

Datavetenskap

Ann-Marie Jansson

Petra Olsson

Schemapussling

Examensarbete, C-nivå

2000:25

Schemapussling

Ann-Marie Jansson

Petra Olsson

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Ann-Marie Jansson

Petra Olsson

Godkänd, 2000-05-31

Handledare: Stefan Lindskog

Examinator: Stefan Lindskog

Tack

Vi vill ge ett stort tack till vår förstående handledare, Stefan Lindskog, för vägledning och goda råd i samband med vårt examensarbete. Vi vill också tacka alla på EuroTime AB i Säffle för trevligt bemötande. Ett speciellt tack till Joakim Näs (vår handledare på EuroTime) för undervisning av systemet, hjälp vid dokumentationen och som bollplank för våra idéer. Även ett tack till Richard Åkerman, utbildare vid Eurotime i Skara, för hjälp med att förstå problematiken med schemapussling och hjälpen att sätta oss in i uppgiften.

Sammanfattning

Att schemalägga en avdelnings arbetstimmar så att alla blir nöjda och resursbehovet är uppfyllt, är ingen lätt uppgift. Ofta får förmän och chefer sitta i timmar för att pussla ihop ett fungerande schema.

EuroTime AB är ett ledande företag som arbetar med tidmätning och registrering av tid. 1989 började företaget att utveckla ett system för att registrera tid. Detta system kallas TID2000 och är en applikation för Windows. Det är en klassisk klient/server applikation som lagrar data i en relationsdatabas. TID2000 är en flexibel applikation som ger kunden möjlighet att anpassa den till hur de är vana att arbeta med tid. En finess i systemet tillåter de anställda att själva schemalägga sin arbetstid. Efter att de anställda gjort sina önskningar är det upp till ansvarig schemaläggare eller förman att fylla i och göra korrigeringar.

Avsikten med examensarbetet var att hitta en algoritm som tar hänsyn till de anställdas önskningar och sysselsättningsgrad samt företagets resursbehov. Algoritmen ska vara så rättvis som möjligt för samtliga personer. En slags poängsättning för de anställda ska användas. Algoritmen ska ta hänsyn till poängen vid nästa schemaperiod. Utvärderingen har visat att schemapussling är genomförbart och en möjlig algoritm redovisas i denna rapport.

Automatic scheduler

Abstract

To schedule working hours for a group of people so that everyone is satisfied and that the resource requirements are fulfilled, is not an easy thing to do. Often a foreman or supervisor spends several hours trying to get a complete schedule finished.

EuroTime AB is a leading company, which works with the measurement and registration of working time. In 1989 they started to develop a system that registers working time. This system is called TID2000, which is an application for Windows. It is a classic client/server application, which stores data in a relational database. TID2000 is a flexible application and the users can adjust it to how they are used to working with time. One aspect of the system is a component, which allows the employees to determine their own work schedule. After the employees have scheduled their requirements it is up to the responsible scheduler or the supervisor to fill in and correct the schedule.

The objective with this study was to find an algorithm that takes into consideration peoples' requests to work special hours, their employment time and the company's resource requirements. The algorithm has to be as fair as possible to everyone. A fair point allocation system for the employees should be used. The algorithm should take into consideration the next scheduling period. A feasibility study has shown that it is possible to implement such an algorithm and one possible algorithm is presented in this report.

Innehållsförteckning

<u>1</u>	<u>Introduktion</u>	xiii
1.1	<u>EuroTime AB</u>	xiii
1.2	<u>Bakgrund</u>	xiv
1.3	<u>Syfte och mål med uppgiften</u>	xiv
<u>2</u>	<u>Systembeskrivning</u>	15
2.1	<u>TID2000</u>	15
2.2	<u>Klient/Server och SQL</u>	15
2.3	<u>Systemöversikt</u>	15
2.4	<u>Detaljerad beskrivning</u>	17
2.4.1	<u>Person</u>	17
2.4.2	<u>Resursbehov</u>	18
2.4.3	<u>Schema</u>	20
2.5	<u>TID2000 installation</u>	22
<u>3</u>	<u>Förutsättningar och krav</u>	25
3.1	<u>Parametrar som påverkar schemapusslingen</u>	25
3.1.1	<u>Avvikelse från sysselsättningsgraden</u>	25
3.1.2	<u>Timbanken</u>	25
3.1.3	<u>Popularitetspoäng</u>	25
3.1.4	<u>Passens prioritet</u>	26
3.1.5	<u>ATL-parametrar</u>	26
3.1.6	<u>Frånvaro</u>	26
3.2	<u>Avgränsningar</u>	26
<u>4</u>	<u>Algoritm</u>	27
4.1	<u>Regler</u>	27
4.1.1	<u>Uppbyggnad</u>	27
4.1.2	<u>Hur en regel läggs till</u>	28
4.2	<u>Listor</u>	28
4.2.1	<u>Popularitetslista</u>	29
4.2.2	<u>Prioritetslista</u>	29
4.2.3	<u>Omplaceringslista</u>	29
4.3	<u>Personligt Schema</u>	30
4.3.1	<u>Parametrar som testas i det personliga schemat</u>	31
4.4	<u>Schemapussling</u>	32
4.4.1	<u>Schemapusslingsalgoritm</u>	32

4.5	Manuell testning av algoritmen	33
4.5.1	Algoritm I	33
4.5.2	Analys av Algoritm I	34
4.5.3	Algoritm II	35
4.5.4	Analys av algoritm II	35
4.5.5	Resultat	36
4.6	Misslyckad schemapussling	37
4.6.1	Alternativa scheman	37
5	Erfarenheter	39
6	Slutsatser	41
Referenser	43
A	Resurser	44

Figurförteckning

Figur 2.1 Systemöversikt	16
Figur 2.2 Exempel på hur man lägger in en nyanställd person	17
Figur 2.3 Exempel på hur ett personkort kan se ut	18
Figur 2.4 Exempel på valda inställningar	19
Figur 2.5 Exempel på hur man gör ett nytt resursbehov	20
Figur 2.6 Exempel på hur man gör ett nytt önskeschema	21
Figur 2.7 Exempel på hur ett önskeschema kan se ut	22
Figur 2.8 Exempel på en TID2000 installation	23
Figur 4.1 Regler	27
Figur 4.2 Översiktsbild av listorna	28
Figur 4.3 Funktionaliteten för personligt schema	30
Figur 4.4 Kontroll av dygnsvilan	32

Tabellförteckning

Tabell 4.1 Sammanställning av resultaten i algoritm I	34
Tabell 4.2 Sammanställning av resultaten i algoritm II	36

1 Introduktion

Denna uppsats är en redovisning av ett examensarbete inom datavetenskap som vi utfört vid EuroTime AB [4] i Säffle. Examensarbetet är en obligatorisk C-kurs omfattande 10 poäng som ingår i dataingenjörsprogrammet vid Karlstads universitet. Examensarbetet är en uppsats av utredande karaktär och utfördes under våren 2000 i samarbete med EuroTime AB.

1.1 EuroTime AB

EuroTime AB startades 1988 med utveckling av diverse elektronikkomponenter. Under 1989 påbörjades utvecklingen av tidregistreringssystem TID2000 för DOS, först tillsammans med ICA-rörelsen och därefter med landsting och industrier. Från våren 1990 har konceptet TID2000 marknadsförts.

Kunderna finns inom bland annat vård och omsorg, det vill säga kommun och landsting, handel och tjänsteföretag, bank, försvar, statliga verk, organisationer samt industri.

Förutom de egna programmen TID2000 för DOS och TID2000 för WINDOWS så har EuroTime som enda partner, medverkat i utvecklingen av IBM:s personalprogram Palett. Dessutom har EuroTime AB utvecklat ett Windows baserat schemabemanningssystem till Securitas koncernen som finns i drift inom Sverige och Norge. EuroTime utvecklar i C, C++, SQL Windows, Visual Basic och Active Server Pages (ASP) mot relationsdatabaserna SQL Base och Oracle. När det gäller utrustning för "tidsfångst" har EuroTime AB varit det ledande företaget i utvecklingen av olika sätt för att "fånga" tid. Tidregistrerings-terminaler/flexklockor finns i olika varianter och unikt har varit att kunna ansluta dessa direkt till kundens datanätverk, både till Ethernet och till Token ring [6]. EuroTime har också en klockserverlösning som även den är unik eftersom den både kan kommunicera med såväl EuroTimes som med konkurrenters program. Denna lösning ger kunden stor valfrihet. Genom nätanslutna flexklockor tillsammans med klockserver kan stora globala klocknät skapas genom användandet av befintliga datanät. Talsvar är ytterligare en lösning där telefonen används för att ge eller att få besked när det gäller tid och personaladministrativa frågor. En Webblösning är det senaste som lanserats, detta för att ge kunden maximal valfrihet och kombinationsmöjlighet.

