets funktioner. De arrays, som bliver dannet, er todimensionelle. Pladsen til de arrays, som bliver initialiseret allokeres dynamisk. Pladsen bliver alokeret dynamisk for at sikre, at der kun bliver allokeret den præcise mængden plads til arrays'ne. Dernæst forberedes den genetiske algoritme. Srand funktionen klargøres ved at kalde seeded til rand med antallet af sekunder siden 1. januar 1970 (time(NULL)).[57] Individerne bliver givet tilfædige værdier i lesson_num (de 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 et crossover mellem to tilfældige forældre. Efter denne process bliver skemaernes fitness igen regnet ud, inden for loopet begynder forfra.

9.3 Dataindlæsning

For at det er nemmere at lave ændringer i antallet af lærer, de timer de kan tage og det antal timer de forskellige fag skal have, er dette opstillet i en tekstfil. På figur 11 ses et grafisk diagram over funktionen. Elementerne i dokumentet står som følgende:

Lærerforkortelse fag antal timer klasse

Et eksempel på dette kunne være:

CA Dan 6 7a

JA Mat 4 7a

[57] *C library function - time().* URL: `https://www.tutorialspoint.com/c\_standard\_library/c\_function_time.htm`.

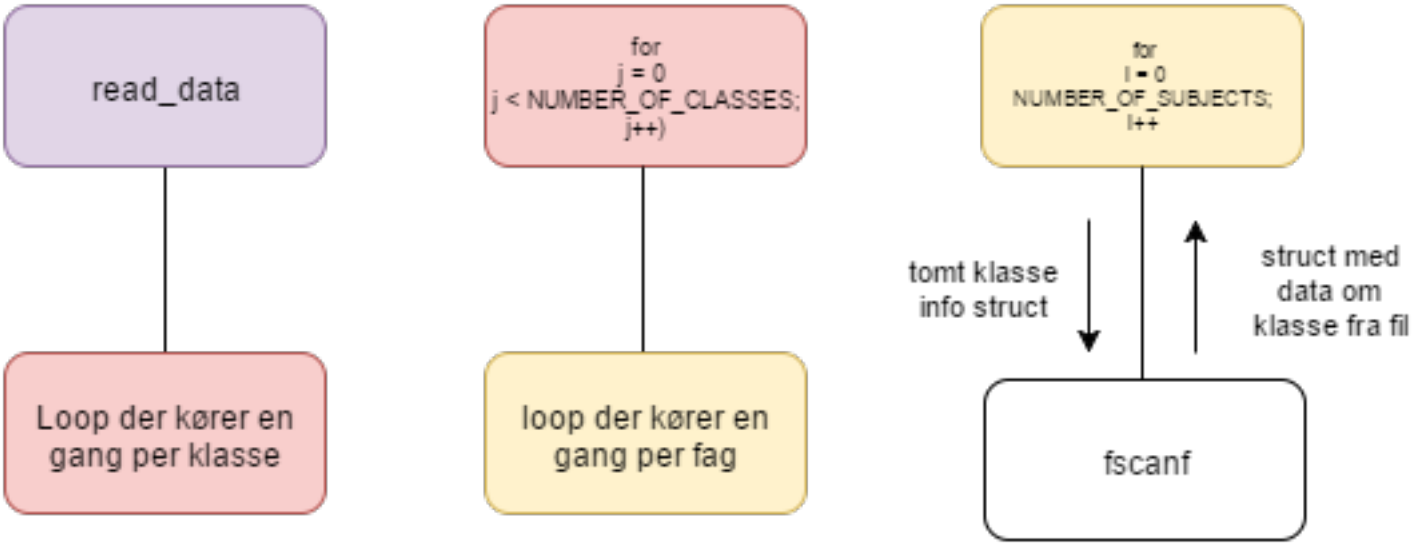


Figure 11: Grafisk diagram over read_teachers_name

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 et array af structs.

Dette er implementeret på følgende vis:

```
void read_teachers_name(class_info **class_data){
  int i, j;
  class_info local_class_data;
  FILE *teacherinfo = fopen("teacherinfo.txt", "r");
  if(teacherinfo == NULL){
    perror("Error the file is empty");
    fclose(teacherinfo);
    exit(1);
  }

  for(j = 0; j < NUMBER_OF_CLASSES; j++){
    for(i = 0;i < NUMBER_OF_SUBJECTS; i++){
      fscanf(teacherinfo,
      " %s %s %d %s ",
      local_class_data.teacher_name,
      local_class_data.lesson_name,
      &local_class_data.number_of_lessons,
      local_class_data.class_name);
      class_data[j][i] = local_class_data;
```

```
    }
  }
  fclose(teacherinfo);
}
```

Først åbnes filen, hvorefter det tjekkes om filen er tom. Hvis dette er sandt, bliver en fejl-besked printet ud, og programmet lukker igen.

Dernæst bliver der lavet local_class_data af typen 'class_info', 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_class_data. Efter dette bliver det overført til 'class_info' som er det endelige struct, med alt informationen omkring lærerne, deres dag, antal timer og deres klasse. Efter denne løkke bliver filen lukket igen.

9.4 Fitness

Figur 12 viser et diagram om fitness funktionen.

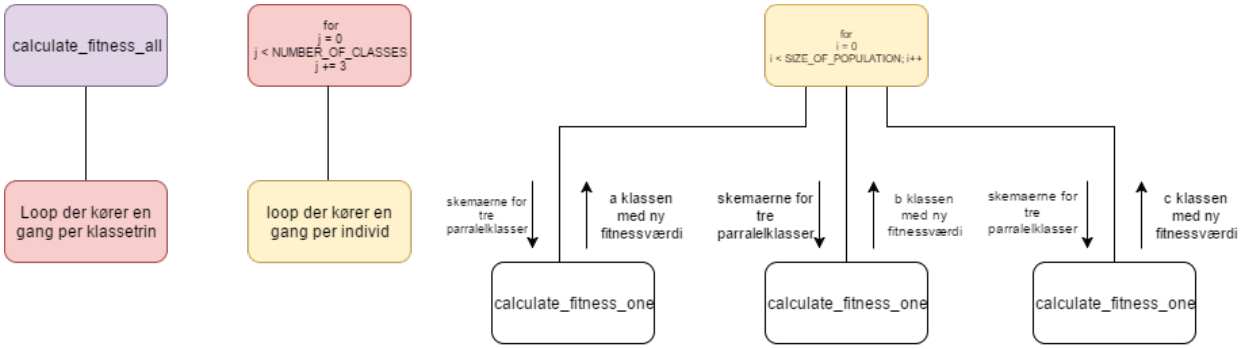


Figure 12: Grafisk diagram over funktionen for fitness

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 på kodeeksemplet herunder:

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

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_other1->lesson_num[i][j]){
        if (individual_master->lesson_num[i][j] != fri){
          individual_master->fitness += FITNESS_PARALEL_CLASS;
          individual_master->lessons_with_parallel++;
          test_parallel_both++;
        }
      }
      if (individual_master->lesson_num[i][j]
          == individual_other2->lesson_num[i][j]){
        if (individual_master->lesson_num[i][j] != fri){
          individual_master->fitness += FITNESS_PARALEL_CLASS;
          individual_master->lessons_with_parallel++;
          test_parallel_both++;
        }
      }
      if (test_parallel_both == 2){
        individual_master->lessons_with_both++;
      }
      test_parallel_both = 0;
    }
  }
```

I fitness bruges de forskellige variabler 'individual_master', 'individual_parallel1' og 'individual_parallel1'. Disse variabler er af datatypen 'individual', som er et struct med informationer omkring et skema. Funktionen 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. Derefter testes om 'master' har samme time på samme plads, som hver af parallelklasserne. Hvis dette er sandt, bliver fitness talt op på 'master' og de enkelte bindingers variabel bliver talt op i structet for individet.

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.

9.5 Selektion

Figur 13 er et diagram over selektions metoden.

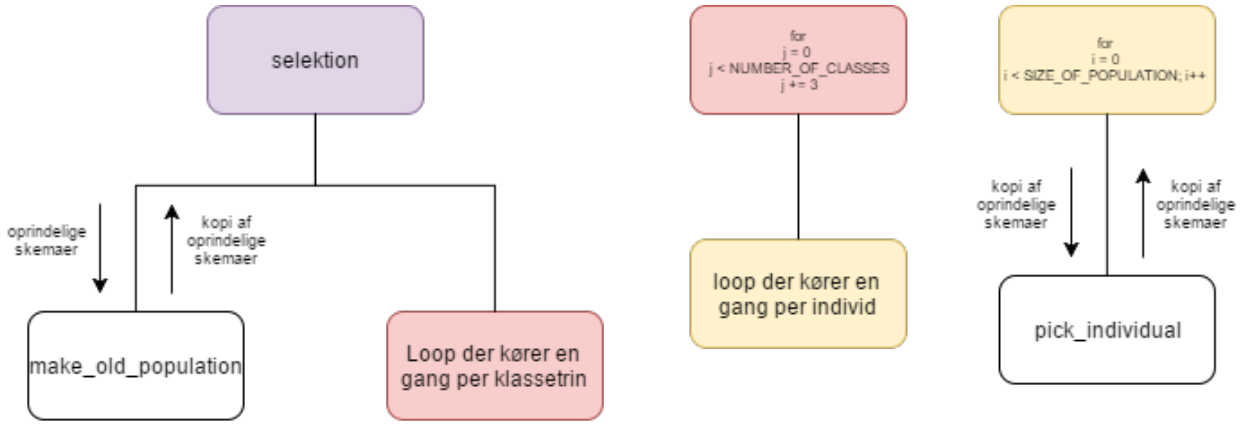


Figure 13: Grafisk diagram over funktionen for selektion

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 **population, individual **old_population,
                                    int class, int indi_num){
  int i;
  int fitness_test = 0, sum = 0;
  int *sum_parrallel;
  sum_parrallel = (int *)calloc(SIZE_OF_POPULATION, sizeof(int));

  for(i = 0; i < SIZE_OF_POPULATION; i++){
    if ((old_population[class][i].fitness == 1)
        || (old_population[class+1][i].fitness == 1)
        || (old_population[class+2][i].fitness == 1)){
      sum_parrallel[i] = 1;
    }
    else {
      sum_parrallel[i] = old_population[class][i].fitness
                                + old_population[class+1][i].fitness
                                + old_population[class+2][i].fitness;
    }
    sum += sum_parrallel[i];
  }
```

```
  int field = rand()% sum;

  for(i = 0; i < SIZE_OF_POPULATION; i++){
    fitness_test += sum_parrallel[i];
    if(field < fitness_test){
      population[class][indi_num] = old_population[class][i];
      population[class+1][indi_num] = old_population[class+1][i];
      population[class+2][indi_num] = old_population[class+2][i];
      return 1;
    }
  }
  free(sum_parrallel);
  return 0;
}
```

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 timer med en parrallelklasse, og det ville derfor være et problem, hvis skemaet blev sat sammen med skemaer fra nye parallelklasser, da fitnessen for sammenhængende timer ville være forkert.

9.6 Mutation

Det følgende er en oversigt over mutationsfunktionen:

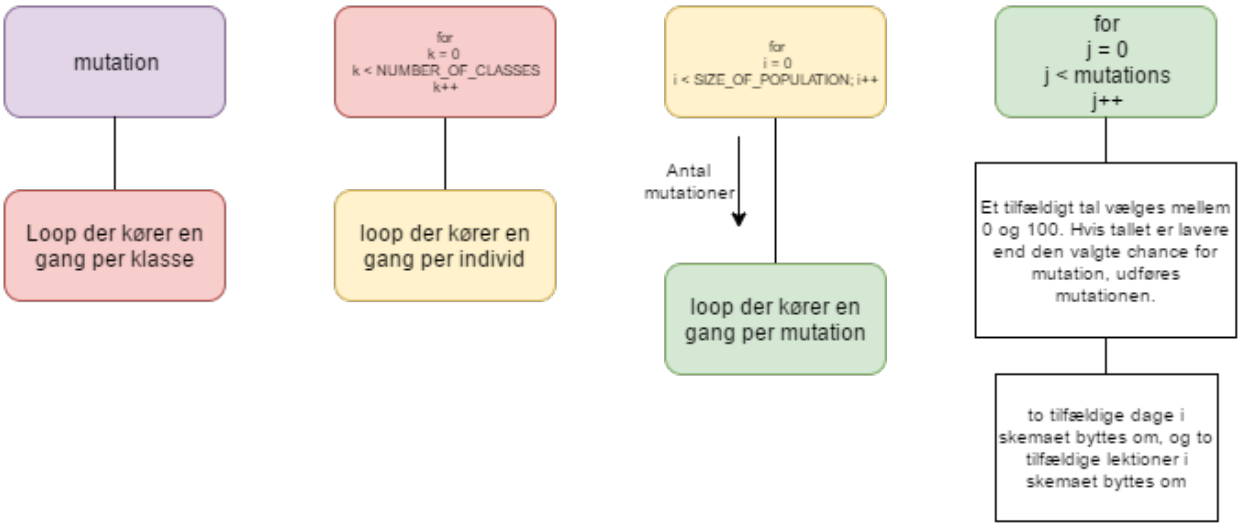


Figure 14: Grafisk diagram over funktionen for mutation

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 **population){
  int i, j, k;
  int ran1Day = 0, ran1Week = 0, ran2Day = 0,
       ran2Week = 0, chance = 0, mutations = 0, temp = 0;

  for(k = 0; k < NUMBER_OF_CLASSES; k++){
    for(i = 0; i < SIZE_OF_POPULATION; i++){
      mutations = rand()% (MAX_MUTATIONS_PER_INDIVIDUAL+1);
      for (j = 0; j < mutations; j++){
        chance = rand()% 100;
        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 = population[k][i].lesson_num[ran1Day][ran1Week];
          population[k][i].lesson_num[ran1Day][ran1Week]
          = population[j][i].lesson_num[ran2Day][ran2Week];
          population[k][i].lesson_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.

9.7 Print funktion

Figur 15 er et diagram over hvordan skemaerne printes:

I struct'et 'individual' findes et multidimensionalt integer array 'lesson_num'. Arrayet indeholder heltal, som repræsentere forskellige fag. Det er disse tal, der ændres i de forskellige funktioner.

