Master thesis

Work Distribution of a Heterogeneous Library Staff - A Personnel Task Scheduling Problem

Claes Arvidson, Emelie Karlsson

LiTH - MAT - EX - - 04 / 04 - - SE

Work Distribution of a Heterogeneous Library Staff - A Personnel Task Scheduling Problem

Optimeringslära, Linköpings Universitet

Claes Arvidson, Emelie Karlsson

LiTH - MAT - EX - - 04 / 04 - - SE

Exam work: **30 hp**

Level: **A**

Supervisor: **T. Larsson**,
Optimeringslära, Linköpings Universitet

Examiner: **E. Rönnberg**,
Optimeringslära, Linköpings Universitet

Linköping: **June 2016**

Abstract

Here is where you can write your abstract. It may be very long, or it may be very short, the reason you have an abstract is for people not to be forced to read lots of crap.

But still, they will have to read your abstract. After all, the abstract is what everyone reads...

Keywords: Keyword One, Chemostat, Another Key-Word, Key, Clé, Mot de cle, Nyckelhål, XBOX, Dagens viktigaste nyckelord, and Keywords.

URL for electronic version:

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-77777

Acknowledgements

I would like to thank my supervisor, I would like to thank my supervisor, I would like to thank my supervisor, I would like to thank my supervisor…

I also have to thank, I would like to thank my supervisor, I would like to thank my supervisor, I would like to thank my supervisor, I would like to thank my supervisor…

My opponent NN also deserves my thanks, I would like to thank my supervisor, I would like to thank my supervisor, I would like to thank my supervisor…

Nomenclature

Most of the reoccurring definitions, symbols and abbreviations are described here.

Definitions

Plocklista	Text
Library on wheels	Text
Rostering	Text

Symbols

Y_0	The amount of the variable Y inserted into a system.
$\hat{Y}$	The unit-dimension of the variable Y, for example $\hat{t} = 1s$.
$\bar{Y}_i$	A steady state (number i) value of Y.
K_i	Constants used in kinetic expressions, for example K_I.
$\mathbf{A}$	The system matrix.

Abbreviations

Exp	Text
Info	Text
PL	Text
PTSP	Text
SMPTSP	Text
CPI	Competitive Product Inhibition (or Inhibited)
CSI	Competitive Substrate Inhibition (or Inhibited)
CSTR	Continuous Stirred Tank (bio)Reactor
MMI	Michaelis-Menten Inhibition (or Inhibited)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

At a library absence can cause problems, both due to lack of personnel as well as due to the qualifications required to perform a task varies. If a worker were to be unavailable a day because of a meeting or being ill it would require for a stand-in to fill the vacancy. Therefore, it is of great interest to have a schedule with as many skilled stand-ins as possible to overcome such disturbances. Furthermore, the library personnel have certain demands and preferences as to how a satisfactory working schedule should be. For instance, it is neither preferable to work more than one evening each week nor work more weekends than required.

The central library of Norrköping is more than 100 years old. The library has more than 40 employees. These consist of both librarians and assistants, who handle simpler tasks. The library is open both weekdays until 20:00 and during weekend days until 16, which is challenge for the schedulers of the library since this requires time compensation for the staff. The library also provides its services to more than five other smaller libraries.

1.2 Problem description

1.2.1 Description of the daily tasks at the library

The most important activity at a library is the activity directed towards the public. This includes lending books services as well as providing customers with helpful information about the resources at their disposal. These are referred to as "outer tasks". At the same time, the uppsättning of books must be maintained, the returned books must be sorted and put back, the web page must be up to date and so on. Such work is often referred to as "inner work" and is equally part of the everyday tasks of a librarian.

At the library of Norrköping, three main outer tasks can be identified as working in the service counter (sv. expiditionsdisken), working in the information counter (sv. informationsdiken) and assembling books according to the "fetch list" (sv. plocklista). These tasks can be performed by either librarians or assistants, as descirbed in Table 1.1. Note that the fetch list is actually only one task which covers the whole day.

Table 1.1: Outer tasks can be performed exclusively by librarians or by both assistants and librarians.

Task	Description	Qualification
Service Counter (Exp)	Administring loans, library cards and the loaning machine	Assistants, Librarians
Information Counter (Info)	Handling questions about the library's resources.	Librarians
Fetch List (PL)	Fetching books that are to be sent to other libraries.	Assistants, Librarians

As the number of visitors in the library differs at different times of the day and during different days so does the demand for people for the three tasks. The demand of people at for the different tasks is illustrated in table 1.2.

Table 1.2: Demand of staff for the three daily tasks.

Day	Time	Exp demand	Info demand	PL demand
Mon-Fri	08:00-10:00	2	2	1
Mon-Fri	10:00-13:00	3	3	1
Mon-Fri	13:00-16:00	3	3	1
Mon-Fri	16:00-20:00	3	3	-
Sat	11:00-16:00	3	3	-
Sun	11:00-16:00	3	3	-

As is the case with most libraries, the Central Library of Norrköping also has responsiblities that fall outside of it's normal daily activities. One such resposibility is the running of a smaller library filial in Hageby, situated in a suburbian area of Norrköping, during weekends. For this tasks only librarians are qualified as the placement implies all types of library tasks.

Similarly, only librarians are qualified for the task known as "Library on Wheels" (sv. Bokbussen), which is a type of library bus, providing citizens in remoter areas of the city with books and other library services. The Library on Wheels has only operates a few times a week and the schedule differs between even and odd weeks.

1.2.2 Personnel attributes

At a library, the most important workers are the librarians. They make sure the library's resource collection is up to date, that the visitors find what they're looking for and perform a larger number of inner tasks. Simpler tasks can also be performed by a group of workers called assistants and they help the librarians in keeping the library running in many ways.