1.2 Bakgrund

Idag är EuroTimes stora applikation TID2000 för Windows. Detta är ett Windows program som lagrar data i en relationsdatabas, en klassisk klient/server-applikation. Denna applikation är flexibel och kunderna kan anpassa denna till hur de arbetar med tid. TID2000 är utvecklad för att stödja ett flertal olika arbetstidsmodeller. En del av dessa innefattar att personalen själv får önska när de ska arbeta, något som kallas för önskeschema. Detta fungerar som så att resursbehov definieras för de delar av företaget man valt att arbeta med, normalt arbetar man så att varje avdelning lägger sitt eget schema och resursbehov. Utifrån resursbehovet ska sedan personalen själva gå in och schemalägga sig. Målet är att när schemat är lagt ska resursbehovet vara uppfyllt. Detta är en modell som när den fungerar är mycket omtyckt av personalen, eftersom den ger stora friheter att schemalägga sig själv. Hur hanterar systemet de dagar som inte har ett uppfyllt resursbehov? Idag är det upp till schemaansvarig att göra de korrigeringar som behövs. Tanken är att TID2000 ska kunna ge ett bättre automatiskt stöd för detta i framtiden.

1.3 Syfte och mål med uppgiften

Syftet med examensarbetet är att ta fram en algoritm för så kallad schemapussling. Det gäller att först definiera ingående parametrar till en sådan algoritm. Algoritmen ska sedan pussla ihop schemat så att resursbehovet uppfylls och att schemat blir så bra som möjligt för samtliga personer. Algoritmen ska också ta hänsyn till tidigare pusslingar. Som utdata ska förutom ett schema också finnas en poängsättning på de ingående personerna. Denna poäng ska sedan användas vid nästa pussling.

2 Systembeskrivning

Detta kapitel beskriver systemet TID2000 [3], dess programdelar och hur de fungerar. Sedan ges en mer detaljerad beskrivning av de viktigaste delarna i systemet som berör schemapusslingen.

2.1 TID2000

Systemet TID2000 består bland annat av en relationsdatabas som innefattar personuppgifter, arbetsscheman, fackliga avtal och frånvaro. TID2000 är uppdelat i fyra delar:

Dagligt arbete: Här sköts det löpande dagliga arbetet såsom personalhantering, scheman, bemanning, bokningar, frånvarohantering, rättning av avvikelser, framtagning av rapporter.

Tabellunderhåll: Här läggs all grundinformation in som företagsnamn, kostnadsställen, personkategorier, försäkringskassor.

Behörighet: Här hanteras de som skall använda TID2000 och vad de får lov att göra. I behörighet finns även funktioner för att administrera databasen och göra programuppdateringar.

Klockslagskalkylatorn: Detta är en enkel miniräknare som beräknar tider. Den kan räkna med timmar och minuter.

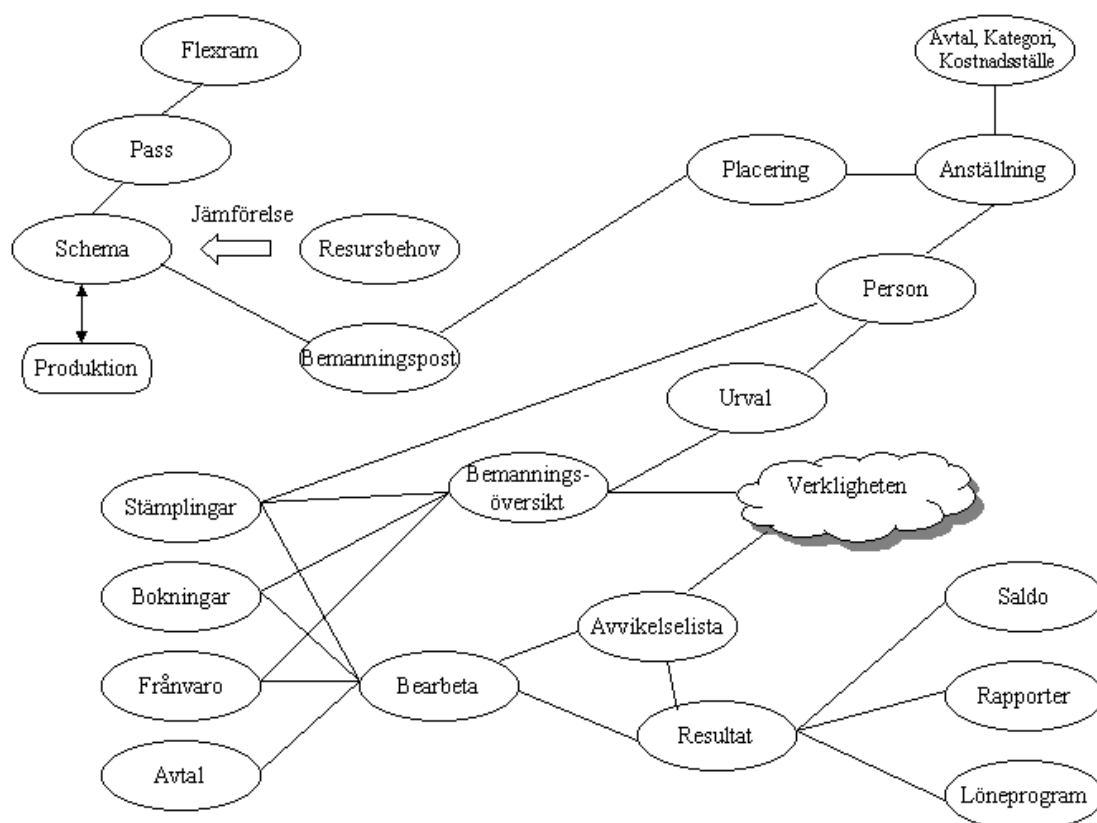
2.2 Klient/Server och SQL

TID2000 är gjort med en teknik som kallas klient/server och arbetar mot en SQL-databas. All tidsinformation lagras i en central databas och med hjälp av frågor plockas informationen fram från databasen. Frågespråket som används är SQL (Structured Query Language) [2].

Det fungerar i princip så att klienten, en arbetsstation, ställer en fråga till databasen. Databasen tar hand om frågan och sammanställer svaret i en tabell. Klienten får tabellen och presenterar den för användaren. Detta sätt att hantera stora datamängder innebär att allt det tunga arbetet görs i databasen (server) och din dator (klient) enbart ställer frågor och presenterar svaren på ett begripligt sätt.

2.3 Systemöversikt

I figur 2.1 visas en bild över hur den del av systemet som berör scheman hänger ihop.



Figur 2.1 Systemöversikt

En person har bland annat personnummer, adress och telefonnummer. Anställning är knuten till en person som visar sysselsättningsgrad, avtal, kategori och kostnadsställe. En person kan ha flera anställningar. Varje person kan arbeta på olika avdelningar och tillhöra olika kostnadsställen, för att möjliggöra detta finns placeringar. Flera personer utgör en grupp som kallas ett urval. En person kan tillhöra många urval.

Ett schema består av arbetspass med flexramar. Schemat bemannas med placeringar vilket ger en bemanningpost. Schemat kan jämföras med resursbehovet. Ett schema sätts i produktion, men kan även tas ur produktion för att göra ändringar.

I bemanningsöversikten ses urvalets aktuella bemanning. Här syns också stämplingar, bokningar och frånvaro. Bemanningsoversikten kan även ses grafiskt. I bokningar görs byten av arbetspass och redovisning av övertid. Under frånvaro registreras planerad och oplanerad frånvaro.

Avvikelselistan anropar bearbeta som kommer fram med ett resultat och som redovisas i avvikelselistan. Där felstämplingar ses och korrigeringar görs. Data från stämplingar, bok-

ningar, frånvaro och avvikelser bearbetas med hänsyn till avtal. Detta redovisas som saldo, rapporter eller skickas direkt till ett löneprogram.