Når et endeligt skema skal printes 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:

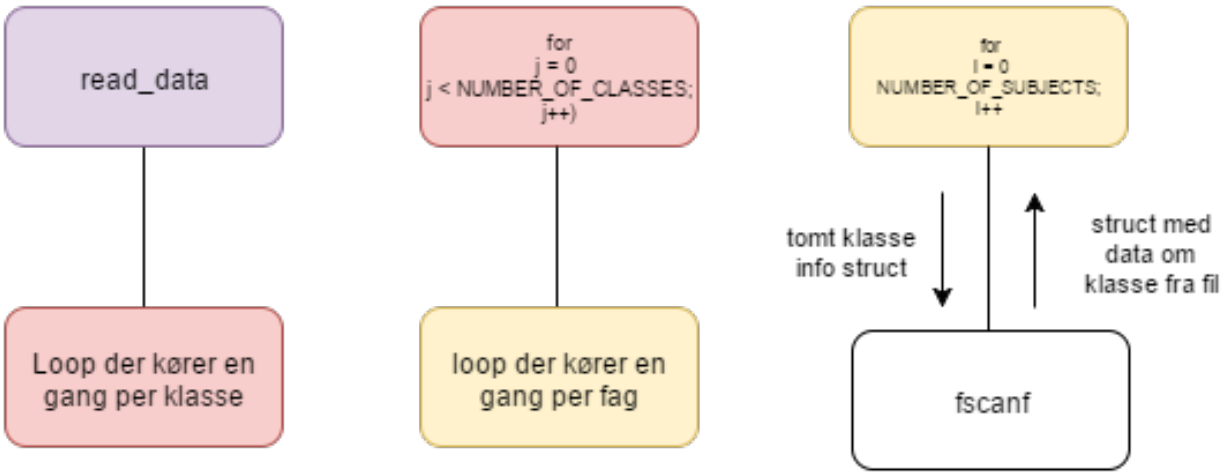


Figure 15: Grafisk diagram over funktionen for printningen af skemaerne

```
void print_func(individual best_of_best[], requirements requirements_classes[],
                        class_info **class_data){
  int i, j, c;
  printf("\n\n    teachers\n");
  for (i = 0; i < NUMBER_OF_CLASSES; i++){
    for (j = 0; j < NUMBER_OF_SUBJECTS; j++){
      printf("%s  %s %s\t", class_data[i][j].teacher_name, class_data[i][j].lesson_na
    }
    printf("\n");
  }
  printf("\n\n");

  printf("taken from      7: %d   8: %d   9: %d   \n\n", best_of_best[0].best_gena7, bes

  for (c = 0; c < NUMBER_OF_CLASSES; c++){

    print_req(best_of_best[c], requirements_classes[c]);
    printf("Has a fitness of: %d    The perfection grade is: %d "
           "Lessons with parallel: %d  Lessons With Both: %d "
           " Heavy lessons after: %d Before: %d    Overbooked: %d \n\n",
          best_of_best[c].fitness, best_of_best[c].perfection,
          best_of_best[c].lessons_with_parallel, best_of_best[c].lessons_with_both,
          best_of_best[c].heavy_lesson_after, best_of_best[c].heavy_lesson_before,
          best_of_best[c].teacher_overbooked);

    printf("   Class: ");
    print_class_name(c);
    printf("\n");
    printf("   Tidspunkt\t\tMandag\t\tTirsdag\t\tOnsdag\t\tTorsdag\t\tFredag\n");
    printf("   ----------
```

```
    for (i = 0; i < LESSONS_PER_DAY_MAX; i++){
      print_time_func(i);

      for (j = 0; j < SCHOOL_DAYS_IN_WEEK; j++){
        print_teacher_and_lesson(best_of_best[c].lesson_num[i][j], c, class_data);
        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-lykke, som gentages 'antal klasser'-gange. Klassenavnet, dens fitness og antallet af krav den overholder bliver printet i forhold til antallet af timer den skal have, . Efterfølgende køres endnu en for-lykke, som gentages 'antallet af lektioner'-gange. Dette printer tidspunktet på dagen ud, hvorefter endnu en for-lykke printer de første lektioner af hver dag ud, samt deres lærer. Efter dette bliver der printet nye linjer ud, som starter de næste timer. Efter hver anden dag der bliver printet printes en ekstra linje ud, som tydeliggøre hvor der er pauser på skemaet.

10 Program test

10.1 Test af initierende skema generering

Figur 16 viser at programmet starter med at fylde alle blokke med tilfældige skemaer, printes her skemaerne for syvende klasse for tre parallelklasser lige efter create_individual funktionen.

```
Must have:      6 4 3 3 2 2 2 2 2 2 2 0 2 0
Have:           2 5 3 3 1 2 5 3 0 4 3 2 2 5
Has a fitness of: 1   The perfection grade is: 7  Lessons with parallel: 6  Lessons With Both: 0  Heavy lessons after: 6 Before: 8   Overbooked: 1

 Class: 7.A
 Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
 ------------------------------------------------------------------------------------------
  8.00 -  8.45  |      AY  Bio         AY  Gym         AD  Mat         AY  Bio         AT  Eng
  8.45 -  9.30  |        ---           AD  Sam         AT  Val         AA  Dan         AT  Tys

  9.50 - 10.35  |      AZ  Fys         AD  Mat         AT  Tys         AP  Rel         AD  Sam
 10.35 - 11.20  |      AH  His         AY  Gym         AT  Tys           ---           AY  Gym

 11.50 - 12.35  |        ---           AD  Mat         AD  Mat         AW  Pra         AY  Bio
 12.35 - 13.20  |      AH  His         AD  Sam         AT  Val         AT  Eng         AP  Rel

 13.30 - 14.15  |        ---           AD  Mat         AT  Eng         AW  Pra         AD  Sam
 14.15 - 15.00  |      AY  Bio         AT  Val         AA  Dan           ---           AD  Sam



Must have:      6 4 3 3 2 2 2 2 2 2 2 0 2 0
Have:           3 5 1 3 2 5 3 2 2 4 3 4 2 1
Has a fitness of: 1   The perfection grade is: 8  Lessons with parallel: 7  Lessons With Both: 1  Heavy lessons after: 6 Before: 8   Overbooked: 4

 Class: 7.B
 Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
 ------------------------------------------------------------------------------------------
  8.00 -  8.45  |        ---           AL  Bio         AZ  Eng         AN  Tys         AM  Mat
  8.45 -  9.30  |      AX  Gym         AM  Mat         AP  Rel         AL  Bio         AM  Mat

  9.50 - 10.35  |      AL  Fys         AG  Val         AN  Tys         AO  Dan         AL  Bio
 10.35 - 11.20  |      AC  Sam         AO  Pra         AR  Geo         AH  His         AO  Pra

 11.50 - 12.35  |      AL  Fys         AO  Dan         AH  His         AP  Rel         AH  His
 12.35 - 13.20  |      AN  Tys         AP  Rel         AC  Sam         AH  His         AP  Rel

 13.30 - 14.15  |      AL  Bio         AH  His         AR  Geo         AG  Val         AO  Dan
 14.15 - 15.00  |      AX  Gym         AM  Mat         AM  Mat         AC  Sam         AX  Gym



Must have:      6 4 3 3 2 2 2 2 2 2 2 0 2 0
Have:           1 3 5 3 0 4 1 2 5 3 4 6 1 2
Has a fitness of: 1   The perfection grade is: 3  Lessons with parallel: 5  Lessons With Both: 0  Heavy lessons after: 5 Before: 7   Overbooked: 2

 Class: 7.C
 Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
 ------------------------------------------------------------------------------------------
  8.00 -  8.45  |      AN  Tys         AN  Tys         AR  Geo         AP  Rel         AR  Geo
  8.45 -  9.30  |      AR  Geo         AP  Rel         AY  Eng         AP  Rel           ---

  9.50 - 10.35  |      AA  Gym         AW  Mat         AH  His         AY  Eng         AA  Gym
 10.35 - 11.20  |      AA  Gym         AV  Val         AC  Dan         AR  Geo         AW  Mat

 11.50 - 12.35  |      AV  Val         AR  Geo         AH  His         AH  His         AH  His
 12.35 - 13.20  |      AJ  Bio         AY  Eng         AP  Rel           ---           AY  Eng