The Central Library of Norrköping currently has 39 workers, 23 of which are librarians and 16 of which are assistants. All staff have different availability for performing tasks, depending on their working hours and the amount of inner work they are in charge of. Generally, each worker is assigned one evening per week and once per five weeks, he/she is asssigned to weekend work. The

Table 1.3: Demand of staff Library on Wheels

Day	**Time**	**LoW - odd week**	**LoW - even week**
Mon	08:00-10:00	1	1
Mon	16:00-20:00	1	-
Tue	08:00-10:00	-	-
Tue	16:00-20:00	-	-
Wed	08:00-10:00	1	1
Wed	16:00-20:00	1	1
Thu	08:00-10:00	1	1
Thu	16:00-20:00	1	1
Fri	08:00-10:00	1	-
Fri	16:00-20:00	-	-

Table 1.4: Demand of staff at Hageby

Day	**Time**	**Hageby**
Sat	11:00-16:00	1
Sun	11:00-16:00	1

following week is compensated by two free days, placed according to the wishes of the worker.

Let's consider a sample worker Bob. Bob works full time and is also assigned to work on Wednesday evenings. Bob works weekend on the fourth week and has chosen to take out his days off on Thursdays and Fridays on week five. His availability is thus illustrated in table 1.5. The availability schedule repeats itself after five weeks (Note: Whether or not Bob is assigned any tasks during these weeks is not illustrated in the table).

All workers have a five week schedule in the same manner as sample worker Bob. However, in order to meet the weekend demands as illustrated by Tables 1.2, 1.3 and 1.4, it is evident that not all workers can be assigned for weekend work at week 4. Instead, a worker's schedule should be seen as a *relative schedule*, which can be shifted by whole weeks. We refer to this as the *week rotation*. The relative schedule is relative to the *overall schedule* for the library. If Bob for example has week rotation 1, this means the week 4 in his relative schedule would be placed on week 1 in the overall schedule.

Considering again sample worker Bob's schedule, it may look as if Bob can be scheduled for tasks up to four times a day. However, there are regulations controlling how the outer tasks are to be distributed among the workers. Regulations say that:

1. A worker can at a maximum perform one task per day.

2. A worker works only one weekend in five weeks.

3. A worker must work Friday evening shift, Saturday shift and Sunday shift consecutively. For HB-workers, no Friday evening shift.

4. A worker is only allowed to work one evening per week, excluding the Friday evening of the weekend week.

Table 1.5: Schedule for sample worker Bob. Yellow signifies availability. In parenthesis, the weekend shift. Week rotation 4.

Week 1	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00-10:00 (11:00-16:00)	🟨	🟨	🟨	🟨	🟨		
10:00-13:00	🟨	🟨	🟨	🟨	🟨		
13:00-16:00	🟨	🟨	🟨	🟨	🟨		
16:00-20:00			🟨				
Week 2	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00-10:00 (11:00-16:00)	🟨	🟨	🟨	🟨	🟨		
10:00-13:00	🟨	🟨	🟨	🟨	🟨		
13:00-16:00	🟨	🟨	🟨	🟨	🟨		
16:00-20:00			🟨				
Week 3	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00-10:00 (11:00-16:00)	🟨	🟨	🟨	🟨	🟨		
10:00-13:00	🟨	🟨	🟨	🟨	🟨		
13:00-16:00	🟨	🟨	🟨	🟨	🟨		
16:00-20:00			🟨				
Week 4	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00-10:00 (11:00-16:00)	🟨	🟨	🟨	🟨	🟨	🟨	🟨
10:00-13:00	🟨	🟨	🟨	🟨	🟨		
13:00-16:00	🟨	🟨	🟨	🟨	🟨		
16:00-20:00			🟨		🟨		
Week 5	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00-10:00 (11:00-16:00)	🟨	🟨	🟨				
10:00-13:00	🟨	🟨	🟨				
13:00-16:00	🟨	🟨	🟨				
16:00-20:00			🟨				

An example of a feasible schedule for the sample worker is provided in Table 1.6.

Table 1.6: Example of a feasible first week for the sample worker.

Week 1	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
08:00-10:00 (11:00-16:00)	🟨	Exp	🟨	PL	🟨		
10:00-13:00	Exp	🟨	🟨	PL	🟨		
13:00-16:00	🟨	🟨	🟨	PL	Info		
16:00-20:00			Info				

1.2.3 Main objective: increase number of stand in personnel

The real challenges in creating a schedule according to the instructions above, is not to distribute the tasks but to make sure there is stand in personnel ready to take shifts in the case other personnel is otherwise occupied or is on sick-leave. This emergency back-up system is crucial for the library in order to be able to keep the library desks open and fully staffed. This is the highest priority of the library.

A stand in is defined as a worker who is available for inner tasks during the three day-shifts, but has not any shift scheduled for that day. Inner work sometimes prevents personnel from being scheduled as a stand-in for the outer tasks, and therefore also information about such work must be known to the scheduler. Both librarians and assistants can be scheduled as a stand in, but only librarians are qualified to take emergency shifts at the Info desk. Since the regular library activity is the most crucial activity, there is no need for stand ins during evening and weekends. Similarly, there is specific personnel assigned as stand-ins for the Library on Wheels or for Hageby.

1.2.4 Secondary objectives: repetitiveness of the schedule

1.3 Method

1.4 Topics Covered

”The thesis is divided into...”

Table 1.7: Overall schedule for week 1.

Schedule week 1					
Monday	**Exp**	**Info**	**PL**	**LoW - odd**	**LoW-even**
08:00 - 10:00	W1, W2	W1,W2	W1	W1	W1
10:00 - 13:00	W1, W2, W3	W1, W2, W3	W1	-	-
13:00 - 16:00	W1, W2, W3	W1, W2, W3	W1	-	-
16:00 - 20:00	W1, W2, W3	W1, W2, W3	W1	-	-
Tuesday	**Exp**	**Info**	**PL**	**LoW - odd**	**LoW-even**
Wednesda	**Exp**	**Info**	**PL**	**LoW - odd**	**LoW-even**
Thursday	**Exp**	**Info**	**PL**	**LoW - odd**	**LoW-even**
Friday	**Exp**	**Info**	**PL**	**LoW - odd**	**LoW-even**

Chapter 2

Literature review

The scheduling problem is a mathematical optimization problem which has been studied since the 1950's and concerns creating a feasible and satisfactory schedule for workers or machines performing tasks. One of the most extensive overviews in the area is provided by Ernst et al. (2004). They state that, although the complexity of the scheduling problem has not increased in recent years, the mathematical models used to solve the scheduling problems have become more realistic and refined. Modern scheduling problems often concern the distribution of tasks and the creation of worker shifts, taking also into account softer values such as worker satisfaction and worker fatigue. Due to this modeling refinement as well as the development of more powerful computational methods, it is possible today to solve scheduling problems in a more satisfactory way than before.