2.4 Detaljerad beskrivning

Vi kommer nedan att behandla en liten del av systemet TID2000. Det som berör schema-pusslingen är person, resursbehov och schema.

2.4.1 Person

Vid ny anställning fylls tidsperioden i samt vilken sysselsättningsgrad personen kommer att ha, se figur 2.2. Dessutom fylls kostnadsställe, kategori, avtal, förmånsgrupp och anställningstyp i. Fälten befattning/tjänst, lön samt befattningsbeskrivning behöver ej fyllas i. Befattning/tjänst kan vara av typen personalansvarig för sin kategori.

The screenshot shows a Windows-style dialog box titled "Ny anställning (Jönsson, Jan)". It contains the following fields and controls:

- From:** 00-02-10
- To:** TV-
- Syss.grad:** 100,00%
- Kostnadsställe:** kontor
- Kategori:** programmerare 40:1
- Befattning/Tjänst:** (empty)
- Avtal:** Dagtid AB Flex
- Förmånsgrupp:** 11 Grupp 1a
- Anst.typ:** 1 Tillsviare...

Buttons on the right: OK, Avbryt, Hjälp.

Below the fields, there is a "Lön" section with a table and a "Befattningsbeskrivning" text area.

From	To	Månad	Timmar	Sen. dgr

Buttons at the bottom: Ny, Ja bort.

Figur 2.2 Exempel på hur man lägger in en nyanställd person

Med hjälp av personkortet, se figur 2.3, fås en snabb överblick av den anställda. Förutom personnummer, adress och telefonnummer visas också vilken typ av anställning och placering personen har och vilket schema personen går på. Det finns även ett fält där speciella noteringar kan fyllas i.

Ant:		Personnr:		Adress 1:	
4		701231-0000		Bergsg. 5, 661 00 Säfte	
Anställning:				Placering:	
00-01-20	TV	100,00%	tekniker	verkstad	
				00-01-20	TV
				tekniker	verkstad (5)
Schema:				Telefon:	
00-01-20	00-03-14	76,66%	1,00	1,40	
				Hemtelefon	0355-45678
Noteringar:					

Figur 2.3 Exempel på hur ett personkort kan se ut

2.4.2 Resursbehov

I resursbehovet anges hur många personer som verksamheten kräver av olika kategorier samt vilket kostnadsställe. Det finns tre olika typer av tidsintervall. Det är starttid, kontinuerlig och passvis.

Starttid: En grupp består av en eller flera perioder med starttid. Ett arbetspass startar inom en period. Därför är det bäst att passet börjar vid starttid.

Exempel:

Grupp: Namn på gruppen

Perioder: 00:00-08:00, 08:00-16:00, 16:00-24:00

Starttider: 00:00, 08:00, 16:00

Om ett pass börjar klockan 07:00 ligger den i perioden 00:00-08:00.

Kontinuerligt: Beskriver behov av antal personer för varje tidsperiod.

Datum, från klockan – till klockan och antal personer (min – normal – max) dessa är valbara.

Exempel:

Datum: 2000-02-07, klockan 06:00-10:00, antal: två personer (normal).

Datum: 2000-02-07, klockan 10:00-14:00, antal: fyra personer (normal).

Datum: 2000-02-07, klockan 14:00-18:00, antal: tre personer (normal).

Passvis: Beskriver behov av personal på varje pass per dag. Datum, pass och antal personer (min – normal – max).

Exempel:

Datum: 2000-02-07

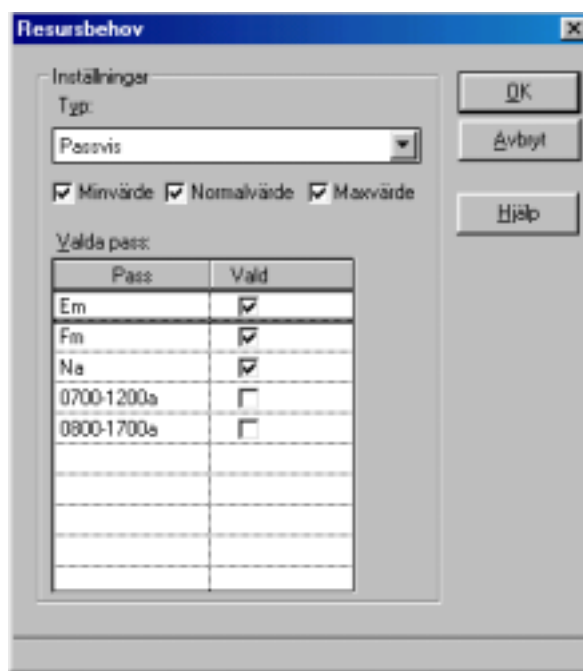
Pass:

Förmiddag: 06:00-14:00, antal: tre personer (normal).

Eftermiddag: 14:00-23:00, antal: två personer (normal).

Natt: 23:00-06:00, antal: en person (normal).

När resursbehov läggs in görs detta i tabellunderhåll i TID2000 och vissa inställningar görs. I fältet typ väljs vilken typ av resursbehov som ska gälla. Det finns som sagt tre olika typer: starttid, kontinuerligt och passvis. Resursbehovet kan visa minvärde, normalvärde och/eller maxvärde. Dessa värden anger minsta, normala respektive maximala behovet av arbetskraft. I figur 2.4 är passvis den valda typen och här görs val för de pass som skall användas i resursbehovet. När inställningarna är gjorda är det dags att logga in på dagligt arbete i TID2000.



Figur 2.4 Exempel på valda inställningar

Resursbehovet tilldelas ett namn som är en slags identifikation. Här anges vilken typ, kostnadsställe och kategori som resursbehovet ska ha. Kompetens/Meriter behöver inte vara med. Här fylls även i hur länge resursbehovet ska gälla, hur många veckor det avser och om det ska vara rullande, se figur 2.5.

Figur 2.5 Exempel på hur man gör ett nytt resursbehov

2.4.3 Schema

Först läggs ett tomt schema upp och detta fylls sedan med arbetspass. Ett arbetspass innehåller information om klockslag då arbetet ska starta och sluta samt rastens längd. Arbetspassen kan komma från passburken som innehåller fördefinierade pass. De kan matas in direkt i schemat eller kopieras från andra scheman. Arbetspassen placeras i schemats veckodagar för varje dag och varje schemarad. När ett schema är färdigt krävs det två åtgärder för att det ska bli aktiv, sätta schemat i produktion och bemanna. Om man väljer att sätta i produktion först kan man låta olika personer ha schemat under olika perioder, schemat går då ej att ändra. Om man å andra sidan väljer att bemanna först, gäller bemanningen automatiskt hela giltighetstiden, detta kallas att preliminärbemanna schemat.

Vid preliminärbemanning kopplas varje person till varsin schemarad. När ett schema fylls i skapas en bemanningspost. Varje person får en placering vid anställningen och egentligen är det bemanningsposten och placeringen som kopplas till varandra vid preliminärbemanning.

Det finns två sätt att ange flexramar:

Relativt: Flexramen anges relativt arbetspassets början och slut. Exempel ± 1 timme.

Absolut: flexramar anges i form av fasta klockslag.

En schemagrupp är en grupp av schemarader som används för att schemalägga en grupp av personer samtidigt där personerna ska ha individuella scheman. En rad i schemat för varje person. Denna schemagrupp kan sedan jämföras mot resursbehovet för samma grupp. När

schemagruppen bemannas, vilket ofta görs först och sedan sätts i produktion, hanteras varje rad för sig.

I ett önskeschema får personalen själva fylla i de arbetspass som önskas. Dessa önskningar jämförs sedan med det resursbehov som verksamheten kräver. Först skapas ett nytt schema med det datumintervall som perioden skall omfatta och en rad i schemat för varje person, se figur 2.6.

Figur 2.6 Exempel på hur man gör ett nytt önskeschema

Sedan preliminärbemannas schemat. Med det menas att varje rad i schemat knyts ihop med en viss person. Detta blir en mall som personalen själva får fylla i med sina önskningar. Arbetstiderna läggs in som pass eller genom att ange tiden i klockslag. Varje dag i önskeschemat kan anges med prioritet, hur viktigt det är att vara ledig eller arbeta en dag. Prioriteten kan anges med v = mycket viktig, 1, 2, 3 = lägre grader av angelägenhet, s = semester, t = tjänstledighet, se figur 2.7. När det är klart och datumet för önskningar har gått ut tas önskeschemat till schemagrupp, denna schemagrupp är en kopia av önskeschemat. Eventuella ändringar görs manuellt i schemagruppen. När schemat överrensstämmer med resursbehovet skall schemat sättas i produktion. Beräkning av popularitetspoängen fungerar endast för tidsintervallet kontinuerligt.