 13.30 - 14.15  |      AY  Eng         AW  Mat         AJ  Bio         AA  Gym         AX  Sam
 14.15 - 15.00  |      AL  Pra         AN  Tys         AP  Rel         AJ  Bio         AP  Rel
```

Figure 16: Genereret udkast af skemaer til 7.a, b og c

10.2 Test af fitness funktion

Skemaernes fitness er beregnet ud fra værdierne vist på figur 17.

De forskellige fitnessparametre kan godt give negative eller give positive værdier mere end en gang. Det vil sige at, hvis der er to steder på skemaet, hvor der er en lærer der er overbooked giver det -2000 i fitness to gange. I denne test, undersøges det om fitnessen udregnes korrekt. De nedenstående skemaer er generet gennem softwareløsningen. De tre skemaer er alle skemaer for 7.C, de skal altså derfor alle have det samme antal af timer og de samme fag. I det første skema mangler 7.C to dansk lektioner, det kan ses på at perfection graden er på 12, hvis skemaet havde opfyldt alle krav om timeantal, ville perfection graden have været 13. Udover at der mangler to dansk lektioner har dette skema også for mange af nogle lektioner. Dette kan observeres i Must have og Have, hvor fagene står i følgende rækkefølge dansk, matematik, engelsk, tysk, fysik/kemi,

```
#define FITNESS_LESSONS_IN_ROW 80
#define FITNESS_PARALEL_CLASS 100
#define FITNESS_HEAVY_LESSONS -200
#define FITNESS_HEAVY_LESSONS_BEFORE 200
#define FITNESS_FREE_IN_MIDLE -100000
#define FITNESS_MANY_LESSONS_IN_ROW -500
#define FITNESS_TEACHER_OVERBOOKED -2000
#define FITNESS_TEACHER_PREPARATION 60
#define FITNESS_NOT_MEET_REQ -60
#define FITNESS_WAY_OVER_REQ -8000
#define FITNESS_CORRECT_LESSONS 100
#define FITNESS_BONUS_FREE_END 70
#define FITNESS_PERFECTION_BONUS 500
#define FITNESS_NO_FREE_TIME -100
#define FITNESS_NOT_OVERBOOKED 50
```

Figure 17: Billede over de fitness værdier skemaerne bliver tildelt, i tilfælde af at de opfylder visse parametre

historie, samfundsfag, valgfag, geografi, biologi, idræt, kristendom, praktiske fag og fri timer til sidst. I det første skema er der fem fag, hvor der er for mange lektioner, navnlig engelsk, hvor der er to lektioner for meget, tysk hvor der er en lektion for meget, historie hvor der er en for meget, samfundsfag hvor der er en for meget og biologi hvor der er en for meget. Dette har negativ indflydelse på skemaet. Skemaets positive fitness er en kombination af lessons with parallel, som er det antal af lektioner der foregår på samme tid som en af parallelklasserne og lessons with both, hvor der er mulighed for samarbejde mellem alle tre parallelklasser på en gang. Dette skema har otte lektioner, hvor det er muligt at arbejde sammen med en a parallelklasserne. Dette skema har dog ingen lektioner på sammen tid med begge parallelklasser. Derudover får dette skema mindre i fitness, ved at det har 'tunge' fag som ligger over middag. I dette skema ligger er der otte tunge fag, som ligger efter middag. Derudover er lærerne også overbooked fire gange i dette skema, det vil sige at der er fire tilfælde, hvor en af klassens lærer, har en eller flere lektioner på samme tid. Skemaets samlede fitness er 5561.

Det andet skema har en perfction grade på 13, alle kravene for antallet af lektioner for de forskellige fag er altså opfyldt. Der er dog tre fag hvor der er en lektion for meget, fagene for disse lektioner er samfundsfag, hvor der er en lektion for meget, geografi, hvor der er en lektion for meget og praktiske fag, hvor der er to lektioner for meget. I skemaet er der tre fælles lektioner med parallel klasserne, der er dog ingen lektioner, hvor der er mulighed for samarbejde med begge parallelklasser på samme tid. På skemaet er der syv lektioner med tunge fag over middag. Der er to steder, hvor en lærer er overbookedt. Dette skemas samlede fitness er 10341.

```
Must have:       6 4 3 3 2 2 2 2 2 2 2 0 2 0
Have:            4 4 5 4 2 3 3 2 2 3 2 0 2 4
Has a fitness of: 5561   The perfection grade is: 12  Lessons with parallel: 8  Lessons With Both: 0  Heavy lessons after: 8 Before: 11   Overbooked: 4

 Class: 7.C
 Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
 ------------------------------------------------------------------------------------------
  8.00 -  8.45  |      AA  Gym         AY  Eng         AY  Eng         AR  Geo         AR  Geo
  8.45 -  9.30  |      AJ  Bio         AN  Tys         AL  Pra         AY  Eng         AV  Val

  9.50 - 10.35  |      AY  Eng         AY  Eng         AH  His         AL  Pra         AW  Mat
 10.35 - 11.20  |      AN  Tys         AN  Tys         AV  Val         AW  Mat         AG  Fys

 11.50 - 12.35  |      AA  Gym         AH  His         AH  His         AJ  Bio         AG  Fys
 12.35 - 13.20  |      AC  Dan         AX  Sam         AX  Sam         AN  Tys         AC  Dan

 13.30 - 14.15  |      AX  Sam         AJ  Bio         AW  Mat           ---           AC  Dan
 14.15 - 15.00  |      AW  Mat           ---             ---             ---           AC  Dan
```

Figure 18: Billede af et generet skema

```
Must have:       6 4 3 3 2 2 2 2 2 2 2 0 2 0
Have:            6 4 3 3 2 2 3 2 3 2 2 0 4 4
Has a fitness of: 10341   The perfection grade is: 13  Lessons with parallel: 3  Lessons With Both: 0  Heavy lessons after: 7 Before: 11   Overbooked: 2

 Class: 7.C
 Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
 ------------------------------------------------------------------------------------------
  8.00 -  8.45  |      AN  Tys         AA  Gym         AN  Tys         AA  Gym         AG  Fys
  8.45 -  9.30  |      AW  Mat         AC  Dan         AW  Mat         AC  Dan         AV  Val

  9.50 - 10.35  |      AJ  Bio         AX  Sam         AJ  Bio         AH  His         AY  Eng
 10.35 - 11.20  |      AN  Tys         AV  Val         AY  Eng         AY  Eng         AH  His

 11.50 - 12.35  |      AG  Fys         AC  Dan         AX  Sam         AX  Sam         AW  Mat
 12.35 - 13.20  |      AC  Dan         AC  Dan         AR  Geo         AR  Geo         AW  Mat

 13.30 - 14.15  |        ---           AC  Dan         AL  Pra         AL  Pra           ---
 14.15 - 15.00  |        ---           AR  Geo         AL  Pra         AL  Pra           ---
```

Figure 19: Billede af et generet skema

Perfection graden er 13 i det tredje skema, der er altså ikke for lidt lektioner af nogle fag. Der er dog tre fag hvor der er for mange lektioner. Disse fag er dansk hvor der er en lektion for meget, samfundsfag hvor der er en lektion for meget og praktiske fag hvor der er en for meget. På dette skema er der fire lektioner, hvor der er mulighed for at samarbejde med en parallelklasse i otte af skemaets lektioner, der er en lektion, hvor der er mulighed for at samarbejde med begge parallelklasser sammentidigt. På det tredje skema er der fem gange, hvor et tungt fag ligger over middag, dette er det mindste antal ud af de tre skemaer. Det forekommer ingen gang at lærerne i det tredje skema har to planlagte skemaer på samme tid.