In this section, the scheduling problem is classified into different subcategories which are areas related to the work of this paper. A few relevant areas for our work include Personnel Task Scheduling Problems (PTSP), Shift Minimization Task Scheduling Problems (SMTSP), Tour Scheduling Problems (TSP) and a few variations of these. Models and solutions methods of these problems will be discussed during each topic and a summary of the discussion will be provided.

2.1 Personnel Task Scheduling Problem

In many practical instances production managers will face the Personnel Task Scheduling Problem (PTSP) while scheduling plant operations. It occurs when the rosterer or shift supervisor need to allocate tasks with specified start and end times to available personnel who have the required qualifications. Furthermore, it also occurs in situations where tasks of fixed times shall be assigned to machines. Decisions will then have to be made regarding the amount of maintenance workers needed and which machine the workers are assigned to look after Krishnamoorthy and Ernst (2001).

There are numerous variants to the PTSP. Studies on these have been made in article Krishnamoorthy and Ernst (2001) by Krishnamoorthy et al. who gives a list of attributes that commonly appear in a PTSP and which are listed in Table 2.1 below. There are furthermore traits that always appear in a PTSP;

tasks with fixed start and end time are to be distributed to staff members that possess certain skills, allowing them to perform only a subset of the available tasks. Start and end time of their shifts are also predetermined for each day.

One variant, which also is the most simple, is mentioned in Krishnamoorthy and Ernst (2001) and is called the *Feasibility Problem* where the aim is to just find a feasible solution. This requires that each task is allocated to a qualified and available worker. It is also required that a worker cannot be assigned more than one task simultaneously as well as tasks cannot be pre-empted, meaning that each task has to be completed by one and the same worker.

In Table 2.1 one can see attributes of PTSP variants. The nomenclature of the attributes T, S, Q, O refer to the *Task type*, *Shift type*, *Qualifications* and *Objective function* respectively.

Table 2.1: PTSP variants

Attribute	**Type**	**Explanation**
T	F	Fixed contiguous tasks
	V	Variable task durations
	S	Split (non-contiguous) tasks
	C	Changeover times between consecutive tasks
S	F	Fixed, given shift lengths
	I	Identical shifts which are effectively of infinite duration
	D	Maximum duration without given start or end times
	U	Unlimited number of shifts of each type available
Q	I	Identical qualification for all staff (homogeneous workforce)
	H	Heterogeneous workforce
O	F	No objective, just find a feasible schedule
	A	Minimise assignment cost
	T	Worktime costs including overtime
	W	Minimise number of workers
	U	Minimise unallocated tasks

Many of the most basic problems and a few more complex ones can be described with this definition of PTSP attributes. It is, however, not possible to describe all of the numerous types of PTSP using these nomenclatures Krishnamoorthy and Ernst (2001).

By combining attributes it is possible to obtain more complex variants of the PTSP. An example would be the PTSP[F;F;H;A-T-W] mentioned in Krishnamoorthy and Ernst (2001) where multiple objectives are used. This problem has fixed contiguous tasks, fixed shift lengths, heterogeneous workforce and three objective functions; A-T-W, which represent assigment costs, work time with overtime included and requirements to minimize the number of workers respectively. For this problem the objective function is then a linear combination with different parameters used to prioritize (weigh) them against each other.

Given the nomenclature above, our problem would be most related to the PTSP[F;F;H;F]. The difference is that the objective function is not empty. We are looking to maximize the number of qualified stand-ins each day as well as maximize employee satisfaction by meeting their recommendations. Since we have a fix number of workers, no costs and no unallocated tasks when a feasible solution is found, this cannot be described with the type of objective attributes given in Table 2.1. Therefore, none of the objective function types are relevant in our case.

Different variants of PTSP are given names in the literature. An example is when the shifts and qualifications are identical (S=I and Q=I) and the objective function is to minimize the number of workers that are used (O=W). This variant, PTSP[F;I;I;W], has been published as the *"fixed job schedule problem"* and is described in Section 2.4 Krishnamoorthy and Ernst (2001).

2.1.1 Applications

An example where PTSP can be found is when developing a rostering solution for ground personnel at an airport. Such a problem can be dealt with by first assigning the workers to days in order to satisfy all the labour constraints, followed by assigning the tasks to the scheduled workers Krishnamoorthy and Ernst (2001).

Three problems of type PTSP related to airplanes can be found when scheduling for either airport maintenance staff, planes to gates or staff that do not stay in one location, such as airline stewards. Scheduling for airport maintenance staff can lead to either PTSP[F;I;H;U-A] or PTSP[F;I-U;H;W], which are similar problems but are given two different names; Operational Fixed Interval Scheduling Problem and Tactical Fixed Interval Scheduling Problem respectively. These are described further in Section 2.4 Krishnamoorthy and Ernst (2001).

Another application, which has been frequently studied, is classroom assignments and is discussed in Krishnamoorthy and Ernst (2001). Based on specifications such as the amount of students in a class or the duration of a class, different classrooms have to be considered. Requirements of equipment, e.g. for a laboratory, may also greatly limit the available classrooms to choose from. A majority of the complications of this problem is due to the fact that lessons can span over multiple periods.

Worth noting for classroom assignment problems is that there are no start or end times for the shifts, as they represent the rooms. The aim in the present problem would be to simply find a feasible assignment of classrooms. Therefore the nomenclature of the problem would be PTSP[S;I;H;F], with the possibility of adding preferences to the objective function. An example of a preference would be to assign the lessons as close to each other as possible on a day, preventing traveling distances between classes for teachers and students Krishnamoorthy and Ernst (2001).

2.2 Shift Minimisation Personnel Task Scheduling Problem

A close relative to the PTSP is the Shift Minimisation Personnel Task Scheduling Problem (SMPTSP) and is a special case in which the aim is to minimize the cost occuring due to the number of personnel (shifts) that are used. The same common traits are valid in this problem as in the PTSP; workers with fixed work hours are to be assigned tasks, with specified start and end times, that they are qualified for Krishnamoorthy et al. (2012).