Sista datum för önskeschemat är 00-02-10 Gäller för period: 00-02-10 - TV

Radnamn	Nr	000207 P 1:Mån	000208 P Tis	000209 P Ons	000210 P Tor	000211 P Fre	000212 P Lör	000213 P Sön
5, Forskare, Albert	1	Fm	1 Em	1 Em	1 Fm	v	s	
8, Assistent, Anita	2	Em	1 Fm	3 Fm	2 Em	1 Em	1	

Passförklaring:

Kod	Dag	Start	Stop	A.tid	Rast
Fm	0	0600	1500	08:00	01:00
Em	0	1400	2300	08:00	01:00

Redovisning av kvoter mm:

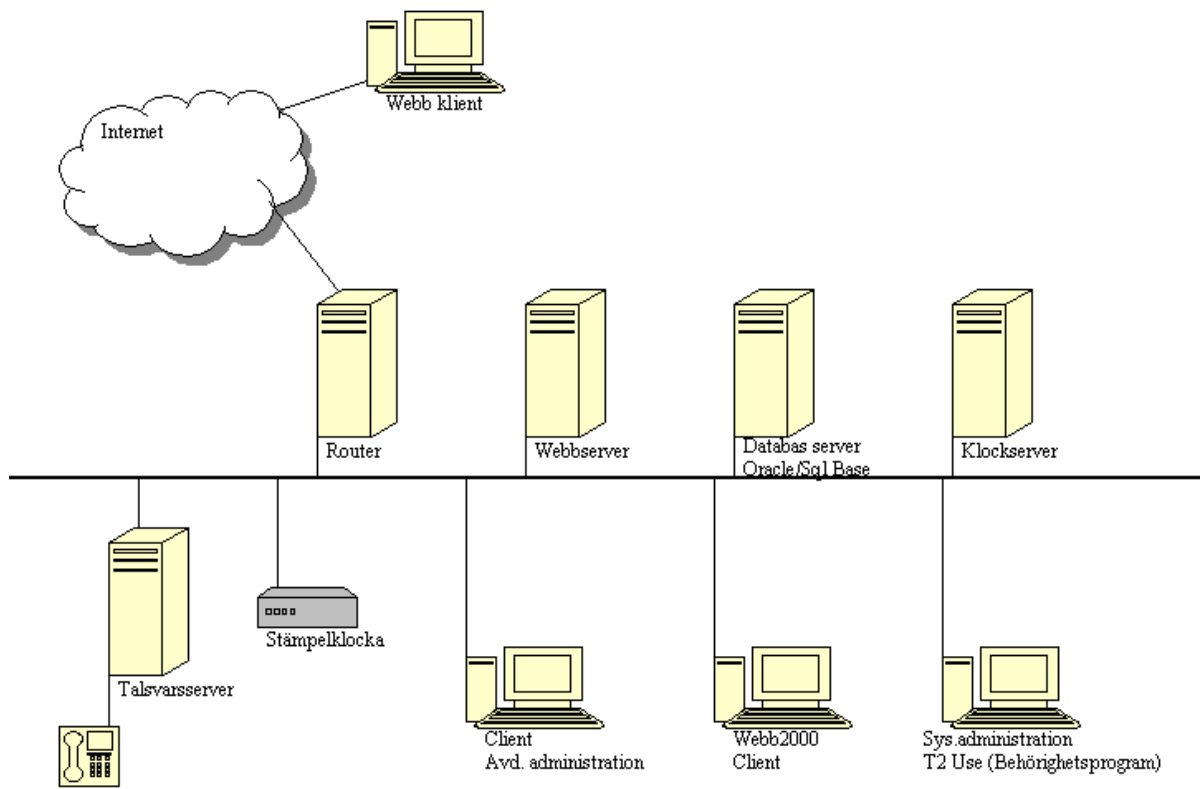
Nr	Tot.tid	V.tid	Syss.gr	A.dag	F.dag	Sem.k	K
1	32:00	32:00	80,00%	4	3	1,25	1
2	40:00	40:00	100,00%	5	2	1,00	1

Figur 2.7 Exempel på hur ett önskeschema kan se ut

Vid schemaläggning finns kontroller för veckovila, sysselsättningsgrad och överlappande pass. Veckovilan är minst 36 timmar enligt ATL (Arbetsbetslagen) [1] men kan justeras i lokala avtal. Personens arbetstid kontrolleras mot dennes sysselsättningsgrad.

2.5 TID2000 installation

En vanlig installation av TID2000 ser ut som i figur 2.8. Avdelningsadministratören sköter om den lokala delen av TID2000 när det gäller schemaläggning och dylikt. Avdelningsadministratören får sina administrationsrättigheter från systemadministratören som använder ett program för att registrera behörighet för olika användare. Stämpelklockan har en knappsats samt en magnetkortsläsare för bland annat registrering av in/ut stämplingar, frånvaro etc. Instämplingarna registreras sedan i klockservern. Det finns även möjligheter att registrera sig via telefon. Registrering sker då via en talsvarsserver som rings upp. En relativt ny och omtyckt tjänst är möjligheten att via Internet/intranet ansluta sig till en webbserver för att registrera sig. Möjligheten att registrera sig via webben kan vara bra om man befinner sig på resande fot. De databashanterare som TID2000 idag använder är SQL Base samt Oracle.



Figur 2.8 Exempel på en TID2000 installation

3 Förutsättningar och krav

Vår uppgift var att hitta en algoritm som ska konstruera ett schemaförslag utifrån personalens egna önsknings. Detta så att resursbehovet uppfylls och att schemat blir så rättvist som möjligt för samtliga personer. Algoritmen ska också ta hänsyn till tidigare scheman. I uppgiften ingick också att ta fram parametrar som påverkar schemapusslingen och de beskrivs i detta kapitel.

3.1 Parametrar som påverkar schemapusslingen

De regler som algoritmen ska använda för att avgöra om en person kan tillsättas på ett tänkt pass, kallas parametrar. Om personen uppfyller alla krav blir denne tillsatt på passet.

3.1.1 Avvikelse från sysselsättningsgraden

Samtliga anställda bör komma upp i sin sysselsättningsgrad. Avvikelsen är en gräns som talar om hur mycket sysselsättningen får avvika. Den kan variera ett antal timmar både uppåt och nedåt, per person och period. Parametern skall gå att ändra. Avvikelsen behövs för att få ihop ett komplett schema. Efter varje arbetad tidsperiod räknas timmarna ihop och avvikelsen från sysselsättningsgraden förs över till timbanken.

3.1.2 Timbanken

Vid varje ny schemapusslingsperiod skall hänsyn tas till timbankens saldo, saldot ska helst vara lika med noll. Efter varje arbetad tidsperiod sammanställs timbanken. Timbanken kan till exempel innehålla ± 20 timmar. Om någon arbetat mycket en period och dennes timbank är full, det vill säga $+20$ timmar, ska hänsyn tas till detta vid nästa schemapussling och schemaläggas mindre tid.

3.1.3 Popularitetspoäng

Popularitetspoäng beskriver hur populärt ett pass är. Här används formeln $P = B/A * L$ för att beräkna ett pass popularitetspoäng, där P = passets popularitetspoäng, B = behov i antal personer, A = antal personer som önskat pass och L = intervall i antal timmar. Vid beräkning av popularitetspoäng, se kapitel 2.4.3, används denna formel. Eftersom begränsning har gjorts till tidsintervallet passvis, sätts $L = 1$, alltså att alla pass oberoende längd är lika mycket värda.

Denna poäng ligger till grund för varje persons popularitetspoäng. En persons popularitetspoäng överförs till nästa tidsperiod. Bonuspoäng ges vid tvångsbemanning och vid pass som valts men inte tilldelats, ges tröstpoäng.

