SOLVING NP COMPLETE PROBLEMS WITH MEMETIC ALGORITHMS:

How can Memetic Algorithms be used efficiently to generate timetables for the knowledge center of Info Support?

MASTER THESIS COMPUTING SCIENCE

RADBOUD UNIVERSITY NIJMEGEN

Martijn Ermers

February 13, 2009

Thesis number 599

SUPERVISORS

Computing Science – Elena Marchiori

Management & Technology – Geert Vissers

Second Reader – Peter Lucas

Info Support – Arjan ter Horst & Marian Wagenaar

Abstract

This thesis addresses the specification of a Memetic Algorithm, designed for coping with the timetabling problem. Timetabling problems are known to be NP-complete. The proposed algorithm consists of an adaptive local search algorithm and an adapted selection method. The local search algorithm starts out as a hill climbing algorithm for initial candidates and gradually gets more characteristics of a steepest ascent algorithm for better candidates. The selection occurs within the set of candidates of a crossover pair to ensure variation in the population. Test results show that the algorithm performs well for the specified problem with a very low chance to converge into local optima.

Another part of the thesis addresses the market determination of the courses on offer from the Knowledge Center from Info Support. In this part, the currently used sources and techniques for the market determination are described from the viewpoint of different stakeholders within the company. The measurements Info Support is taking that improve the knowledge of the market for courses are elaborated. This information is accompanied by suggestions how the market determination could be further improved.

Table of Contents

1 Introduction

In the modern world there is a striking cliché called “time is money”. In order to use time efficiently it is necessary to make a schedule for one’s activities. There are multiple advantages of constructing a timetable instead of just starting with an arbitrarily chosen activity. One advantage is that you can make appointments with other people on chosen times. Another advantage is that it is possible to manage your scarce time efficiently to be as productive as possible.

Traditionally timetables are constructed by hand. A human scheduler typically adjusts a single timetable until all restrictions are satisfied. A typical restriction is that a person can be at only one place at a time. This process is a time consuming matter, which is why it is useful to automate it. Another reason for automation is that the complexity of some timetables is exceeding the limits of human capacity.

Solving a scheduling problem is not trivial, even for a computer. If you want to determine the best timetable by brute force, it will take a tremendous amount of computation time. In order to get a timetable faster you can use Memetic Algorithms (MA) [Mitchell, 1994]. These work in a probabilistic fashion and make a tradeoff in quality versus time. If you want to make sure to have the best possible timetable, you need to validate all timetables. In this case, MA will not help in improving the performance. Most of the time you want a timetable which is sufficient to meet your requirements. In these cases, MA has a better performance. More information about the problem domain can be found in chapter 2.

The main goal of this thesis is the determination what characteristics are contributing to an effective and efficient timetable solving MA. The entire MA is evaluated in Part I to create a definition of what characteristics the MA ought to contain. With this definition, a concrete implementation will be made in Part II. This implementation is evaluated on the quality of the generated timetable versus the computation time.

Another goal of this thesis is to determine the market for courses for the Knowledge Center of Info Support. Info Support is further introduced in chapter 8.. The demand for courses is relevant, because an innovative and competitive curriculum provides internal and external advantages. The internal advantage is that the consultants from Info Support have access to the newest technologies available. The external advantage is that new technologies attract more customers than older technologies. This subject will be handled in part III

In part IV, the future research recommendations are proposed.

2 Context

2.1. NP-Complete problems

A NP-Complete problem is a problem that cannot be solved in polynomial time. This means that the calculation time finding the best solution for such a problem rapidly increases when the scope increases. A well-known NP-Complete problem is the three-dimensional matching problem. Three-dimensional matchings are important in scheduling problems.

2.2. Scheduling problems

Scheduling problems are optimization problems. This means that there are multiple possible answers to the problem, but the challenge is to find the best suitable solution. The amount of restrictions in a schedule is proportional with the difficulty of obtaining a solution that meets these restrictions. It can also occur that there is no solution, meeting all restrictions. It is desirable to know the one that fits best.

For instance, solve a timetable creation problem involving the parameters: work periods, workers, tasks. The problem with scheduling is that the choices you make for each parameter influence other parameters as well. If workers have a day off this leads to a limitation concerning a combination of work periods and workers. Not all workers can perform all tasks, so not all worker-task combinations are valid either.

The total scheduling problem becomes rather complex because of this. If you let a computer solve this problem the conventional way, you have to check all combinations of parameters in the worst case. This is because if you adjust one parameter, it can interfere with another parameter. Therefore, it is not possible to adjust each variable a single time to its best value. Every change of a parameter can make another parameter become sub-optimal. This behavior makes it a NP-Complete problem. For a formal determination of the complexity of timetable generation see [Even, 1975].

2.3. Genetic Algorithms

John Holland developed Genetic Algorithms (GA) [Holland, 1975]. Genetic algorithms are a search technique used in computing to find approximate solutions for problems that are NP-Complete.