In article Krishnamoorthy et al. (2012) they "... concentrate mainly on a variant of the PTSP in which the number of personnel (shifts) required is to be minimised.". In doing so, it is possible to determine the lowest number and mix of skilled staff a company should have to be able to complete the tasks and still be operational. They also presumed that the pool of workers are unlimited for either skill group, which is not the case in our problem due to the limitations on the amount of librarians and assistants available.

SMPTSP can be applied when there are a large number of workers available with different qualifications and it is needed to ensure that the tasks for that day are performed. The PTSP and SMPTSP are therefore useful day-to-day management tools that commonly occurs in many practical instances where tasks are allocated on a daily basis Krishnamoorthy et al. (2012).

It is shown in Kroon et al. (1995) that SMPTSP is a complex problem even if the preemption constraint were to be removed. However, if the qualifications of the workers were identical it would become an easily solvable problem Krishnamoorthy et al. (2012).

SMPTSP is almost identical to another problem introduced by Kroon et al. which is called the Tactical Fixed Interval Scheduling Problem and is described in Section 2.4 below Krishnamoorthy et al. (2012).

During the last decade, a couple heuristics have been implemented to deal with the SMPTSP. One method introduced by Krishnamoorthy et al. (2012) is a Lagrangean relaxation approach that combines two problem specific heuristics; Volume Algorithm (VA) and Wedelin's Algorithm (WA). These heuristics exploit the special structure of the SMPTSP by relaxing some of the harder constraints into the objective function, thus being problem specific heuristics. All that remain after the relaxation is a problem decomposed into several independent problems that can be solved independently. A solution to these decomposed problems is discussed further in Krishnamoorthy et al. (2012).

Another method to solve the SMPTSP is to use a very large-scale neighbourhood search algorithm, which was presented by Smet and Vanden Berghe (2012). The main purpose of their implemented hybrid local search is to repeatedly fix and optimize to find neighbouring solutions. Using this method on 137 benchmark instances introduced by Krishnamoorthy et al. (2012), Smet and Vanden Berghe managed to find 81 optimal solutions compared to Krishnamoorthy et al. who only managed to find 67. Furthermore, both methods found feasible solutions for 135 out of the 137 problem instances. This comparison is presented in Smet et al. (2014).

Smet et al. (2014) introduced a third and most effective method to solve the SMPTSP. By using a versatile two-phase matheuristic approach solutions to all 137 benchmark instances could be found for the first time. The procedure used in their implementation is to; first generate an initial solution by using a constructive heuristic, followed by improving the solution using an improvement heuristic. Which heuristics they considered can be seen in their article by Smet et al. (2014).

2.3 Tour Scheduling Problem

The Tour Scheduling Problem (TSP) is described by Loucks and Jacobs (1991) as a combination of shift scheduling and days-off scheduling. Using the notation used described in the section 2.1, this would be be classified as a variant of the PTSP[F;D;I;W] or PTSP[F;D;H;W]. Shift scheduling refers to creating sets of contiguous hours during which a worker is assigned for work. The need for days off scheduling typically occurs when the time horizon for scheduling is weekly or more and when weekend staff is needed.

According to Loucks and Jacobs (1991), the vast majority of all tour scheduling problems up to 1991 involved a homogeneous workforce, that is, any worker can perform any assigned task. One such early study of the our scheduling problem often mentioned in literature is provided by Roberts and Escudero (1983a). The problem studied in this PhD thesis concern only homogeneous work forces and the task assignment part is lacking.

In the article by Loucks and Jacobs, the authors study a tour scheduling problem with a heterogeneous work force. The problem both involves tour scheduling and task assignment, where the latter part is most interesting to us. The problem is studied in the context of fast food restaurants, where certain personnel is qualified only for certain stations in the restaurant. In such industries, the demand of staff differs between different weekdays and different times of the day. Two worker attributes are considered; their availability for work and their qualification for performing different tasks. The problem concerns finding shifts for all workers which are to have a lenght between a minimum and maximum number of hours per day.

The representative problem studied in the article involves creating a one-week schedule for 40 workers in a fast food restaurant, available for eight different tasks with a seven-day, 128-hour workweek. Several synthetic problems are studied in the article, all, however with minimum shift length three hours, maximum shift length eight hours and five maximum number of work days.

A similar problem to the one descibed by Loucks and Jacobs is studied by Choi et al. (2009). They focus on a particular fast food restaurant in Seoul, which is made a representative of fast food chains in general. In this study, only two types of workers are available; fulltime and part time workers, with no other reference to difference in skill. The different shifts are already given by the reastaurant managers and the task is to combine them into a tour. The task assignment aspect is lacking in this article.

In both articles the main objective is to minimize both overstaffing and understaffing, which will both have economical consequences for the fast food chain. This is done by redusing or increasing the work force. For a problem with a fixed work force, such as ours, this objective is not relevant. In the example studied by Loucks and Jacobs there is also a goal to meet staff demand on total working hours. This is modeled as a secondary goal and is similar to our goal and somehow models a ”soft” value, which is of interest to us.

A more recent tour scheduling problems concern monthly tour scheduling, as opposed to most literature which concerns only weekly scheduling. Such a study was done by Rong (2010). The main advantage of monthly scheduling over shorter time periods, as stated in the article, is the possibility to plan a schedule with respect to fairness and balance over a longer period of time. The problem concerns workers with different skills, where each worker also can possess multiple skills. This is referred to as a mixed skill problem. Thus the problem is similar to our problem, where mixed skill is also present. In the study, workers have individual weekend-off requirements. The problem does not involve task assignment, which makes it less relevant for us.

The solution methods used to solve the tour scheduling problem differs greatly between the articles studied. In the older articles, such as Roberts and Escudero (1983a) and Loucks and Jacobs (1991), custom made algorithms very similar to the methods used in manual scheduling are proposed to solve the problem. These solution methods involve classifying staff and distributing them according to some rule (for example, the staff with the most scarce skill is assigned first). General commercial solvers are not proposed, due to lack of efficiency during these times. In Hojati and Patil (2011), the same model and data is used as in Loucks and Jacobs (1991). Also this article states that commercial solvers are insufficient for solving the problem. Instead two methods are proposed, one which decomposes the problem into two problems solvable by commercial solvers and one customized heuristic method based on the Lagrangian relaxation method. in Choi et al. (2009), a pure integer programming method is used.