3.1.4 Passens prioritet

Varje pass i önskeschemat kan anges med prioritet, hur viktigt det är att vara ledig eller arbeta en dag. Prioriteten kan anges med v = mycket viktig, 1, 2, 3 = lägre grader av angelägenhet, s = semester, t = tjänstledighet. Passens prioritet är den viktigaste parametern som avgör var i prioritetslistan och omplaceringslistan personerna placeras.

3.1.5 ATL-parametrar

Veckovila som är 36 timmar i följd och dygnsvila som är 11 timmar i följd ska vara uppfyllt enligt arbetstidslagen. Dessa parameter skall kunna ändras om undantag förhandlats fram.

Max arbetstid är 12 timmar i följd enligt arbetstidslagen. Denna parameter skall gå att ändra om lagen ändras.

3.1.6 Frånvaro

Här finns regler som gäller för varje enskild person, de kallas personregler och kan omfatta mycket, se kapitel 4.1. För att det ska vara en personregel måste det vara en planerad handling som upprepar sig. Frånvaro kan också vara en engångsföreteelse som är planerad i förväg, detta kallas planerad frånvaro. Systemet stöder redan denna tjänst. Dessa två regler skiljer sig från varandra men behandlas på samma sätt i det personliga schemat.

3.2 Avgränsningar

De avgränsningar som gjorts är för att inte försvåra uppgiften, den är komplex nog som den är. En avgränsning är att endast behandla passvis resursbehov. Detta för att färdiga pass är lättare att hantera. Om det finns tid över, skulle det behövas en funktion som kan generera pass ur ett kontinuerligt resursbehov. Det finns störst behov av kontinuerligt resursbehov för att det stämmer bättre överens med arbetsgivarens behov av resurser.

Storhelger och helger behandlas som vanliga arbetsdagar och algoritmen tar ingen hänsyn till flexitid

När vi ska implementera algoritmen i en prototyp kommer många förenklingar och avgränsningar att behöva göras. Detta för att prototypen inte ska bli för stor och komplex.

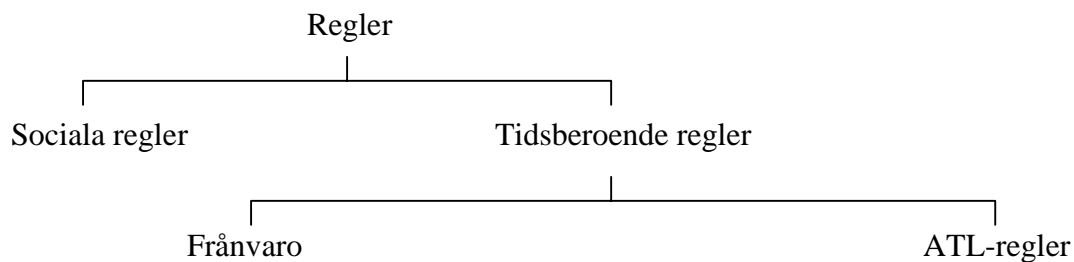
4 Algoritm

I detta kapitel beskrivs en tänkbar automatisk lösning på problemet med schemapussling. Först beskrivs de beståndsdelar som ingår i algoritmen.

4.1 Regler

Algoritmen behandlar två typer av regler, sociala regler och tidsberoende regler. Ett exempel på en social regel: "Vill gärna arbeta samma pass som X på grund av samåkning." Ingen hänsyn tas till dessa regler just nu.

Tidsberoende regler har alltid en början och ett slut. Dessa kan delas in i två grupper, ATL-regler som alltid gäller och frånvaro som har ett bestämt datum, se figur 4.1. Ett exempel på en regel som alltid gäller är dygnsvilan. "Arbetar inte måndagar" är ett exempel på en regel som är beroende av datum.



Figur 4.1 Regler

4.1.1 Uppbyggnad

En regel består av:

- Regelns namn
- Beskrivning av regeln
- Vem regeln gäller (en person, vissa grupper eller alla).
- Tid för när regeln gäller (början och slut eller hur länge).

4.1.2 Hur en regel läggs till

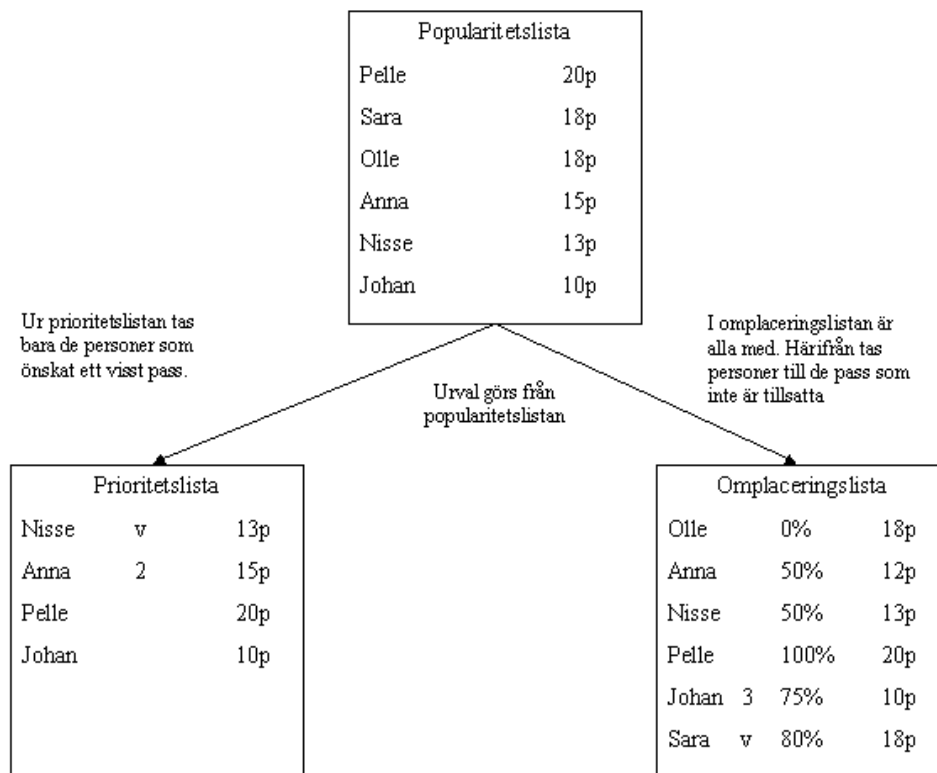
När det gäller regler från avtal och ATL knyts dessa regler till alla personer, genom urval. Alla parametrar som till exempel 11 timmars dygnsvila kan då lätt ändras i ett formulär för ”avtalsregler”.

En personregel är en planerad handling som upprepar sig. De läggs till i ett formulär som är knutet till den person regeln gäller. Här får man fylla i namn på regeln och beskrivning, tid när regeln börjar och slutar samt om periodiskt återkommande varje dag, vecka eller månad.

Planerad frånvaro fylls i under frånvaro. Detta stöds redan av systemet.

4.2 Listor

Det finns tre olika listor. Popularitetslistan som ligger till grund för prioritetlistan och omplaceringslistan, se figur 4.2.



Figur 4.2 Översiktsskild av listorna

4.2.1 Popularitetslista

För varje person beräknas summan av de önskade passens popularitetspoäng under en schemaperiod. För att det inte ska bli orättvist för deltidare räknas allas poäng om. Här använder vi formeln $N = P/S * H$, där N = ny popularitetspoäng, P = poäng man fått när man önskat, S = sysselsättningsgrad och H = heltid (100 %).

Den person med högst poäng hamnar överst i popularitetslistan. Om det förekommer samma popularitetspoäng på två eller flera personer slumpas deras ordning i popularitetslistan.

För varje pass som önskats med prioritet i önskeschemat görs ett avdrag på popularitetspoängen. Detta för att undvika ”max antal prioriteter”. Om det ska finnas ”max antal prioriteter” bör en kontroll ligga i önskeschemat.

Hänsyn bör också tas till en persons popularitetspoäng från tidigare schemaperioder. En person som arbetat många pass med höga poäng det vill säga mindre populära pass under en schemaperiod ska kunna få med sig den poängen till nästa period.

4.2.2 Prioritetslista

När ett önskat pass skall tillsättas görs ett urval från popularitetslistan bland de som önskat det passet. Denna lista kallas för prioritetslista. Överst i listan hamnar de personer som önskat passet med en viss prioritet. Om det skulle vara flera som önskat med samma prioritet eller ingen prioritet gäller popularitetspoängen. Den översta personen i prioritetslistan testas. Om något krav inte uppfylls fås inte det önskade passet och då tilldelas tröstpoäng. Personer plockas i turordning ur listan tills passet är tillsatt. När passet är tilldelat beräknas popularitetslistan om.