2.3.1. The reason for Genetic Algorithms

If you want to be 100% sure you have the best possible solution to your problem, you must pay the price for computation time needed for evaluating all solutions. Most of the time people are satisfied with solutions that are good enough, if it saves them a huge computation time. Genetic Algorithms are useful in this situation.

2.3.2. The idea behind Genetic Algorithms

A genetic algorithm mimics natural evolution. In nature, each creature consists of DNA. All life adjusts to its environment in order to become better at surviving. The most suitable species reproduce and occasionally mutate into other species. The least suitable species become extinct and disappear from the eco system. This is a continuous process, which keeps making creatures better suitable.

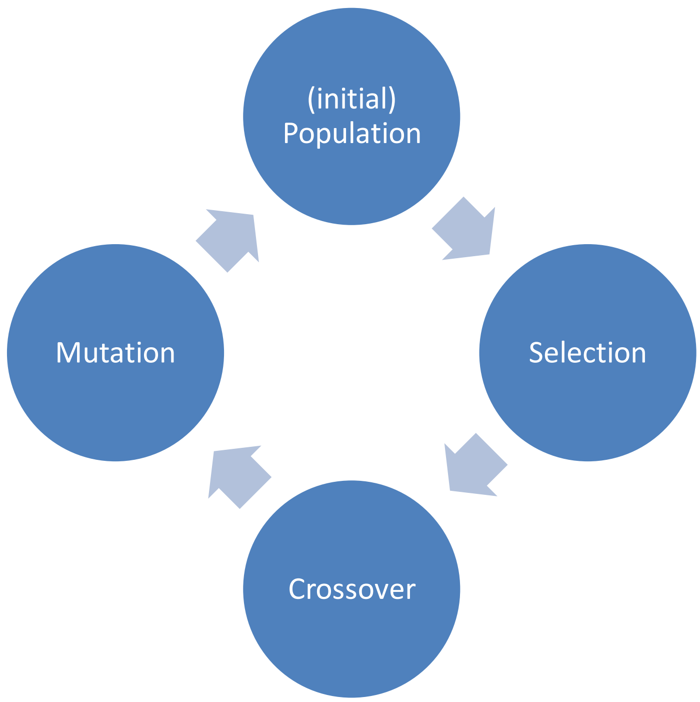

Figure 1 – Genetic Algorithm

The Genetic Algorithm proposed by Holland mimics this behavior, by using a bit string as DNA and translating ability of surviving into rate of meeting requirements (short: fitness value). Each bit string represents a possible solution to your problem.

In the case of a timetable problem, a bit string represents a timetable. The fitness value of this timetable is determined by the rate of meeting the restrictions set by the user of the Genetic Algorithm.

Like in nature there are multiple candidate solutions available at all times in a Genetic Algorithm. In nature reproduction only happens intra species, while in Genetic Algorithms the reproduction will be done inter species.

In our case, all candidate timetables can reproduce with any other candidate. Mutation can happen to a child that has been formed with reproduction. The chance of mutation is rather limited.

In general, the overall quality of the population tends to increase each iteration step. However, the algorithm proposed by Holland has the disadvantage that new candidate solutions are formed without using domain knowledge. The reason the algorithm works is that better candidate solutions have a bigger chance of surviving, the chance of making a suitable candidate when using characteristics of two more suitable candidates also tends to be higher than using characteristics of two inferior candidates.

2.4. Memetic Algorithms

Hybrid Genetic Algorithms or Memetic Algorithms (MAs) are Genetic Algorithms, which use local search techniques during the evolutionary cycle. In short, they reproduce and mutate using heuristics in order to enlarge the chance of getting better candidate solutions. The more heuristics one uses in a MA the more specialized the algorithm gets. So using heuristics limits the problem domain.

Beside limiting the problem domain one should be aware that heuristics could be a source of slipping into local optima, which in turn can result in suboptimal results. Especially in MA it is very important to keep the population as diverse as possible, this limits the chance that the entire population ends up in local optima it cannot get out. More information about the performance of heuristic approaches and genetic algorithm approaches on different landscapes can be found in [Mitchell, 1994].

2.4.1. From heuristics to Genetic Algorithms

Since a Memetic Algorithm is a hybrid between heuristics and GAs, you can vary the balance of using heuristics in your algorithm. In the figure below, you can see how the rate of heuristic narrows the problem domain, but also increases the performance on problems within that domain.