```
Must have:       6 4 3 3 2 2 2 2 2 2 2 0 2 0
Have:            7 4 3 3 2 2 3 2 2 2 2 0 3 5
Has a fitness of: 15851   The perfection grade is: 13  Lessons with parallel: 8  Lessons With Both: 1  Heavy lessons after: 5 Before: 14   Overbooked: 0

 Class: 7.C
 Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
 ------------------------------------------------------------------------------------------
  8.00 -  8.45  |      AX  Sam         AX  Sam         AX  Sam         AW  Mat         AG  Fys
  8.45 -  9.30  |      AC  Dan         AC  Dan         AA  Gym         AY  Eng         AY  Eng

  9.50 - 10.35  |      AW  Mat         AW  Mat         AC  Dan         AN  Tys         AJ  Bio
 10.35 - 11.20  |      AL  Pra         AC  Dan         AC  Dan         AN  Tys         AW  Mat

 11.50 - 12.35  |      AH  His         AR  Geo         AC  Dan         AY  Eng         AJ  Bio
 12.35 - 13.20  |      AL  Pra         AV  Val         AG  Fys         AN  Tys         AH  His

 13.30 - 14.15  |      AC  Dan         AL  Pra         AR  Geo           ---             ---
 14.15 - 15.00  |        ---           AA  Gym         AV  Val           ---             ---
```

Figure 20: Billede af et generet skema

På tabellen med fitnessværdier, kan man se at fitnessværdien bliver trukket meget ned af have lærere der er 'overbooked,' dette passer med at det tredje skema er det bedste da det ikke har nogle lærere der er 'overbooked'. Derudover er det også den tredje der har mindst tunge lektioner over middag, hvilket også trækker de andre to skemaer ned. Det sidste skema er også det eneste skema, hvor der er mulighed for samarbejde mellem alle parallelklasserne sammentidigt. Ud fra tabellen

med fitnessværdierne og undersøgelsen af de tre skemaer, kan det konkluderes at fitnessfunktionen fungerer som den skal.

10.3 Test mutations funktion

For at teste om mutationen fungerer korrekt, printes i dette eksempel et skema før og efter en mutation, samt de tilfældige dage og lektioner, der blev valgt. I figur 21 ses det, at de tilfældige lektioner blev 2 og 4, og de tilfældige dage blev 3 og 0. Da der tælles fra 0, svarer det til den 3. lektion på 4. dag, der skal byttes med den 5. lektion på 1. dag.

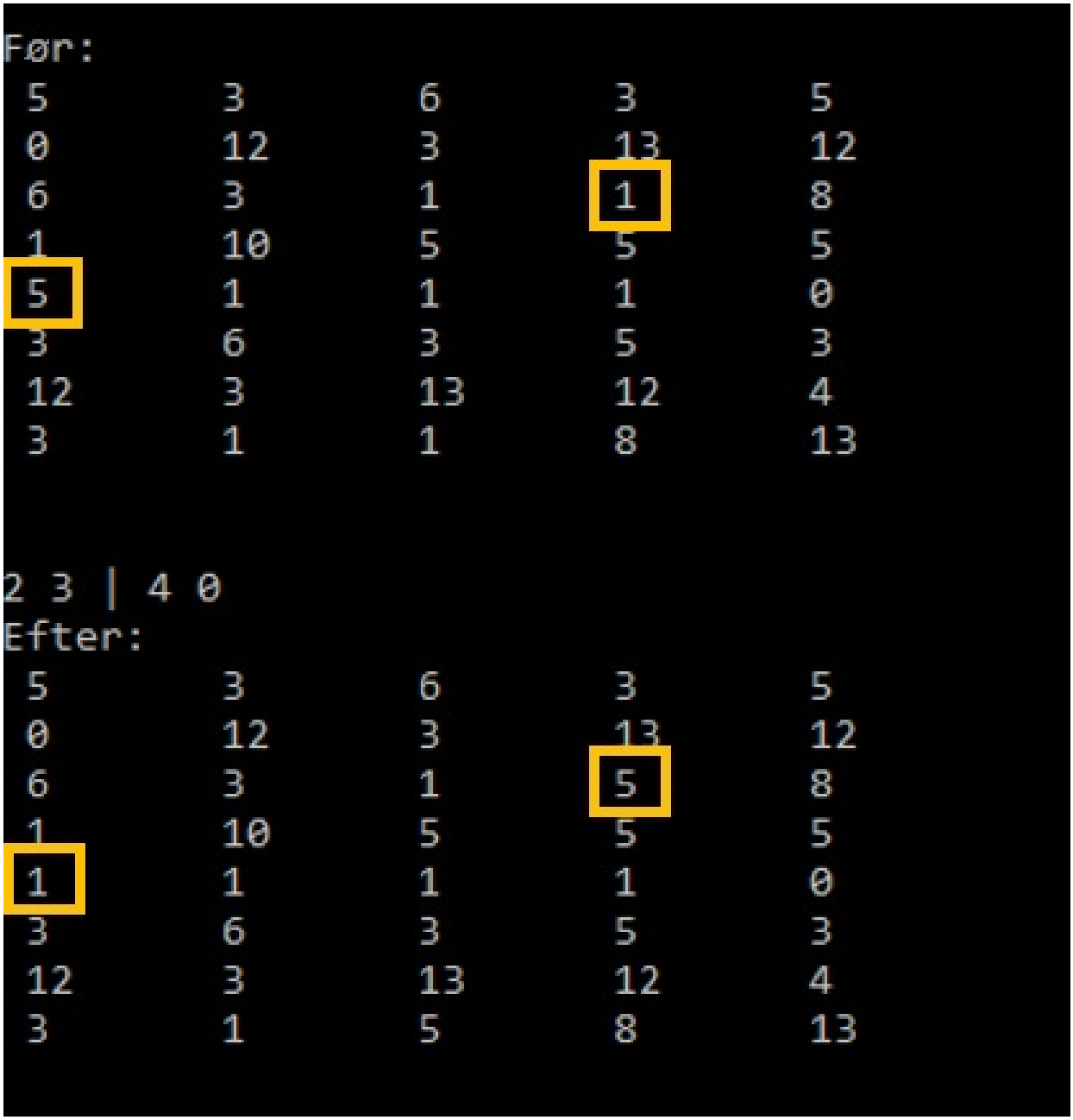


Figure 21: Billede af et skema før og efter mutation.

På figur 21 ses det at netop de nævnte lektioner er blevet flyttet, så mutationen fungerer som den skal.

10.4 Test crossover funktion

På figur 22 er printet to forældreskemaer for et afkomskema med dagene vandret og blokkene lodret. Her ses det, at afkomskemaet har taget sin første dag fra første forældres anden dag, sin anden dag fra anden forældre sidste dag. Tredje dag er første del af fjerde dag fra anden forældre og sidste del af dag et fra første forældre. Fjerde dag foregår på samme måde og femte dag er så de manglende lektioner, der bliver indsat, indtil kravene er nået. Herefter bliver resten af lektionerne sat til at være 13 (fri).