2.4 Other similar problems

In this section a couple of problems similar to our own will be described in order to give clarity as to how closely related many of these problem types are.

2.4.1 Fixed Job Schedule Problem

Variations of the task assignment problem relevant for our problem include for example the Fixed Job Schedule Problem (FJSP). The FJSP has been studied since the 1970s in the context of task assignment in processors. The problem concerns the distribution of tasks with fixed starting and ending times over a workforce with identical skills, such as processing units Krishnamoorthy et al. (2012). Such problmes have been solved by Gertsbakh and Stern (1978) and Fischetti et al. (1992).

In the article by Gertsbakh and Stern (1978) a situation where n jobs need to be scheduled over an unlimited number of procesors is studied. The objective function of such a problem becomes to minimize the number of machines needed to perform all tasks. Fischetti solves a similar problem, but adds time constraints, saying that no processor is allowed to work for more than a fixed time T during a day as well as a spread time constraint forcing tasks to spread out with time gap S over a processor.

2.4.2 Tactical Fixed Interval Scheduling Problem

Another type of problem is the Tactical Fixed Interval Scheduling Problem (TFISP). This is a problem very closely related to the SMPTSP problem with the only difference being that the TFISP concerns workers which always are available, such as industrial machines or processors. The problem is studied by for example Kroon et al. Kroon et al. (1995). A typical TFISP can be expressed using the nomenclature in Table 2.1 and written as PTSP[F;I-U;H;W] Krishnamoorthy and Ernst (2001).

As opposed to the FJSP, the TFISP deals with a heterogeneous workforce. Two different contexts are studied by Kroon et al. One of them concerns the handling of arriving aircraft passengers at an airport. Two modes of transport from the aeroplane to the airport are investigated; directly by gate or by bus.

The two transportation modes thus correspond to two processing units which can only handle a number of jobs at the same time.

2.4.3 Operational Fixed Interval Scheduling Problem

The Operational Fixed Interval Scheduling Problem (OFISP) is a close relative to TFISP, where both types are restricted by the following; each machine (worker) cannot handle more than one job at a time, each machine can only handle a subset of the jobs and preemption is not allowed. The difference between them occurs in the objective function, as TFISP tries to minimize the number of workers while OFISP tries to minimize the operational costs and the number of unallocated tasks Kroon et al. (1995). In the present nomenclature this would give rise to the problem PTSP[F;I;H;U-A] Krishnamoorthy and Ernst (2001). Given the problem definition above, working shifts are to be created for the workers and tasks are to be allocated on a day-to-day basis. OFISP can therefore be seen both as a job scheduling problem and a task assignment problem Kroon et al. (1995).

2.4.4 Stochastic job problems

What differs mostly between the problem types described above and the problem studied in this thesis work, is the difference in objective. The main objective is often to minimize staff for a number of fixed jobs, not taking into consideration the need for stand ins to perform critical tasks. An area where the need for stand in personnel appears is in the maintenance industry, where some jobs can be forseen and other (emergency) jobs are of a stochastic nature, that is, there is a probability that such jobs will occur a certain hour. The problem combing both unplanned and planned maintenance worker scheduling was studied by Duffuaa and Al-Sultan (1999), as a continuation of Roberts and Escudero (1983b).

In the article, a fixed, heterogeneous work force consisting of electricians, plumbers and mechanics is studied. The shifts of the personnel are predetermined by their given work times and thus the problem becomes a pure task assignment problem. An objective function is used where the goal is to maximize the number of planned and unplanned jobs performed by the workers, by taking into account the probability of unplanned work to occur. Thus, certain workers will be left at the station as stand-ins in the case an unplanned job arrives.

2.5 Problems with soft constraints

For most scheduling problems, the main objective is to minimize worker-related costs by reducing the number of workers needed to perform a task, or by reducing the working time for part-time employees. Equivalently, the goal in production industries is to reduce the number of machines needed. Recently, however, many studies have started to focus more or softer values such as worker satisfaction as an objective. Such values are usually considered when scheduling is done manually, but have been forgotten or set aside in mathematical modeling.

In an article by Akbari et al. (2013) a scheduling problem for part-time workers with different preferences, seniority level and productivity is investigated. In this article, these aspects are reflected in the objective function and weighted against each other. A similar problem was also studied by Mohan (2008), but for a work force of only part-time workers .

Other factors which may affect worker satisfaction, and in the long run efficiency and presence at work are fatigue, fairness and boredom. These are discussed by Eiselt and Marianov (2008). Repetitiveness of a job as well as the level of challenge can cause bordom in workers. Increasing variance is done by Eiselt and Marianov (2008) through providing an upper bound of how many tasks can be performed in a given time span. The article suggests some sort of measurement of the distance between the task requirements and the worker abilities is used. This will then be minimized in the objective function.

Another modeling method which is relevant specifically for scheduling problems featuring soft constraints is fuzzy goal programming. The method is discussed by Shahnazari-Shahrezaei et al. (2013), who model soft constraints as "fuzzy goals". These goal can become contradictory, for example could a preference of high seniority level workers come in conflict with a preference in working hours by an employee. The article uses fuzzy set theory, and uses a solution approach involving Li's two-phase method, Lee (1989). Soft constraints are modeled as trapezoid functions and an optimal solution which finds the best average of all of them is found.

The solution methods proposed by Akbari et al. (2013) to solve a scheduling problem with soft constraints are two matheuristics; Simulated Annealing (SA) and Variable Neighbourhood Search (VNS). According to Akbari et al. (2013), SA has been studied as a solution method for the scheduling problem since the 1990's and many studies have shown that it is capable of providing near-optimal solutions in a short time compared to optimal integer-programing models for a variety of problems. The method is a random meta heuristic, which finds new random solutions and chooses it to be the current solution if sufficiently good. With time, the threshold for what is considered good is increasing.

VNS is the other proposed method by Akbari et al. (2013). A big difference between this and other methods is that VNS requires very little parameter tuning while often providing good solutions. The method uses involves a random and a systematic phase. In the random phase, new shifts are generated randomly and better solutions are saved. In the systematic phase, two shifts are swapped and again better solutions are saved.