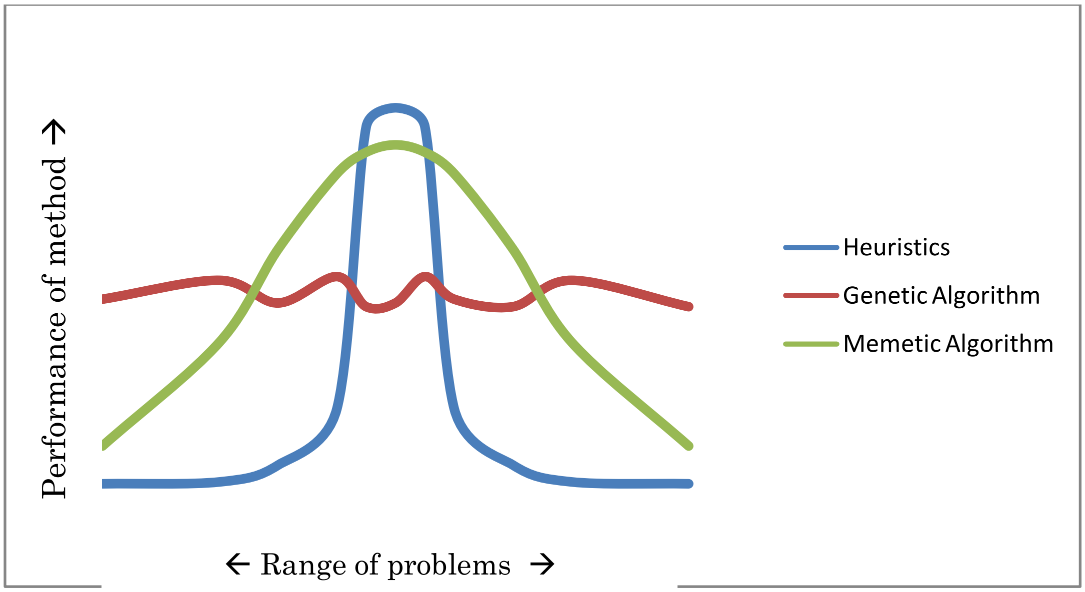

Figure 2 – Heuristics vs. Genetic Algorithm vs. Memetic Algorithm

A Genetic Algorithm is an all-round problem solver that does not have an area of expertise. Since we are going to use the algorithm for a specific domain (generating timetables), it is useful to use some heuristics within the algorithm. Note that the figure seems to imply that heuristics alone have the best possible performance. However, the problem with pure heuristics is that the domain is narrowed to problems without separate local optima. In general, it is plausible that the solution space of timetables contains local optima. Therefore, it is desirable to have at least a minimum amount of GA in your algorithm.

2.5. Multi Threading

Multi threading is a way to calculate multiple tasks at the same time. Nowadays, it is quite common to have multi core processors in pc's. This makes it useful to make a program able to divide its tasks and execute them simultaneously. Since a genetic algorithm performs the same task independently on multiple candidate solutions, it is easy to split these into separate threads. This is easily realized, because the used candidate solutions do not need to be modified (which could make threads intervene with each other). Multi threading is not an objective in this research, but since it is not hard to implement, current multi core processors may just as well be used with their full power.

Part I: Memetic Algorithm

The goal of this part is defining our Memetic Algorithm. This MA is going to be tailored to solving scheduling problems. The definition will be at a conceptual abstraction level: It serves the purpose of getting an overview of what design choices are made. The concrete design, implementation and testing will be addressed in Part II.

This part consists of the determination of the timetable representation, the determination of the input formats, evolutionary operators, replacement strategy and the termination criteria of the Memetic Algorithm.

The general Memetic Algorithm is based on the overview below. This part will define all the aspects mentioned in the overview.

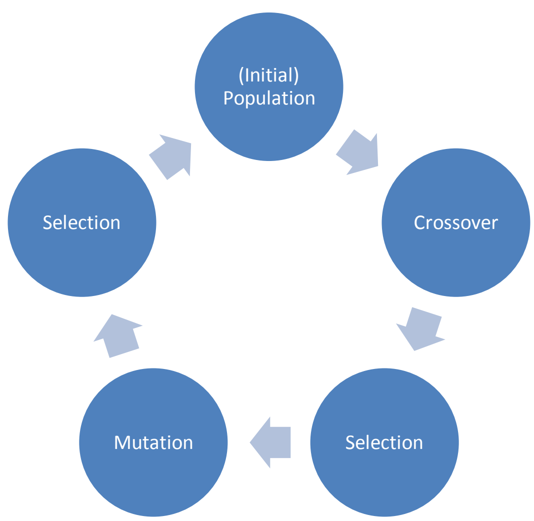

Figure 3 – Overview of the Memetic Algorithm.

In order to keep the algorithm comprehensive for educated people without scientific knowledge of the artificial intelligence domain, the different parts of the algorithm are described in natural language rather than formal language.

3 Timetable representation

The representation of the timetable is one of the most relevant choices to make when you want to use some kind of GA. This decision is important because it has effect on the expressiveness of timetables in general. It is important to maximize the expressiveness, because the algorithm should become able to cope with as many timetable problems as possible.

Perhaps the most important practical property of the representation is the ease of fitness determination of a timetable in that representation. When there is a large population of timetables in a GA, it is relevant that the size of a timetable is within certain boundaries. A timetable is being used in the evolutionary operators, so it is useful that these operators can work efficiently on the chosen representation.

3.1. Expressiveness

In order to keep the algorithm as generally applicable as possible, it will be necessary to make a representation that can describe all relevant timetables. There are massive amounts of possible timetable representations, so it is necessary to set boundaries of what should be representable.