4.2.3 Omplaceringslista

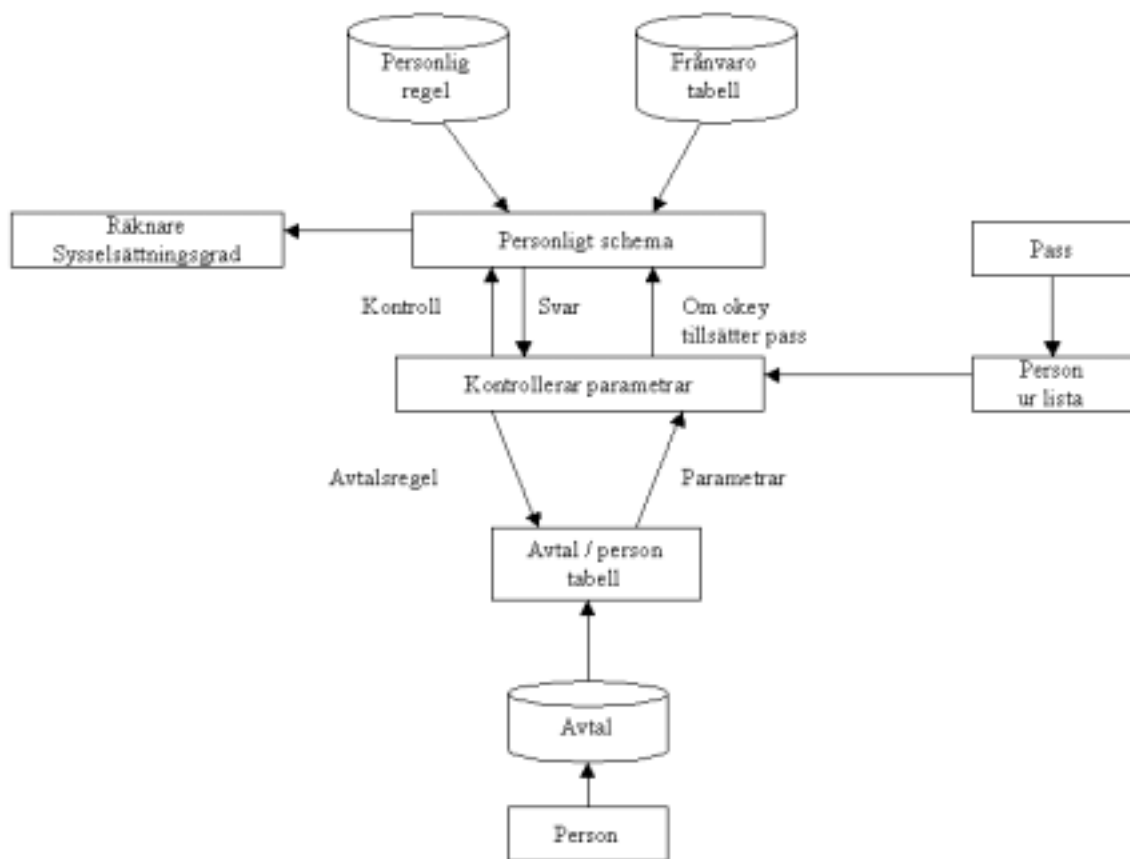
När det saknas personer på ett pass görs ett nytt urval från popularitetslistan. Denna nya lista kallas omplaceringslista. I omplaceringslistan är alla från popularitetslistan med. I den här listan kommer personerna att rangordnas efter sysselsättningsgrad. Den med lägst sysselsättningsgrad hamnar överst i listan och den med högst sysselsättningsgrad längst ned i listan. Om det är några med samma sysselsättningsgrad är det den med lägst poäng som hamnar överst. De som lagt in önskan med prioritet om ledighet läggs underst i omplaceringslistan. Den person som tvångsplaceras får bonuspoäng. När passet är fullständigt tillsatt beräknas popularitetslistan om.

4.3 Personligt Schema

När algoritmen startas genereras ett personligt schema för varje person. Från databasen hämtas information om planerad frånvaro och personregler, om det finns någon information för personen läggs detta in i det personliga schemat, se figur 4.3. Här finns även information om sysselsättningsgraden för perioden. Varje personligt schema utgör en schemarad, schemaraderna tillsammans bildar ett schema.

När en person blivit utvald antingen från prioritetslistan eller från omplaceringslistan tas dennes personliga schema fram. Detta schema används sedan för att göra kontroller av redan tillsatt på pass, sysselsättningsgrad, planerad frånvaro och ATL-regler.

Under schemapusslingen läggs passen in i det personliga schemat allt eftersom passen schemaläggs. Här finns också en räknare som räknar upp sysselsättningsgraden allt eftersom personen tillsätts på pass.



Figur 4.3 Funktionaliteten för personligt schema

4.3.1 Parametrar som testas i det personliga schemat

Redan tillsatt på pass är en kontroll som görs för att testa så att en person inte redan är tillsatt på ett pass som krockar med det passet som ska tillsättas. I denna kontroll ser man också om personen har någon planerad frånvaro eller någon personregel som skulle kunna krocka med tillsättningen av passet.

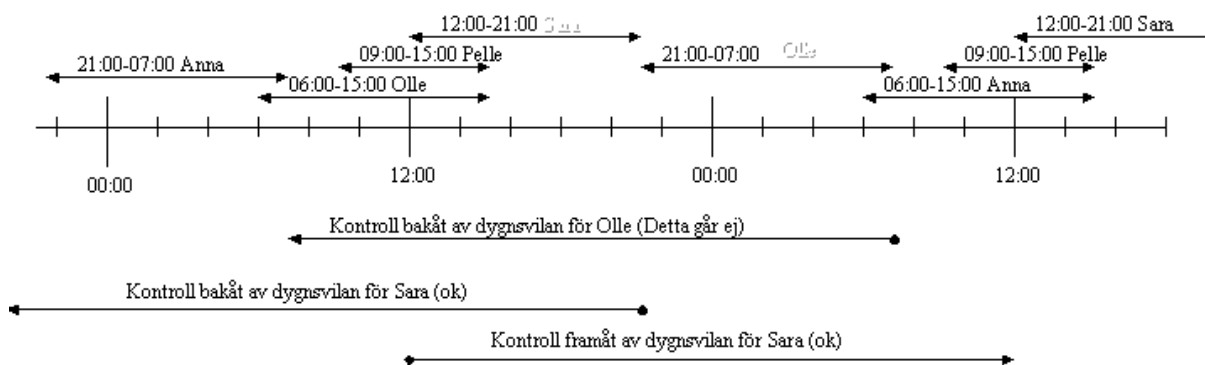
Veckovila kontrollerar ledigheten som måste vara minst 36 timmar i följd under en vecka. En vecka har totalt 168 timmar. Den senaste veckovilan söks upp, från slutet av denna veckovila räknas 168 timmar framåt i tiden. För att få tillsätta passet måste det finnas en veckovila inom dessa 168 timmar. För att kunna kontrollera veckovila i början på schemaperioden tittar man på sista veckan i det föregående schemat.

Dygnsvila kontrollerar ledigheten som måste vara minst 11 timmar i följd under ett dygn. Detta kontrolleras 24 timmar både bakåt och framåt i tiden. Vid kontroll bakåt börjar beräkningen från slutet av passet som ska tillsättas. När kontrollen görs framåt i tiden börjar beräkningen från början av passet som ska tillsättas, se figur 4.4.

För att kunna kontrollera dygnsvila i början på schemaperioden tittar man på sista dygnet i det föregående schemat.

Max arbetstid är en kontroll som kontrollerar att arbetstiden för en person inte får överstiga 12 timmar i följd. Detta kollas 12 timmar både bakåt och framåt i tiden. När kontrollen görs bakåt räknas timmarna från slutet av det pass som ska tillsättas. När kontrollen görs framåt i tiden räknas timmarna från början av passet som ska tillsättas.

Sysselsättningsgrad med avvikelse En persons schemalagda timmar summeras och en kontroll görs varje gång ett pass tillsätts, så att sysselsättningsgraden inte överskrider 110 %. För lite schemalagda timmar finns ingen kontroll på, men hänsyn tas till sysselsättningsgraden i omplaceringslistan.