```
skema forældre 1
3  4  0  6  1  5  1  10
10  7  7  9  2  2  13  13
3  9  1  12  4  1  13  13
7  0  0  8  8  2  2  13
12  5  3  0  0  0  13  13


Skema forældre 2
0  3  3  0  0  4  4  13
 8  4  7  0  7  4  0  0
 5  10  5  1  5  1  8  13
 10  0  0  0  7  4  0  0
 3  9  9  12  12  1  2  1


Det nye skema
10  7  7  9  2  2  13  13
 3  9  9  12  12  1  2  1
 10  0  0  6  1  5  1  10
 7  0  0  0  0  4  4  13
 5  3  3  8  8  13  13  13
```

Figure 22: to parents og et child

10.5 Helheds test

I denne test køres programmet med de valgte værdier på symbolske konstanter, som ses på figur 23 og indholdet i datafilen som ses figur 24.

På figur 25 ses den valgte kombination af skemaer for det niende klassetrin, samt antallet af lektioner, der er kravet for hvert fag, "Must have", hvor mange gange, de har hvert fag, "Have", skemaets fitnessniveau og hvor mange fag skemaet har nok af "perfectiongrade". På sidste linje, ses det hvor mange gange skemaet har fag på samme tid som en parallelklasse, "lessons with

```
#define FITNESS_LESSONS_IN_ROW 80
#define FITNESS_PARALEL_CLASS 100
#define FITNESS_HEAVY_LESSONS -200
#define FITNESS_HEAVY_LESSONS_BEFORE 200
#define FITNESS_FREE_IN_MIDLE -100000
#define FITNESS_MANY_LESSONS_IN_ROW -500
#define FITNESS_TEACHER_OVERBOOKED -2000
#define FITNESS_TEACHER_PREPARATION 60
#define FITNESS_NOT_MEET_REQ -60
#define FITNESS_WAY_OVER_REQ -8000
#define FITNESS_CORRECT_LESSONS 100
#define FITNESS_BONUS_FREE_END 70
#define FITNESS_PERFECTION_BONUS 500
#define FITNESS_NO_FREE_TIME -100
#define FITNESS_NOT_OVERBOOKED 50
```

Figure 23: Eksempel på programmets værdier af symbolske konstanter.

```
AA Dan 6 7a          CB Dan 6 8a          CR Dan 6 9a
AD Mat 4 7a          JD Mat 4 8a          JR Mat 4 9a
AT Eng 3 7a          HC Eng 3 8a          HR Eng 3 9a
AT Tys 3 7a          RC Tys 3 8a          RR Tys 3 9a
AZ Fys 2 7a          MD Fys 2 8a          MF Fys 3 9a
AH His 2 7a          MD His 2 8a          ME His 1 9a
AD Sam 2 7a          KD Sam 2 8a          KF Sam 2 9a
AT Val 2 7a          UD Val 2 8a          UR Val 2 9a
AM Geo 2 7a          RD Geo 1 8a          RR Geo 1 9a
AY Bio 2 7a          MD Bio 2 8a          MR Bio 1 9a
AY Gym 2 7a          WD Gym 2 8a          WR Gym 2 9a
AP Rel 0 7a          DD Rel 1 8a          DR Rel 1 9a
AW Pra 2 7a          UD Pra 0 8a          UR Pra 0 9a
AO Dan 6 7b          CB Dan 6 8b          CS Dan 6 9b
AM Mat 4 7b          JA Mat 4 8b          JS Mat 4 9b
AZ Eng 3 7b          HA Eng 3 8b          HS Eng 3 9b
AN Tys 3 7b          RB Tys 3 8b          RS Tys 3 9b
AL Fys 2 7b          MB Fys 2 8b          MS Fys 3 9b
AH His 2 7b          ME His 2 8b          MS His 1 9b
AC Sam 2 7b          KE Sam 2 8b          KS Sam 2 9b
AG Val 2 7b          UE Val 2 8b          US Val 2 9b
AR Geo 2 7b          RE Geo 1 8b          RS Geo 1 9b
AL Bio 2 7b          ME Bio 2 8b          MS Bio 1 9b
AX Gym 2 7b          WE Gym 2 8b          WS Gym 2 9b
AP Rel 0 7b          DE Rel 1 8b          DS Rel 1 9b
AO Pra 2 7b          UE Pra 0 8b          US Pra 0 9b
AC Dan 6 7c          CF Dan 6 8c          CA Dan 6 9c
AW Mat 4 7c          JF Mat 4 8c          JB Mat 4 9c
AY Eng 3 7c          HF Eng 3 8c          HC Eng 3 9c
AN Tys 3 7c          RB Tys 3 8c          RT Tys 3 9c
AG Fys 2 7c          MB Fys 2 8c          MT Fys 3 9c
AH His 2 7c          MF His 2 8c          MT His 1 9c
AX Sam 2 7c          KF Sam 2 8c          KT Sam 2 9c
AV Val 2 7c          UF Val 2 8c          UC Val 2 9c
AR Geo 2 7c          RF Geo 1 8c          RF Geo 1 9c
AJ Bio 2 7c          MF Bio 2 8c          ME Bio 1 9c
AA Gym 2 7c          WF Gym 2 8c          WB Gym 2 9c
AP Rel 0 7c          DF Rel 1 8c          DT Rel 1 9c
AL Pra 2 7c          UF Pra 0 8c          UT Pra 0 9c
```

Figure 24: Eksempel på datafil.

parallel", og med begge parallelklasser, "lessons with both". Hvor mange gange, der er tunge fag efter, "Heavy lessons after", og før middag, og sidst hvor mange gange skemaet har lektioner, hvor

læreren er lektioner i andre klasser.

```
Must have:       6 4 3 3 3 1 2 2 1 1 2 1 0 0
Have:            6 4 4 4 4 1 2 2 1 1 2 2 0 7
Has a fitness of: 15831   The perfection grade is: 13
Lessons with parallel: 10  Lessons With Both: 1  Heavy lessons after: 7 Before: 15   Overbooked: 0

  Class: 9.A
  Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
  ------------------------------------------------------------------------------------------------
   8.00 -  8.45  |     MF  Fys         MF  Fys         MF  Fys         MF  Fys         JR  Mat
   8.45 -  9.30  |     RR  Tys         RR  Tys         HR  Eng         HR  Eng         CR  Dan

   9.50 - 10.35  |     KF  Sam         KF  Sam         HR  Eng         HR  Eng         RR  Geo
  10.35 - 11.20  |     DR  Rel         DR  Rel         JR  Mat         JR  Mat         JR  Mat

  11.50 - 12.35  |     CR  Dan         CR  Dan         UR  Val         UR  Val         CR  Dan
  12.35 - 13.20  |     WR  Gym         WR  Gym         RR  Tys         RR  Tys         MR  Bio

  13.30 - 14.15  |       ---             ---           CR  Dan         CR  Dan         ME  His
  14.15 - 15.00  |       ---             ---             ---             ---             ---



Must have:       6 4 3 3 3 1 2 2 1 1 2 1 0 0
Have:            6 4 4 3 3 2 4 2 1 2 2 2 0 5
Has a fitness of: 16271   The perfection grade is: 13
Lessons with parallel: 11  Lessons With Both: 1  Heavy lessons after: 6 Before: 14   Overbooked: 0

  Class: 9.B
  Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
  ------------------------------------------------------------------------------------------------
   8.00 -  8.45  |     MS  Bio         MS  Fys         MS  Bio         HS  Eng         RS  Tys
   8.45 -  9.30  |     HS  Eng         WS  Gym         HS  Eng         HS  Eng         WS  Gym

   9.50 - 10.35  |     RS  Tys         CS  Dan         RS  Tys         CS  Dan         MS  Fys
  10.35 - 11.20  |     DS  Rel         JS  Mat         DS  Rel         JS  Mat         MS  Fys

  11.50 - 12.35  |     KS  Sam         CS  Dan         KS  Sam         CS  Dan         RS  Geo
  12.35 - 13.20  |     KS  Sam         CS  Dan         KS  Sam         CS  Dan         JS  Mat