The following definition of timetable will be used in the rest of the thesis. A timetable always has the following core attributes: when, where, what and who.

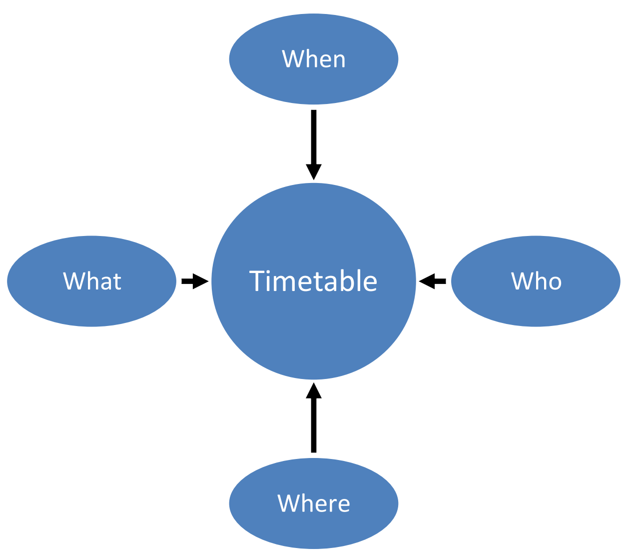

Figure 4 – Overview of a timetable

3.1.1. When

The primary problem with scheduling is *when* something is scheduled. The representation should allow a significant amount of periods. A period is defined as the scope of the timetable. It should be possible to set the period to your wishes, but with boundaries for the sake of efficiency and keeping it countable.

The boundaries are based on human limitations; this is a reasonable thing to do because the timetables are about human resource planning. For a human it is not contributing to have a timetable that has a precision of less than minutes. Because of this limitation it is safe to say that the minimal time can and will be set at a minute. The largest human measure for time is years. There are some larger quantities, but they are easy to calculate based on years. Because of the large capacity for numbers in computers, it makes sense that years make a very large upper bound. Therefore, the maximal time will be set at years.

To keep the algorithm as efficient as possible it makes sense to set the minimal time block for scheduled items. The advantage is that the larger the blocks of the algorithm are the fewer timetables can be generated. The possible disadvantage of making use of large blocks is that some solutions will never be generated, because this limits the amount of freedom for the scheduler. It is up to the user of the algorithm to make this decision.

It is very common in scheduling that not all timeslots can be used. For example, working during the evening or weekends is generally not allowed. It should be possible to set per block whether it is schedulable. These restrictions can also be set for individual persons, locations and courses.

3.1.2. Who

When thinking about scheduling it usually involves people, however it is also possible to make for instance a timetable for a production line involving a variety of products. In this case the products fall under the category 'who' of the timetable. The problem with accepting this kind of scheduling problem is that a batch of products can be produced at multiple product lines. The problem is breaking the basic rule that a person cannot be at multiple places at the same time. This is the reason why the scheduling algorithm is restricted to persons.

There are groups of stakeholders involved in scheduling. A group has certain attributes like "must attend" and "can give". Think for instance about students who must attend specific courses and teachers who can give specific courses. It should be possible to set restrictions like "A teacher must be attached to a course." and "All students must be able to attend a course."

Based on these prerequisites it should be enough to have entities that have the previously described attributes. These entities should be attachable to groups. The relevance of groups is that you can set restrictions like "Each course must have a person from the group teachers attached."

It must also be possible to make a single entity count as multiple persons. This is relevant when there is a group of persons with no relevant specific attributes (like students) and if there are capacity limitations on the locations.

3.1.3. Where

The location is on first sight a rather trivial part of the scheduling. However, there are multiple possible representations of locations. A typical high school has a set of classrooms that are all near each other, but it is possible to have a distance between locations. This makes it impossible to schedule two consecutive things on two distant places, as travel time must be administered in this case.

There may also be restrictions in locations like "Not every course can be given in each location" and "There are capacity limitations on locations".

Finally, it is possible that multiple courses can be given at the same location at the same time. In this case, it must be checked whether the maximum number of courses is not exceeded.

3.1.4. What

Previously, the word course was already mentioned. A course is a possible 'what' in a timetable. In general, it is an appointment between multiple stakeholders. The objective can be information sharing or working together; this does not matter in the scheduling context.

It can occur that a course has to be given a certain amount of times within a period, but it can also be person driven. If there are persons who need to follow a specific course, you can derive from that that the course needs to be given too.

3.2. Fitness Determination

The chosen representation has influence on all operations concerning concrete timetables. The determination of fitness is one of the hardest operations during the evolution process. The reason for the complexity is that not all restrictions can be verified easily.

Take for instance a representation in which courses form the basis with dates, teachers and students connected to them. Now try to determine whether the teacher has to work more than two courses directly after each other. In this case, you must check for each course whether the specific teacher gives that course and if this is true, look at the time. After all times are gathered you can determine how many times the two courses rule is exceeded.