Figur 4.4 Kontroll av dygnsvilan

4.4 Schemapussling

Innan schemapusslingen kan påbörjas måste resursbehovet fyllas i av schemaansvarig. Därefter får personalen göra sina egna önsknings i önskeschemat. Passen finns färdiga att väljas från ”passburken”. Varje dag i önskeschemat kan anges med prioritet, hur viktigt det är att vara ledig eller arbeta en dag. Här finns inga kontroller för hur de önskar sina pass.

4.4.1 Schemapusslingsalgoritm

Innan schemapusslingsalgoritmen görs en första kontroll där hela gruppens sysselsättningsgrad i timmar summeras och jämförs med resursbehovets timmar under hela schemaperioden. Om skillnaden är för ”stor”, är det inte möjligt att schemapussla. Då visas ett förslag på vad som kan ändras för att det ska gå att schemapussla. Det kan till exempel vara att lägga till en person med 40 timmars arbetsvecka. När detta är gjort kan nytt försök göras.

Nu beräknas passens popularitet. Detta görs genom att ett ifyllt önskeschema jämförs med resursbehovet. Poängen fås fram genom att antal personer som önskat pass delas med behovet av arbetskraft för samma pass, enligt formeln: $P = B/A$. P kallas för passets popularitetspoäng. Därefter beräknas personernas popularitetspoäng och en popularitetslista skapas, se kapitel 4.2.

I det tilltänkta schemat behandlas varje pass för sig. Algoritmen går igenom schemat fyra gånger dag för dag. Första gången behandlas de pass där antalet önsknings är mindre än behovet. Andra gången behandlas de pass där önskningarna är lika med behovet. Tredje gången behandlas resten av önskningarna, alltså där önskningarna är större än behovet. Fjärde

och sista gången tillsätts de resterande passen. Detta gör att antalet pass som ska tillsättas elimineras för varje varv i schemat.

Det första till tredje varvet i schemat används prioritetsslistan. Det som görs är att från popularitetsslistan utförs ett urval av de som önskat det pass som just nu ska tillsättas. Denna lista kallas prioritetsslista. Den översta personen i prioritetsslistan står först i tur att testas. Om en person inte uppfyller alla parametrar, se kapitel 4.3.1, tas nästa person ur prioritetsslistan. Alla pass behöver inte vara fullständigt tilldelade för att gå vidare till nästa pass. Detta upprepas för varje pass i schemaperioden.

Sista varvet då det ska tillsättas pass genom tvångsplacering, görs ett nytt urval från popularitetsslistan. Denna nya lista kallas omplaceringsslista. För att tillsätta pass där det saknas personal plockar man överst i omplaceringsslistan, tillvägagångssättet är samma som för prioritetsslistan. Detta upprepas för varje pass i schemaperioden.

4.5 Manuell testning av algoritmen

I det test som utfördes manuellt togs hänsyn till popularitetspoäng, sysselsättningsgrad, ATL-regler och planerad frånvaro. Ingen hänsyn togs till föregående schemaläggning, alla anställda hade noll i popularitetspoäng från början.

Testet gick till så att vi tog ett färdigt önskeschema från ett verkligt företag. Perioden för schemat sträckte sig över sex veckor. På schemat fanns det 14 anställda schemalagda och sju olika pass att välja på. Totalt fanns 331 pass att fördela på de anställda. En heltidstjänst innebar 37 timmars arbetsvecka. Detta användes som resursbehov. Ett antagande gjordes om att de anställdas summerade sysselsättningsgrad i timmar stämde överens med resursbehovets timmar.

Önsknings gjordes så att personerna skulle komma upp i sin sysselsättningsgrad eller i alla fall så nära den som möjligt. Även planerad frånvaro lades in på några av de anställda. När önskningarna gjordes togs ingen hänsyn till resursbehovet.

Popularitetspoängen beräknades och för att det inte skulle bli orättvist för deltidstjänsterna räknades allas poäng om. Formeln som användes för att räkna om poängen utifrån de olika sysselsättningsgraderna beskrivs i kapitel 4.2.1.

4.5.1 Algoritm I

När testet utfördes behandlades schemat dag för dag från början till slut. Passen tillsattes i ordning, det pass med störst behov först.

I tröstpoäng tilldelades fem poäng till person som inte fick önskat pass, denna poäng är ett förslag och går att ändra. Vid tvångsplacering gavs inga bonuspoäng, på grund av att ingen hänsyn togs till att popularitetspoängen skulle överföras till nästa schemapusslingsperiod. Enbart sysselsättningsgraden skulle föras med till nästa period.

Schemat behandlades fyra gånger.

1. Först tillsattes alla som önskat sig ett pass där antalet önskningar var mindre än behovet.
2. Här tillsattes alla som önskat sig ett pass där antalet önskningar var lika med behovet.
3. Här tillsattes de med högst popularitetspoäng av de som önskat sig ett pass, där antalet önskningar var mer än behovet.
4. Sist tillsattes de resterande passen, de med lägst sysselsättningsgrad tillsattes först.

4.5.2 Analys av Algoritm I

Spridningen av popularitetspoäng blev väldigt stor från 2 till 100. Det gavs inga poäng till de som önskat pass där behov saknades. Två personer önskade pass som saknade behov, därav deras låga poäng. Resursbehovet borde kanske vara tillgängligt för personalen vid önskningar.

Tröstpoäng tilldelades till person som inte fick önskat pass, här gavs fem poäng. En person ökade sina poäng med 65 på det viset. Spridningen minskade något men inte tillräckligt. Här drabbas de igen som inte visste hur resursbehovet ser ut. De är inte med i urvalet om önskat pass och kan heller inte få några tröstpoäng. Önskningarna ledde till att det blev grupperingar vid tilldelning av vissa pass, det var samma personer som önskade samma pass. Inom dessa grupper blev det en rättvis fördelning av tröstpoäng.

En person kom inte upp i sin sysselsättningsgrad, endast 84 %. Hon är en av två som arbetar heltid. Hon fick 22 av sina 30 önskade pass, de flesta låg i slutet av perioden. I början av tvångsplaceringen låg hon bra till med 74 %. Hon hade önskat kvällspass och det innebar att hon inte kunde ta ett förmiddagspass nästa dag på grund av dygnsvila. Eftersom passen tillsattes dag för dag blev det svårt att tvångsplacera henne i slutet av perioden. Vid tvångsplacering tillsattes passen efter sysselsättningsgraden och de som bäst behövde långa pass fick inte alltid dem. De pass med störst behov är inte alltid de längsta passen.

Det tilldelades ingen bonuspoäng till dem som blev tvångsplacerade. Popularitetspoängen blev därför mycket spridd, men eftersom inte poängen förs över till nästa period har det inte någon betydelse, se tabell 4.1.

Tabell 4.1 Sammanställning av resultaten med algoritm I

Anst. nr.	Syssels. grad i procent	Pop poäng efter önskade pass	Pop poäng efter tillsättning	Syssels. grad före tvångsplac. i procent	Syssels. grad efter tvångsplac. i procent	Fått önskade pass i procent	Antal tvångs plac.
2502	100	17	52	54	97	54	14
2503	77,5	91	91	84	101	86	4
2504	77,5	15,5	50,5	32	100	32	18
2505	77,5	4	14	7	99	7	22
2506	77,5	30,5	60,5	74	92	73	4
2507	77	13	78	39	101	38	15
2508	75	17	67	55	99	55	10
2510	74	17,5	62,5	55	100	59	10
2519	74	35	55	104	104	83	0
2520	74	2	7	4	102	3	23
2521	74	14	74	36	100	35	15
2522	74	16	46	40	97	39	13
2526	100	23,5	63,5	74	84	73	3
2527	77,5	17	48	47	100	42	14

4.5.3 Algoritm II

Förutsättningarna är samma som i algoritm I. Det är samma resursbehov, önsknings- och personal. Även tillvägagångssättet är i grunden lika där schemat behandlas fyra gånger. En del förändringar har dock gjorts i algoritm II. Här slumpas dagarnas ordning vid tillsättning av pass och de längsta passen tillsätts först. En bonus på tre poäng tilldelas vid tvångsplacering av pass, denna poäng är ett förslag och går att ändra. Popularitetspoängen och sysselsättningsgraden förs med till nästa schemapusslingsperiod.