  13.30 - 14.15  |     US  Val           ---           MS  His         MS  His         JS  Mat
  14.15 - 15.00  |     US  Val           ---             ---             ---             ---



Must have:       6 4 3 3 3 1 2 2 1 1 2 1 0 0
Have:            6 4 3 3 3 1 4 2 2 1 2 1 0 8
Has a fitness of: 15941   The perfection grade is: 13
Lessons with parallel: 10  Lessons With Both: 1  Heavy lessons after: 6 Before: 13   Overbooked: 0

  Class: 9.C
  Tidspunkt            Mandag          Tirsdag         Onsdag          Torsdag         Fredag
  ------------------------------------------------------------------------------------------------
   8.00 -  8.45  |     RF  Geo         RF  Geo         HC  Eng         WB  Gym         UC  Val
   8.45 -  9.30  |     CA  Dan         CA  Dan         CA  Dan         UC  Val         HC  Eng

   9.50 - 10.35  |     CA  Dan         CA  Dan         WB  Gym         RT  Tys         DT  Rel
  10.35 - 11.20  |     MT  Fys         JB  Mat         CA  Dan         JB  Mat         HC  Eng

  11.50 - 12.35  |     KT  Sam         KT  Sam         KT  Sam         KT  Sam         JB  Mat
  12.35 - 13.20  |     MT  Fys         MT  Fys         RT  Tys         RT  Tys         JB  Mat