The choice for a suitable representation is dependent of the restrictions you want to use in your algorithm. This problem is similar to the "No Free Lunch" theorem [Wolpert, 1997], which states that there is no single optimum algorithm for multiple problem areas. The resemblance is that no matter which representation you choose the average performance over all sets of restrictions does not increase (on average).

Since the aim is to keep the MA as general as possible the choice of representation should also be an input parameter. The problem with this is that for each possible representation the access functions are different. The 'two courses' problem described above is a nice example of usage of an access function. It should be possible to keep the representation generic. However, this very complex problem does not contribute to my main research. This problem is elaborated in more detail in the Future Research chapter. The next best thing is to choose a representation that suits the scheduling problem of Info Support, with the possibility to implement other representations without having to change anything not directly associated with access functions. In other words, the representation is a black box, only the access function names are known for the algorithm.

3.3. Storage Space

Storage space may form a serious problem when computing with GAs. A GA has the characteristic that all candidate solutions of an iteration step need to be stored in the memory. Depending on the chosen algorithm, the results of the evolutionary operators need to be stored for a minimal time too. Depending on the nature of the problem population, the memory typically contains no more than several hundreds or thousands of candidate solutions during execution time.

Because of this significant amount of candidate solutions in memory, it is important to know how large a typical candidate solution is. In the case of a scheduling problem, the worst-case scenario would be that one would choose a representation with time blocks as a basis. Then every time block has a reference (can be null) to a course (or perhaps even a teacher). When the timetable period is large and the time blocks small, there are a huge amount of null-references which all need memory.

Take for instance a timetable for a year with time blocks of 15 minutes. The references are 32 bit and for the ease of calculation ignore every other attribute in the timetable. There are 35 thousand time blocks within that year. Therefore, each empty schedule cost around 137 KB. When you take a large population pool of ten thousand and keep the results of the evolutionary operators temporary in memory too, you will end up with roughly twenty thousand candidate solutions in memory. The total storage space needed will be around 2.6 GB. When you add other information like courses/teachers/students, you will quickly run out of memory. It is possible to use other memory than RAM, but you will suffer serious decrease in performance.

Depending on the size of the timetable, it is wise to choose a representation that does not suffer from large amounts of null references. The memory usage only becomes an issue when the limit is approached. A representation that uses more memory can perform better than a more economical representation.

3.4. Evolutionary Operators

The evolutionary operators from a standard GA work content independent. With this method, the representation of the timetable does not matter at all. However, a MA is described in this paper. The evolutionary operators perform actions of which is likely that the fitness of the candidate will improve. In order to achieve this improvement it is necessary to know which restrictions are violated. It can be wise to change especially the characteristics that are source of the violation.

Determining which characteristics are sources of restriction violations is used in both the determination of fitness and the determination which action is performed by the evolutionary operator. Like with the fitness determination it is relevant to know what the set of restrictions is.

4 Input Format Determination

The determination of the input format is relevant because the format directly effects the restrictions that can be described. The goal is to make a format, which supports all restrictions used in timetables. The first part is to make a definition which inputs should be used in what way in the timetable. The second part is the determination of all restrictions and defining how they ought to be represented in the input.

4.1. Definition of Input attributes

Like described in the previous chapter there are four main attributes (when, who, where and what). For each of these attributes the input format will be defined below based on the choices made in the previous chapter.

4.1.1. When

The input format must meet the following standards concerning the time aspect:

- Every date and duration is represented in a combination of minutes, hours, days, weeks, months and years.
- Timetable period – Start date and stop date or start date and duration.
- Block size – The size of a time unit, the smallest schedulable period.
- Non-schedulable blocks – Start date and duration, can be followed by repetition time, which is calculated, from the start date.

4.1.2. Who

The purpose of this input format is not only specifying which persons and groups there are, but also the specific time restrictions each person has (in example holidays). A person has the following attributes:

- Identifier – A string of characters that serves as an identifier of the person or persons.
- Multiplier – The multiplier is a number equal to the amount of persons the identity represents.
- List of "can give" what's – A list of courses the person is allowed to give and a priority number which indicates the relative preference that that teacher gives the specific course.
- List of "must follow" what's – A list of courses the person should be able to follow.

- Non-schedulable blocks – Start date and duration, can be followed by repetition time that is calculated from the start date. For dates, a person cannot be scheduled like days off and holidays.
- Schedulable time – In the case of “can give” tasks there is a limit possible for courses he can give, this can also be a minimal and maximal time.

A group has the following attributes:

- Identifier – A string of characters, which serves as an identifier of the group.
- List of Persons – A list of all persons belonging to the specific group.

4.1.3. Where

The locations are specified in a direct manner, but they also belong to groups of locations. Groups are specified in order to make a separation between different locations and different rooms. A location has the following attributes:

- Identifier – A string of characters, which serves as an identifier of the location.
- Multiplier – The multiplier is a number equal to the amount of locations the identity represents (useful in cases they are exchangeable).
- Facilities – All facilities, which are available in the location. This is relevant for course requirements. (for instance desks, a beamer)
- Capacity – The amount of persons possible to be attached to the location.