4.5.4 Analys av algoritm II

Samma tröstpoäng tilldelades som i algoritm I när man inte fått önskat pass, det vill säga fem poäng. Spridningen minskade och blev något mindre än i algoritm I. Det berodde på att vi slumpade dagarnas ordning. Passen fördelades olika i algoritm I och algoritm II.

Tre poäng tilldelade i bonuspoäng till dem som blev tvångsplacerade. Detta för att popularitetspoängen ska tas med till kommande period och summeras. Spridningen av popularitetspoängen är inte så stor nu som i algoritm I där poängen inte sparades till kommande period utan hänsyn togs bara till sysselsättningsgraden.

När det gäller sysselsättningsgraden hamnade alla innanför avvikelsegränsen som föreslagits till 10 %. Den person som inte höll sig inom gränsen i algoritm I hamnade nu åtminstone på rätt sida gränsen. Detta för att dagarnas ordning slumpades och att de längsta passen tillsattes först, se tabell 4.2.

Tabell 4.2 Sammanställning av resultaten med algoritm II

Anst.nr.	Syssels.grad i procent	Pop.poäng efter önskade pass	Pop.poäng före tvångsplac.	Pop.poäng efter tvångsplac.	Syssels.grad före tvångsplac. i procent	Syssels.grad efter tvångsplac. i procent	Fått önskade pass i procent	Antal tvångs- plac.
2502	100	17	52	91	58	99	58	13
2503	77,5	91	91	100	84	97	86	3
2504	77,5	15,5	55,5	103,5	33	99	32	16
2505	77,5	4	9	75	11	98	11	22
2506	77,5	30,5	60,5	72,5	74	100	77	4
2507	77,5	13	52	115	39	99	43	14
2508	75	17	62	92	54	100	55	10
2510	74	17,5	62,5	89,5	59	99	55	9
2519	74	35	70	76	88	96	71	2
2520	74	2	12	78	0	99	3	21
2521	74	14	69	99	52	98	45	10
2522	74	16	46	88	40	99	39	14
2526	100	23,5	63,5	78,5	74	91	73	5
2527	77,5	17	52	90	47	97	42	13

4.5.5 Resultat

Det gick bättre att få ihop ett schema än väntat, båda algoritmerna resulterade i ett färdigt schema. Vid kontrollräkandet stämde resursbehovet inte helt överens med sysselsättningsgraden. Det fattades 60 timmar av resursbehovet för att fylla de anställdas totala sysselsättning, kanske var det därför det gick så lätt att få ihop ett komplett schema.

Några parametrar som användes vid testningen i båda algoritmerna är de som ATL föreskriver. De är väl tilltagna och brukar oftast förhandlas om, det försvårade schemalaggingen. Ett exempel är där dygnsvilan med sina 11 timmar är alldeles för lång där ett kvällspass skulle kunna avslutas klockan 22.00 och morgonpasset börja klockan 07.00. Resultatet blir då att ett av passen inte kan tillsättas.

Algoritm II är bättre än algoritm I, för att dagarnas ordning slumpas och detta ger en mer rättvis fördelning av passen. I algoritm II förs både poängen och sysselsättningsgraden över till nästa schemapusslingsperiod. Att enbart ta hänsyn till sysselsättningsgraden från föregående period skulle bli orättvist eftersom sysselsättningsgraden endast används i sista genomgången av schemat.

4.6 Misslyckad schemapussling

Skulle det inte gå att få ihop schemat kan man ändra avvikelser från sysselsättningsgraden. Sedan kör man algoritmen en gång till från början.

Ett annat förslag är att algoritmen skapar ett schema så långt det går och nöjer sig med ett ofullständigt schema. Schemaläggaren får då ansvaret att fylla i där det fattas.

4.6.1 Alternativa scheman

Ett förslag är att man genererar flera alternativa scheman. Vid genomgång av schemat den tredje gången där antalet önskningar är större än behovet genereras ett antal alternativa scheman. Varje kombination av personer för varje pass genererar ett schema. Formel för att ange antalet kombinationer av scheman är n över r [5], där n = antalet önskningar och r = behovet.

Vid tvångsplacering av pass, tvångsplaceras så många personer som krävs för att fylla upp behovet. Det tas också fram alternativa möjliga personer som begränsas till tre stycken. Dessa personer plus de som tvångsplacerats bildar olika kombinationer. Alla dessa kombinationer ger ett alternativt schema. Samma formel används, n över r ger antalet kombinationer av scheman, där n = antalet tvångsplacerade plus tre och r = behovet.

Detta gör att beräkningarna blir väldigt komplexa. När ett alternativt schema hamnar i situationen att det inte är möjligt att tillsätta pass ska detta schema förkastas. Här fås många möjliga scheman och detta gör att algoritmen inte behöver startas om.

5 Erfarenheter

Det har varit nyttigt att göra ett ”riktigt” arbete på ett verkligt företag. Att sätta sig in i ett så stort och användbart system och förstå problematiken i uppgiften har varit både intressant och besvärligt. Vi har behandlat ett antal parametrar som påverkar schemapussling, men det finns troligtvis många fler som vi inte kommit på än. Att sedan ta fram, testa och analysera en algoritm har känts kreativt och berikande. Det första testet av algoritmen gjorde att vi upptäckte några mindre bra detaljer som resulterade i att vi förbättrade algoritmen. Detta tog den tid vi planerat för att skapa prototypen. Det kändes rätt att lägga ner mer tid på algoritmen för att få den så bra som möjligt istället för att implementera en mindre bra algoritm. När vi dokumenterade våra idéer, först då fick vi en strukturerad bild av våra tankar. Alla idéer och tankar vi haft kan omöjligt skrivas ner. Examensarbetet har lärt oss att arbeta mer självständigt. Det vore roligt att se ett fortsatt arbete på schemapusslingsalgoritmen och en utveckling av prototypen.

6 Slutsatser

Syftet med examensarbetet var att hitta en algoritm som rättvist schemalägger anställda och deras arbetstider utifrån deras önskningar. Den skall också ta hänsyn till tidigare schemapusslingar. Resultatet har blivit två teoretiskt möjliga algoritmer, den ena bättre än den andra. Trots ord om omöjligt, tvivlade vi aldrig på att hitta en algoritm som automatiskt schemapusslar.

En prototyp skulle ha utvecklats men eftersom tiden inte räckte till lade vi ner mer arbete på att få fram en bättre algoritm.

Ytterligare ett antal ”verkliga” scheman bör testas med algoritmen och analyseras. Hänsyn bör också tas till tidigare schemapusslingar, där vi inte vet hur popularitetspoängen och sysselsättningsgraden påverkar schemapusslingen.

I kapitel 3.1.2 står det att vi skall använda timbanken. Ingen hänsyn har tagits till detta i den nuvarande algoritmen. Vi har inte lyckats komma på någon bra lösning där timbanken ska ingå, på grund av att timbanken innehåller redan arbetade timmar och sysselsättningsgradens timmar är planerade timmar.

Ett senare förslag har inkommit att vid tilldelning av tröstpoäng ge mer till de som önskat med prioritet men ändå inte får passet. Detta för att kompensera poängavdraget som görs vid önskningar med prioritet.

Angående behovet att generera pass från kontinuerligt resursbehov i kapitel 3.2, fanns ingen tid men behovet kvarstår.

Slutligen har företaget planer på att fortsätta utvecklandet av schemapussling eftersom behovet är stort.

Referenser

- [1] ATL (Arbetsstidslagen) <http://www.jit.se/lagbok/982673t.html>.
- [2] Date C.J, An Introduction to Database System, Addison-Wesley, ISBN: 0201385902, first edition, 1999.
- [3] EuroTime AB, Användarhandbok TID2000 för windows, 1997-10-31.
- [4] EuroTime AB presentationssida <http://www.eurotime.se>.
- [5] Grimaldi Ralph P, Discrete and Combinatorial Mathematics, Addison-Wesley Publishing Company, ISBN: 0201600447, third edition, 1994.
- [6] Tanenbaum Andrew S, Computer Networks, Prentice Hall, ISBN: 0133342481, third edition, 1996.

A Resurser

Programvara:

TID2000 version 1.4.0 PTF 3 Databasmotor: ORACLE 7.5.0.4

Office 2000

Outlook Express