In the article by Eiselt and Marianov (2008), a commercial solver was used. This was also the case in Mohan (2008), although the article compares these results with the results obtained from a branch-ans-cut algorithm. Also Shahnazari-Shahrezaei et al. (2013) uses a commercial solver.

2.6 Summary

Stycke ett: Modeller
Stycke två: Metoder
historiskt kronologiskt

2.7 Relevance to our problem

Kan börjas med syftesformulering. Referera tillbaka i texten, samt lägg till nya reflektioner.

2.8 Solution Methods

Integrate into previous parts!

In many real life situations, the scheduling method used to create worker schedules is a simple matching algoritm between two can do what and when. The process is most often left in the hands of experienced and knowledgable schedulers, who know the capacity of the work force and how to maximize productivity by meeting task demands as well as employee demands and individual personality traits. This is referred to as the "art" of scheduling Roberts and Escudero (1983b). However, when personnel forces grow large and there are regulations, task skill requirements or several personnel preferences to take into account, the problem becomes too large to solve manually in a satisfactory manner.

The first computational methods used for solving scheduling problems were in many cases simple heuristics resembling the scheduling process as performed in a manual way. One example of this is the heuristic presented by Loucks and Jacobs which assigns workers to tasks, following certain rules, until all tasks are assigned Loucks and Jacobs (1991). An overview of solution methods is given by Ernst et al, where almost 30 different methods are presented and it is not uncommon that special purpose algorithms are used to suite a specific problem Ernst et al. (2004). Some of the more interesting solution methods with respect to the probelm studied in this thesis are discussed in this section. These include solving with commercial solvers, matheuristic methods such as simulated annealing and variable neighbourhood search, pure heuristic methods, goal programming and fuzzy goal programming.

2.8.1 Commersial software

Commersially available scheduling programs.

2.8.2 Mathematical Programming

Formulating a mathematical model. Objective function and constraints. Solving using commersial solver such as CPLEX or Guroby.

Stochastic and non-stochastic

2.8.3 Simulated Annealing

Simulated Annealing (SA) has been studied as a solution method to the scheduling problem by researchers such as Brusco and Jacobs in the early 1990's. The method is a metaheuristic method which has the advantage over local search methods that it does not easily get stuck in local optima. The method is a random optimization method designed to find a global optimum solution. The method allows bad moves according to a function ... . According to Akbari, Simulated Anneahling

2.8.4 Variable Neighbourhood Search

Also avoids local optima.

"A variable neighborhood search based matheuristic for nurse rostering problems" Della Croce et Salassa. "VNS outoperforms exact commercial general purpose solvers" Matheuristic approach!

Early work by: Hansen and Mladenovic, 2001, Mladenovic and Hansen, 1997

2.8.5 Tabu Search

Commonly used meta-heuristic.

2.8.6 Goal programming and Fuzzy Goal Programming

GP: Used for multiple goals.
FGP: Used for contradictory goals.

Bellman and Zadeh's max-min operator!

Fuzzy goal programming. "Fuzzy goals" = soft constraints. Fuzzy set theory. The basic idea of FGP is to present some of the model parameters as imprecise numbers. Goal programming: good when combining soft and hard constraints.

Using an average value approach with goals that are contradictory makes it possible to maximize the amount of "goodness" in the solution, by priotritizing one constraint over another, which in total generates the most good.

Chapter 3

The mathematical model

In this chapter the mathematical model implemented to solve this problem will be presented. Prior to the objective function and constraints, the most significant variables and sets will be stated to give the reader an idea of how the work has proceeded. In Section 3.3 the constraints will be presented only in pseudocode. Complete model with the full set of constraints can be found in Attachment **??**.

3.1 Set and variable definitions

To solve the problem many sets and variables had to be declared as there are many unique and personal requirements that have to be met. An example is that some personnel want a day free from outer tasks so that they can focus on other assignments or attend meetings. Another one is that some have two different schedules whether it is an odd or even week. These specific cases have to be modeled and result in a variety of set and variable definitions. Hence, only the most important ones are listed below. A complete list of the definitions can be found in Appendix A.

I	Set of workers
I_lib	Set of librarians (I_lib $\subseteq$ I)
I_ass	Set of assistants (I_ass $\subseteq$ I)
W	Set of weeks
D	Set of days in a week
S_d	Set of shifts day d
J_d	Set of task types day d

In order to further define the problem we introduce the following variables.

Let,

$$x_{iwdsj} = \begin{cases} 1, & \text{if worker } i \text{ is assigned in week } w\text{, day } d\text{, shift } s \text{ to a task } j \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

$$H_{iwh} = \begin{cases} 1, & \text{if worker } i \text{ works weekend h in week } w \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

$$r_{iw} = \begin{cases} 1, & \text{if worker } i \text{ has its scheduled rotated } \textit{w-1 steps} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

$$lib_{iwd} = \begin{cases} 1, & \text{if librarian } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

$$ass_{iwd} = \begin{cases} 1, & \text{if assistant } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

$$y_{iwds} = \begin{cases} 1, & \text{if worker } i \text{ is working } w \text{ day } d \text{ regardless of task type} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

$$hb_{iw} = \begin{cases} 1, & \text{if assistant } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

$$friday\_evening_{iw} = \begin{cases} 1, & \text{if assistant } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

$$lib\_min = \text{lowest number of stand-in librarians found (integer)} \tag{3.9}$$

$$ass\_min = \text{lowest number of stand-in assistants found (integer)} \tag{3.10}$$

Based on these defined variables it has been possible to solve our scheduling problem. *lib_min* and *ass_min* are the variables of most significance as they represent the number of stand-ins found after a run.

3.2 Objective function

Due to multiple objective functions it has been necessary to weigh them against each other using parameters. These are shown in Equation 3.11 below, where *N1l*, *N1a* and *N2* are the parameters.

$$\begin{aligned} & maxN1l * lib\_min \\ & \quad +N1a * ass\_min \\ & \quad -N2 * \sum_{i \in I} \sum_{w=1}^{5} \sum_{d=1}^{5} \sum_{s=1}^{3} shift\_differ\_weeks_{iwds} \end{aligned} \tag{3.11}$$

If $N1a > N1l$ then the function were to prioritize assistants as stand-ins rather than librarians. Assistants are, however, less desired as stand-ins, due to their lack of skill to perform all task types. Therefore, it is desired to set $N1a \leq N1l$.