  13.30 - 14.15  |       ---             ---             ---             ---         MT  His
  14.15 - 15.00  |       ---             ---             ---             ---         ME  Bio
```

Figure 25:

11 Videreudvikling

Grundet manglene tid var nogle ting ikke implementeret i løsningen. I det følgende afsnit vil disse mangler blive pointeret.

11.1 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 udskolingen. Der mangler derfor at blive genereret skemaer for klasserne i indskolingen og mellemskolingen. For at kunne genere skemaer for indskolingen og mellemskolingen, skal der indsættes nye 'requirements' for disse klassetrin.

11.2 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 datafilen. 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 brugervenlighed undersøges, 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 Sofiendalskolens præferencer og ville derfor ikke kunne bruges på en skole, hvor de vægter f.eks. senere mødetimer 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 f.eks. 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 udskolingen med tre parrelelklasser. Programmet kan delvist tilpasses til nye krav ved ændring af defines i kildekoden. Ved videreudvikling vil denne del gøres mere 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 at genere et nyt skema, hver gang programmet køres. I det aktuelle program er brugeren nødt til at gemme en kopi af det genererede 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.

11.3 Brugertest

Hvis en GUI bliver implementeret i programmet, ville det også have været relevant at foretage brugertests. En brugertest ville forsikre at GUI'ens interface er brugervenligt og overskueligt. I interviewet, der blev foretaget med Sofiendalskolen, 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.[58] Det er derfor vigtigt at programmet er brugervenligt, da skolerne ellers ikke vil fortsætte med at bruge produktet.

11.4 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 Sofiendalskolens 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 Sofiendalskolens bindinger, kan løsningen ikke bruges optimalt på andre folkeskoler. 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 ud fra en valgt kommune af brugeren, sørge for hvert år automatisk at opdatere kravene til timeantal osv. udfra kommunernes og undervisningsministeriets krav.

I GUI'en kunne det også designes sådan, at 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.

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

12 Diskussion

Ud fra test afsnittet kan det konkluderes at alle de testede funktioner virker, som de skal. Den genetiske algoritme er dog ikke optimal, dette kan observeres på figur 26.

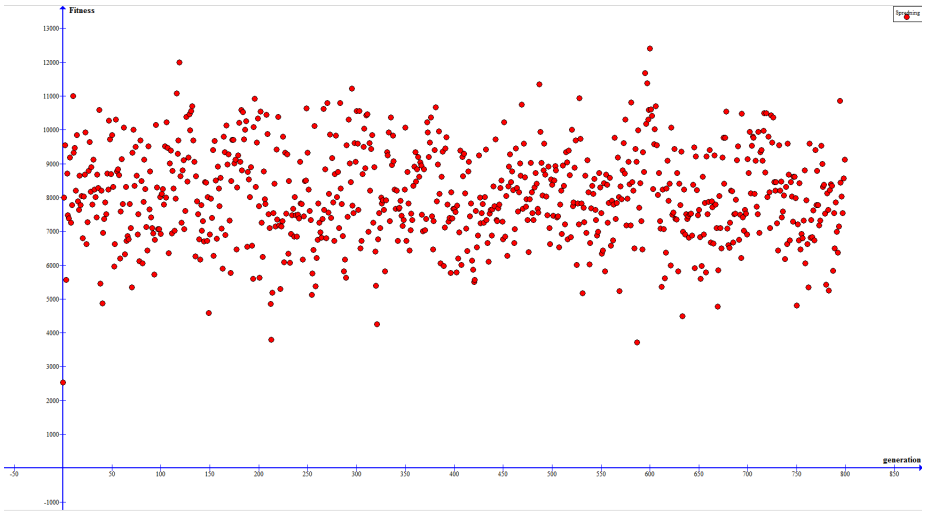


Figure 26: Graf over spredningen af fitness værdier

Fitnessen er tilfældig i gennem programmet. Skemaerne bliver altså ikke bedre efter hver generation. De bliver hellere ikke dårligere, men bliver ved med at blive dannet tilfældige skemaer. Det kan skyldes, at mutationen i algoritmen er for kraftig og sker for hyppigt. Programmet er dog blevet testet med flere forskellige sandsynligheder for at mutationen opstår. Ud fra disse tests kunne det ses at sandsynligheden for at en mutation finder sted, ikke havde nogen synlig effekt på spredningsdiagrammet. Dette kan observeres på figur 27.

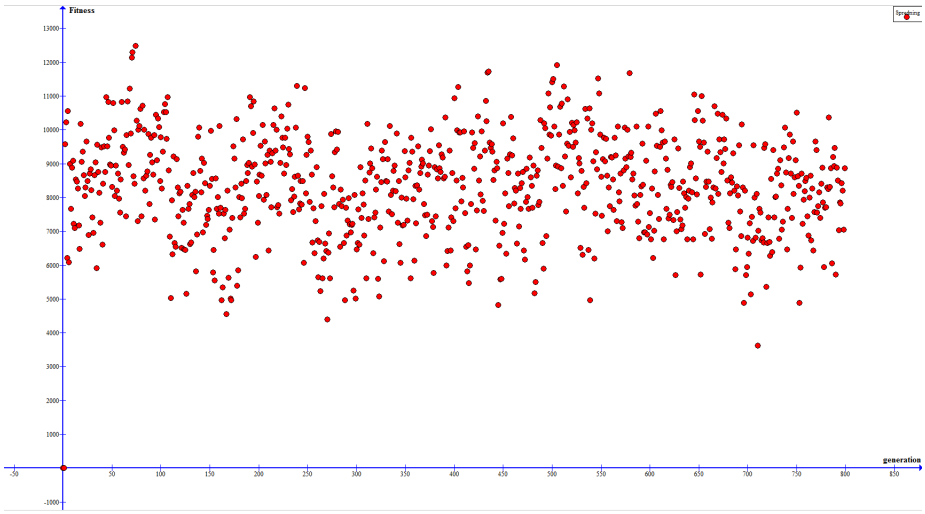


Figure 27: Graf over spredningen af fitness værdier uden mutations funktionen

Det kunne også være, at crossoveren gør skemaerne for tilfældige. På figur 28 ses spredningsdiagrammet da programmet blev kørt igennem en gang uden crossoveren. Der er altså en mere tydelig tendens i grafen, når crossoveren bliver fjernet, men kurven stiger stadig ikke konsekvent i de senere generation. Spredningen er dog mindre tilfældigt. Ud fra denne test kunne crossoverfunktionen godt være problematisk, ved at danne for tilfældige skemaer.

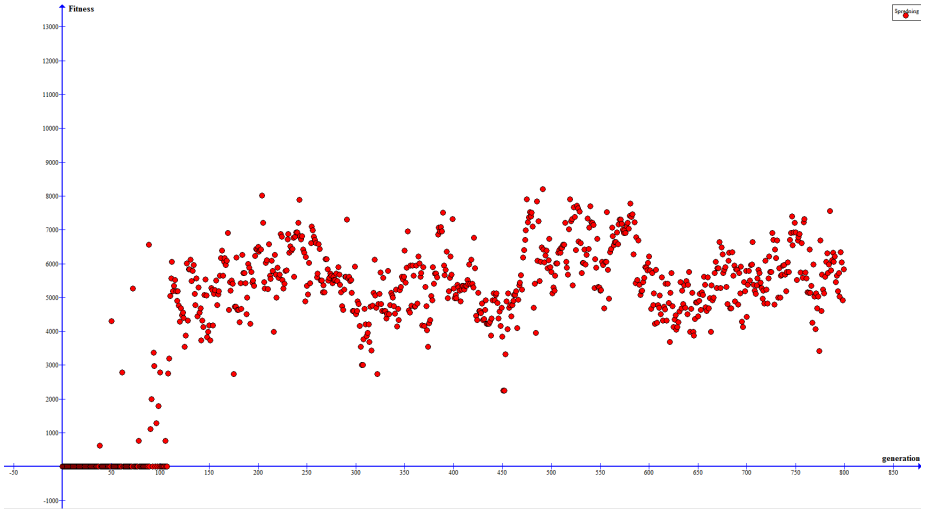


Figure 28: Graf over spredningen af fitness værdier uden crossover funktionen

På figur 29 ses det at der ikke er nogen signifikant vækst i fitnessen efter 30. generation.

Generation	7a fitness	7b fitness	7c fitness	8a fitness	8b fitness	8c fitness	9a fitness	9b fitness	9c fitness
0	1	1	1	1	1	1	1	1	1
1	1	2531	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	12091	5231	2701	4301	8351	7291	1	1	1
4	4621	11561	9401	10701	3251	8811	1	1	1
5	11231	9901	9721	13981	881	10731	1	1	1
6	12821	11081	11611	13581	15061	9921	1	1	1
7	10851	12561	9611	13021	13391	9681	1	1	1
8	7641	10931	1491	11351	12441	4161	1	1	1
9	8821	12781	4821	10611	10191	9881	1	1	1
10	13871	3791	14021	2701	9591	10631	1	1	1
11	13711	7541	9851	10161	11461	7671	1	1	1
12	7601	12291	8541	12061	8411	9441	1	1	1
13	13201	12571	12501	6131	10721	5691	1	1	1
14	7811	10461	2861	9131	12411	8931	1	1	1
15	9921	1961	11691	4541	14051	10321	1	1	1
16	10441	9901	411	3731	1	2011	1	1	1
17	8421	7771	8671	1731	3171	2171	1	1	1
18	8651	11201	5711	4951	1521	3971	1	1	1
19	9761	11891	2031	621	4371	10901	1	1	1
20	9031	6551	7101	501	4781	11271	14351	1	1
21	8081	8461	3861	4041	11821	10571	1	1	1
22	10731	1661	3751	14641	10821	7031	1	1	1
23	11631	9221	7671	13711	13201	4991	1	1	1
24	6491	8211	7891	5121	12401	5741	1	1	6241
25	5601	13541	1771	4121	13261	5921	1	1	1
26	7691	6271	6391	5041	13861	10311	3631	14331	2441
27	11151	11941	10591	4051	12681	10841	5601	15501	13171
28	11311	13231	7451	9621	1391	7371	5291	15141	13041
29	13351	13801	8201	941	10291	9981	5641	14961	13561
30	10311	10391	8811	5931	4121	5721	14231	14091	13981
31	6671	5371	6931	3531	10061	9041	15251	14851	14491
32	11371	5131	7751	9911	10761	6871	14931	13941	15341
33	13601	11641	2051	4701	14391	6031	14651	14911	14721
34	12881	8441	7011	6101	7001	4221	14171	13421	14621
35	9111	10561	5761	2881	7751	1701	14051	13441	13041
36	10611	6811	4201	181	7321	5811	14591	14901	12251
37	14251	5611	12131	8821	2871	5301	5301	14281	14071
38	10881	7291	8041	8751	5141	6801	15411	14751	5581
39	9361	7981	7781	2481	9231	8381	12341	14621	2801
40	8251	1901	7011	7861	4801	11841	14141	14781	11991
41	14921	5301	9851	7371	9411	6501	14121	14731	14311
42	6501	8311	4631	6681	3651	7221	4081	13911	15021
43	9481	10791	3791	8771	1851	7051	14341	14511	14351
44	13821	4681	10501	9011	13051	4781	14661	14381	14111
45	12161	11071	3751	5061	5531	3541	12481	15231	13221
46	12851	6681	4881	7611	4511	6691	5661	13651	15581
47	6991	4841	11031	4741	3941	7081	6951	14681	14971
48	16021	9731	6541	6511	7821	4401	15631	13161	15331
49	15181	8741	4951	6941	7621	6861	14451	4031	14981
50	13681	5511	3251	8901	1631	7821	13801	4741	14561
51	12931	561	9461	2121	2161	1831	14721	3881	14671
52	12661	11	7981	7601	3171	8831	14961	4971	14231
53	8681	4631	2351	11241	4301	6541	14701	13571	13901
54	11131	8951	5971	10981	5071	4891	5151	14611	13051
55	10591	11381	7171	8551	10181	1231	15181	14291	13251
56	13661	13601	6681	1321	6731	11671	14711	2631	12301

Figure 29: Fitness udvikling over generationer.

Udover problemerne med den genetiske algoritme er der ikke det samme antal timer i alle parallelklasserne. Der er altså nogle af parallelklasserne, der har for mange lektioner af diverse fag. Det kan reguleres i fitnessværdierne, hvor meget negativ fitness det skal give at have for mange lektioner af nogle fag. I det den nuværende løsning giver det -8000 i fitnessværdi, hvis der er for mange timer.

13 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 der er en stor mængde bindinger, der skulle tages højde for i skemalægningsprocessen. Derfor blev det 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 dette 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 en 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.

Det kan konkluderes at den softwareløsningen, der blev produceret, ikke virkede optimalt. Fitnessen når et plateau efter ca. den 10. generation, og derefter er det tilfældigt om der bliver generet et godt eller et dårligt skema. Som diskuteret i vurderingsafsnittet kan denne fejl skyldes, at crossoveren skaber for meget tilfældighed i algoritmen. Det blev forsøgt at lave en softwareløsning hvor crossoveren var bedre, og at den genetiske algoritme gjorde at skemaerne blev bedre i takt med, at der blev skabt flere generation. I denne løsning blev skemaer dog værre i forhold til den første løsning, men indivdernes fitness voksede i takt med, at der blev skabt flere generationer op til et lokalt maksimum ved allerede efter ca. 30. generation.

Derudover kræver softwareløsningen flere egenskaber, før den 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 bruger input indgå 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*. 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] *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.

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

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

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

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

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

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

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

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

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

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

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

[19] *C library function - time()*. URL: https://www.tutorialspoint.com/c_standard_library/c_function_time.htm.