A group of locations has the following attributes:

- Identifier – A string of characters, which serves as an identifier of the group.
- List of Locations – A list of all locations belonging to the specific group.
- Distance To – A list of all other location groups with a travel time specification.
- Facilities – All facilities, which are available in the location. This is relevant for course requirements.

4.1.4. What

A course has the following attributes:

- Identifier – A string of characters, which serves as an identifier of the course.
- Location demands – A list of needed facilities on the location.

- Has Prerequisites – A list of needed courses before this course can be participated.

4.2. Definition of Restrictions

Restrictions are very common in scheduling. In order to keep the input flexible restrictions can be made in the form of:

- Identifier – A string of characters, which serves as an identifier of the restriction.
- Argument 1 – One or more objects, optionally with a fact specified.
    - Object – The object of which the fact has to be evaluated
    - Fact – The fact which has to be evaluated
- Argument 2 – One or more objects, optionally with a fact specified.
    - Object – The object of which the fact has to be evaluated
    - Fact – The fact which has to be evaluated
- Argument 3 – Optional. A number, time, object or string
- Comparison rule – The rule that has to be used on the two arguments. The rule can pass or fail which indicates the violation of a restriction. The third argument can be used for comparisons that are more elaborate.
- Weight – The importance of the restriction.

The explanation of the comparison rule is not elaborate enough. For instance, it would be great if the algorithm could comprehend a proposed comparison. In order to achieve this, the comparison rule must follow a certain syntax and semantics. The evaluation of this code must be done at runtime of the algorithm, so it is not feasible to make an elaborate input language. The proposed solution consists of a set of operations that can be used; the program checks what operation is selected and then executes the right operation.

4.2.1. Restriction operators

The following operators can be used in the restriction rules:

- Equals – A check whether the two arguments are the same.
- Smaller then – A check whether the first argument is smaller than the second argument.
- Contains – A check whether the collection from argument 1 contains all elements from argument 2.
- Is over scheduled – A check whether the argument 1 is scheduled too many times. Argument 2 points to the desired value. Argument 3 holds an optional threshold for over scheduling.

- Is under scheduled – A check whether the argument 1 is scheduled too few times. Argument 2 points to the desired value. Argument 3 holds an optional threshold for under scheduling.
- Is Better Teacher – A check whether the preference number for teacher given in argument 1 is higher than the preference number for teacher given in argument 2 for the course given in argument 3.
- Are scheduled before – A check whether the courses from argument 1 are scheduled before the course from argument 2.
- Total persons is valid – A check whether the sum of the persons in argument 1 is below the value in argument 2.

5 Evolutionary Operators Determination

The evolutionary operators are responsible for the generation of new candidate solutions in a GA. This gives the evolutionary operators a vital role in the efficiency of the GA. The difference between GA and MA is that MA use heuristics to generate new candidate solutions. A standard GA makes new candidate solutions, which have in general an equal chance to improve or deteriorate. The reason why a standard GA works with these operators is that they work for all problem domains. The natural selection makes that poor results are discarded in evolution. The advantage of using heuristics is that the generated offspring will be better for a specific problem domain.

In this chapter, the two evolutionary operators are modified to increase the efficiency for generation of timetables.

5.1. Crossover operator

The crossover operator is a function, which uses two or more candidate solutions to generate one or more new candidate solutions. The general idea with crossover is using characteristics of each parent candidate solution in order to generate new child candidate solutions. The main idea is recombination features from multiple candidate solutions to obtain new and hopefully better solutions.

In standard GA, the crossover operator works on bit strings. New solutions are formed by using a selection of bits from one parent and the missing bits from the other parent (see picture below). This method is easy to implement since the crossover operator does not know or care for the implications of combinations of bits. The disadvantage of this method is that the fitness of the new candidates is completely random. The answer to this poor generation method is using heuristics.

Parent 1	1	0	1	1	0	1	0	0	0	1
Parent 2	1	1	0	1	1	0	0	1	1	1
Child 1	1	0	0	1	1	0	0	0	0	1
Child 2	1	1	1	1	0	1	0	1	1	1

Figure 5 – Two Point Crossover Example on bit strings

5.1.1. Heuristic Determination

The first step in heuristics is determining what all bits in the representation do. The best way to achieve this is completely release the bit string approach. Let the operators work on the representation as it is intended; use factors like dates/teachers/locations which can be adjusted to different instances of the same factor. In [Alkan, 2003] the writer proposes to do crossover with complete factors or within factors. Since all factors have a narrow scope of legal input it makes sense to do crossover with complete factors. The advantage of this method is that there is no chance of creating non-existing values for the factors.