3.3 Constraints

To model this problem it has been of relevance to divide many of the constraints into weekend- and weekday constraints. Several help constraints have also been added to avoid quadratic variables. These would otherwise occur when two decision variables are multiplied with each other.

Chapter 4

The ideal CSTR: the chemostat

In this chapter we study exponential growth, the logistic. . . .

4.1 Some simple models of biological growth

4.1.1 Exponential growth

If $\mu = \text{constant} > 0$, we get $X(t) = X_0 e^{\mu t}$.

4.1.2 The logistic equation

Let us assume that $\frac{dX}{dt} = \mu \cdot X$, with $\mu = \mu(S) = k \cdot S \ldots$

$$\begin{cases} \frac{dX}{dt} = kSX & (a) \\ \frac{dS}{dt} = -\alpha kSX & (b) \end{cases}$$

$$\frac{dX}{dt} = r(1 - \frac{X}{B})X \tag{4.1}$$

An explicit solution to (4.1) is: $X(t) = \frac{X_0 B}{X_0 + (B - X_0)e^{-rt}}$, if $0 < X_0 < B$. It can be found by separating variables in equation (4.1)

4.2 The chemostat

A chemostat is made of two main parts; a nutrient reservoir, and a growth-chamber, reactor, in which the bacteria reproduces.

$$\begin{cases} \frac{dX}{dt} = \mu(S)X \overbrace{-X\frac{F}{V}}^{\text{new}} \\ \frac{dS}{dt} = -\alpha\mu(S)X \underbrace{-S\frac{F}{V} + S_0\frac{F}{V}}_{\text{new}} \end{cases} \tag{4.2}$$

$$\mathbf{A} = \begin{pmatrix} 0 & \sigma\alpha_1 \\ -\frac{1}{\alpha_1} & -\sigma - 1 \end{pmatrix}$$

The invariant line: conclusions

Model	Monods Chemostat	CSI-CSTR
μ	$\frac{S}{1+S}$	$\frac{S}{1+S+\frac{S^2}{K_I}}$
$\frac{dX}{dt}$	$\alpha_1 \frac{S}{1+S} X - X$	$\alpha_1 \frac{S}{1+S+\frac{S^2}{K_I}} X - X$
$\frac{dS}{dt}$	$-\frac{S}{1+S} X - S + \alpha_2$	$-\frac{S}{1+S+\frac{S^2}{K_I}} X - S + \alpha_2$
XNC	$S = \frac{1}{\alpha_1 - 1}$	$S = \frac{K_I(\alpha_1-1)}{2} \pm \sqrt{(\frac{K_I(\alpha_1-1)}{2})^2 - K_I}$
SNC	$X = \frac{(\alpha_2 - S)(1+S)}{S}$	$X = \frac{(\alpha_2 - S)(1+S+\frac{S^2}{K_I})}{S}$
limit	–	$K_I \to \infty$

The other three models, the chemostat, the MMI-CSTR and the CPI-CSTR are quite similar in comparison to the CSI-CSTR.

Monods chemostat does not “feel” this inhibition and does not care. . .

This document is an example of BibTeX using in bibliography management. Three items are cited: *The LaTeX Companion* book **?**, the Einstein journal paper **?**, and the Donald Knuth's website **?**. The LaTeX related items are **??**.

Bibliography

Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, Bowie Owens, and David Sier. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1-4):21–144, 2004.

Mohan Krishnamoorthy and Andreas T Ernst. The personnel task scheduling problem. In *Optimization methods and applications*, pages 343–368. Springer, 2001.

Mohan Krishnamoorthy, Andreas T Ernst, and Davaa Baatar. Algorithms for large scale shift minimisation personnel task scheduling problems. *European Journal of Operational Research*, 219(1):34–48, 2012.

Leo G Kroon, Marc Salomon, and Luk N Van Wassenhove. Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82(1):190–205, 1995.

Pieter Smet and Greet Vanden Berghe. A matheuristic approach to the shift minimisation personnel task scheduling problem. *Practice and theory of automated timetabling*, pages 145–151, 2012.

Pieter Smet, Tony Wauters, Mihail Mihaylov, and Greet Vanden Berghe. The shift minimisation personnel task scheduling problem: A new hybrid approach and computational insights. *Omega*, 46:64–73, 2014.

John S Loucks and F Robert Jacobs. Tour scheduling and task assignment of a heterogeneous work force: A heuristic approach. *Decision Sciences*, 22(4): 719–738, 1991.

SM Roberts and LF Escudero. Scheduling of plant maintenance personnel. *Journal of Optimization theory and Applications*, 39(3):323–343, 1983a.

Kyuwan Choi, JOHyE HWaNG, and Myoungju Park. Scheduling restaurant workers to minimize labor cost and meet service standards. *Cornell Hospitality Quarterly*, 50(2):155–167, 2009.

Aiying Rong. Monthly tour scheduling models with mixed skills considering weekend off requirements. *Computers & Industrial Engineering*, 59(2):334–343, 2010.

Mehran Hojati and Ashok S Patil. An integer linear programming-based heuristic for scheduling heterogeneous, part-time service employees. *European Journal of Operational Research*, 209(1):37–50, 2011.

Ilya Gertsbakh and Helman I Stern. Minimal resources for fixed and variable job schedules. *Operations Research*, 26(1):68–85, 1978.

Matteo Fischetti, Silvano Martello, and Paolo Toth. Approximation algorithms for fixed job schedule problems. *Operations Research*, 40(1-supplement-1): S96–S108, 1992.

Salih O Duffuaa and KS Al-Sultan. A stochastic programming model for scheduling maintenance personnel. *Applied Mathematical Modelling*, 23(5): 385–397, 1999.

SM Roberts and LF Escudero. Scheduling of plant maintenance personnel. *Journal of Optimization theory and Applications*, 39(3):323–343, 1983b.

Mohammad Akbari, M Zandieh, and Behrouz Dorri. Scheduling part-time and mixed-skilled workers to maximize employee satisfaction. *The International Journal of Advanced Manufacturing Technology*, 64(5-8):1017–1027, 2013.

Srimathy Mohan. Scheduling part-time personnel with availability restrictions and preferences to maximize employee satisfaction. *Mathematical and Computer Modelling*, 48(11):1806–1813, 2008.

Horst A Eiselt and Vladimir Marianov. Employee positioning and workload allocation. *Computers & operations research*, 35(2):513–524, 2008.

Parisa Shahnazari-Shahrezaei, Reza Tavakkoli-Moghaddam, and Hamed Kazemipoor. Solving a multi-objective multi-skilled manpower scheduling model by a fuzzy goal programming approach. *Applied Mathematical Modelling*, 37(7):5424–5443, 2013.

Bum-Il Lee. Multiple objective decision making in a fuzzy environment. 1989.

Appendix A

Problem definitions

A.1 Sets

I	Set of workers
I_lib	Set of librarians (I_lib $\subseteq$ I)
I_ass	Set of assistants (I_ass $\subseteq$ I)
W	Set of weeks
D	Set of days in a week
S_d	Set of shifts day d
J_d	Set of task types day d
I_LOW	Set of librarians available to work in library on wheels
I_free_day	Set of workers that shall be assigned a free weekday per week
I_odd_even	Set of all workers with odd or even weeks
I_weekend_avail	Set of workers available for weekend work

A.2 Variables

$$x_{iwdsj} = \begin{cases} 1, & \text{if worker } i \text{ is assigned in week } w\text{, day } d\text{, shift } s \text{ to a task } j \\ 0, & \text{otherwise} \end{cases} \tag{A.1}$$

$$H_{iwh} = \begin{cases} 1, & \text{if worker } i \text{ works weekend h in week } w \\ 0, & \text{otherwise} \end{cases} \tag{A.2}$$

$$r_{iw} = \begin{cases} 1, & \text{if worker } i \text{ has its scheduled rotated } \textit{w-1 steps} \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

$$lib_{iwd} = \begin{cases} 1, & \text{if librarian } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{A.4}$$

$$ass_{iwd} = \begin{cases} 1, & \text{if assistant } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{A.5}$$

$$y_{iwds} = \begin{cases} 1, & \text{if worker } i \text{ is working } w \text{ day } d \text{ regardless of task type} \\ 0, & \text{otherwise} \end{cases} \tag{A.6}$$

$$hb_{iw} = \begin{cases} 1, & \text{if assistant } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{A.7}$$

$$friday\_evening_{iw} = \begin{cases} 1, & \text{if assistant } i \text{ is a stand-in week } w \text{ day } d \\ 0, & \text{otherwise} \end{cases} \tag{A.8}$$

$$lib\_min = \text{lowest number of stand-in librarians found (integer)} \tag{A.9}$$

$$ass\_min = \text{lowest number of stand-in assistants found (integer)} \tag{A.10}$$

A.3 Parameters

$$N1l = \text{a value to prioritize the amount of stand-in librarians} \tag{A.11}$$

$$N1a = \text{a value to prioritize the amount of stand-in assistants} \tag{A.12}$$

$$N2 = \text{a value to prioritize similar weeks} \tag{A.13}$$

$$avail\_day_{iwd} = \begin{cases} 1, & \text{if worker } i \text{ is available for work week } w \text{, day } d \\ 0, & \text{otherwise} \end{cases} \tag{A.14}$$

$$task\_demand_{dsj} = \text{number of workers required day } d\text{, shift } s \text{ for task type } j \tag{A.15}$$

$$qualavail_{iwdsj} = \begin{cases} 1, & \text{if worker } i \text{ is qualified and available week } w\text{, day } d\text{, shift } s \text{ for task type } j \\ 0, & \text{otherwise} \end{cases} \tag{A.16}$$

$$LOW\_demand_{wds} = \text{number of workers required day } d\text{, shift } s \text{ at the library on wheels} \tag{A.17}$$

Appendix B

The Linearized stability

B.1 The Linearization

$F(x)$, a one-variable function of x can be Taylor-expanded around a fix X. We get $F(X+x) = F(X) + F'(X)x + O(x^2)$. For small perturbations of x around X we get the linearization: $F(X+x) \approx F(X) + F'(X)x$, containing only the constant and the linear terms.

For functions of two variables $F(X+x, S+s)$ and $G(X+x, S+s)$:

$$\begin{cases} F(X+x,S+s) = F(X,S) + F'_X(X,S)x + F'_S(X,S)s + O((x+s)^2) \\ G(X+x,S+s) = G(X,S) + G'_X(X,S)x + G'_S(X,S)s + O((x+s)^2) \end{cases}$$

```
function chemostat_inhibited(alpha1, alpha2, xp0, sp0, xc)
%
%chemostat_inhibited Displays a phaseportrait, nullclines
%   and an Euler-path of an inhibited Chemostat.
%   chemostat_inhibited(alfa1, alfa2, np0, cp0, nc) will run if
%   alpha1 > 1/xc, thus there is a reproduction.
%   alpha2 > 1/(xc*alpha1-1), thus there is sufficient stock-nutrition.
%   xp0 > 0 , you can not have a nonpositive population.
%   sp0 > 0 , you can not have a nonpositive concentration.
%   xc  > 0
%
%   The blue arrows represent the vectorfield.
%   The black lines are two of the three nullclines.
%   The black dotted line is the invariant line (no solution crosses it).
%   The red line is an Eulerpath, starting in + and ending in *.
%
%   Try the following:
%   chemostat_inhibited(5, 3, 0.2, 0.3, 6)
%
%   by Per Erik Strandberg, 2003-2004.
%

% Start-condition:
%------------------
if ((alpha1>1) & (alpha2>0) & (sp0>0) & (xp0>0) & xc>0),

    if (alpha2<1/(alpha1-1)),
        disp(' ')
        disp (' (HINT: Only the trivial steady state, alpha2 is too small...)')
    else
        disp(' ')
        disp (' (HINT: Two steady states, alpha2 is quite large...)')
    end
```

```
% The illegal indata case:
%---------------------------
else
    disp('   chemostat_inhibited.m by Per Erik Strandberg, 2003-2004.')
    disp('   Did not Finish OK. (You used illegal indata.)')
    disp(' ')
    disp('   For syntax help type: help chemostat_inhibited .')
    disp(' ')
end
```