5.1.2. Heuristic Implementation

Since the format of the representation is defined in chapter 4, the only decision to be made is to choose between single point and multi point crossover. In order to keep the algorithm non-biased it is also possible to decide per factor whether to cross or not. This can be done by chance, the more the chance shifts to 50% the more diversification occurs. When the chance shifts to 0% or 100%, the child solutions will have a higher degree of resemblance with the parent solutions. The figure below shows the resemblance of the child solutions with their parents. On the vertical side, the resemblance level (0% ~ 100%) is indicated. On the horizontal side, the rate of mutation (0% ~ 100%) is expressed. It shows that on 50% the child will (on average) be mostly differed from their parents. The different curves represent different rates of resemblance between the two parents. If the parent solutions are the same, so will be the child solutions. Maximum differentiation occurs with maximum differentiation between parents.

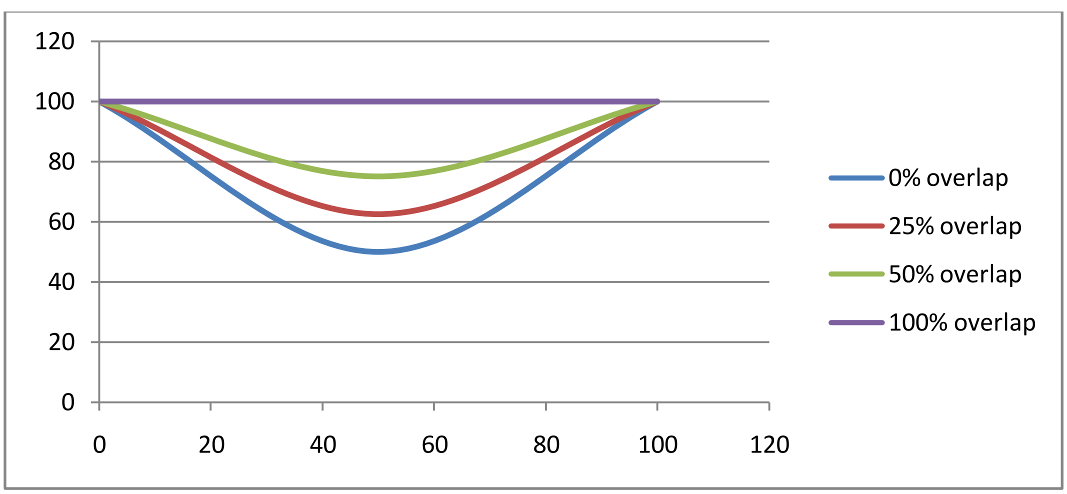

Figure 6 – The overlap between parent and child solutions

The differentiation between parent solution and child solution is proportionate to the chance on improvement and deterioration. It is up to the user to choose the rate of crossover. This rate is typically low, since the chance to adjust multiple factors with a positive effect is low. Note that the chance crossover happens over the entire candidate is much higher than on factor basis. For example the chance that crossover happens in a solution with 20 factors is with a 10% crossover rate 88% (1 – ((1 – 0.1) ^ 20) = 0.88). It is wise to have a near certainty that crossover happens, since not doing crossover makes it possible to discard one parent because the other parent and child are identical.

Another possibility is determining the crossover chance per factor depending on the combined violation rate of the factors of both parents. The advantage of this method is that the fitness of the child solution is likely to be better than the fitness of the parent solution. The downside is that the violations of all factors have to be determined separately, which will cost an enormous overhead to determine. This overhead per factor is of the same order as a normal fitness evaluation. Therefore, the advantage of better offspring is negated by the computation cost.

5.1.3. Pseudo Code

The following pseudo code is implementing the fixed rate crossover operator.

```
Candidate Solution 3 = new Candidate Solution ();
Candidate Solution 4 = new Candidate Solution ();

For (int I = 0; I < Candidate Solution1. Length; I++)
{
            If (Random Number (0% ~ 100%) < Crossover Rate)
            {
                        Candidate Solution3 [I] = Candidate Solution2 [I];
                        Candidate Solution4 [I] = Candidate Solution1 [I];
            }
            Else
            {
                        Candidate Solution3 [I] = Candidate Solution1 [I];
                        Candidate Solution4 [I] = Candidate Solution2 [I];
            }
}
```

As you can see, two new solutions are being generated. The original solutions are needed for selection afterwards. The pseudo code lets the solution three to be similar to solution one, with the assumption that crossover rates are lower than 50%.

The proportionate crossover operator calculates the factor violation level of the factor from both candidates. If the violation level is zero, the chance on crossover is 0%. If the violation level is equal to the maximum factor violation level, the chance on crossover is 100%. The following pseudo code is implementing the proportionate crossover operator.

```
For (int I = 0; I < Candidate Solution1. Length; I++)
{
	If (Random Number (0.0 ~ 1.0) < (Factor Violation Level (Candidate Solution1 [I])
	                                + Factor Violation Level (Candidate Solution2 [I]))
	                                / (2 * Maximum Factor Violation Level))
	{
		Candidate Solution3 [I] = Candidate Solution2 [I];
		Candidate Solution4 [I] = Candidate Solution1 [I];
	}
	Else
	{
		Candidate Solution3 [I] = Candidate Solution1 [I];
		Candidate Solution4 [I] = Candidate Solution2 [I];
	}
}
```

This crossover method is impractical because for each factor the violation level has to be determined. This is a very costly operation, which negates the positive effect of stimulating better factors to survive.

The crossover operator will be used on a percentage of the population of candidate solutions. The other candidate solutions are preserved in the population. More about this will be elaborated in chapter 6.4.

5.2. Mutation operator

The mutation operator is used after the crossover operator. The main purpose of this operator is introducing new values to the candidates. The crossover operator only recombines existing solutions without introducing new values. The mutation operator is, compared to the crossover operator, relatively easy to improve with heuristics. Like with the crossover operator it is desirable to work with factors instead of bit strings. Below both the bit string method and factor approach are visualized.

Before	1	0	1	1	1	1	0	1	0	1
After	1	0	0	1	1	1	0	1	0	1

Figure 7 – Mutation Example on bit strings

Before	Monday	Tuesday	Sunday	Friday
After	Monday	Saturday	Sunday	Friday

Figure 8 – Mutation Example on factors

The advantage of this method is that there is no chance of creating non-existing values for the factors; the bit wise method can create in the example above a non-existing day.

5.2.1. Heuristic Determination

The first step in adjusting the mutation operator is only adjusting the factors that cause restriction violations. Like with crossover it is useful to make the mutation chance proportionate with the rate of violations. Then the choice of substitute value has to be made. It is relevant to use values that are valid in the context.

The decision has to be made whether the best substitute or a random substitute has to be chosen. The problem with choosing the best substitute is that the randomness decreases which could increase the chance of stranding in local optima. Another issue is that you have to evaluate all possibilities before you can make a choice, this can seriously decrease performance.

A well-known selection method is tournament selection. The idea is that the algorithm selects a set of values from the entire set of values. The best of these contestants is selected. The advantage of tournament selection is that it is able to balance between quality, performance and randomness. More information about tournament selection can be found in [Eiben, 2007].

To continue with the same example as in figure the figure below an example of a tournament is displayed below. The tournament size is three in this example. This means that for the chosen factor three alternate values are tried, of which the best one is kept. The best individual in this example is Mutant 1. This individual will survive in the evolution.

					Fitness
Before	Monday	Tuesday	Sunday	Friday	100
Mutant 1	Monday	Saturday	Sunday	Friday	110
Mutant 2	Monday	Friday	Sunday	Friday	70
Mutant 3	Monday	Monday	Sunday	Friday	65

Figure 9 – Example of Mutation with Tournament Selection

In the determination of fitness between different values, it is only necessary to check the relative difference between the values. It is not useful to check on other restriction violations, because they have the same violation level for each of the values.

5.2.2. Heuristic Implementation

In order to keep the algorithm fast tournament selection is an appropriate choice for factor selection. Pick a random set of factors and determine their violation rate. Mutate the factor with the highest violation rate. The replacement strategy can also be done with tournament selection, pick some random alternate values for the factor and determine their violation rate. Pick the one with the lowest violation rate.

The size of the tournaments should be reversed proportionate with the violation rate of the candidate solution. The violation rate of the worst candidate solution evaluated by the algorithm is used as lower bound. The upper bound for violation is set to zero, with the assumption that the best solution possible should approximate to no violations. This relation is visualized below.

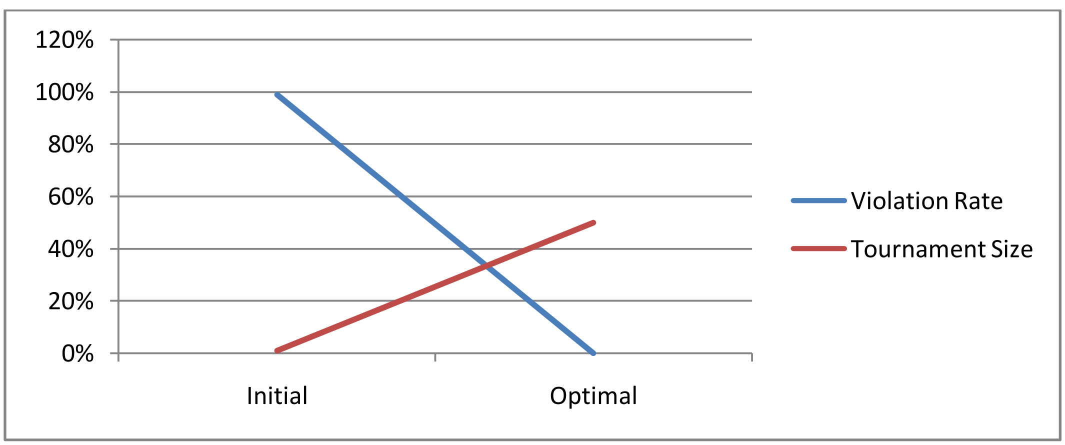

Figure 10 – Relation between Tournament Size and Violation Rate

The result is a hill climber algorithm for worse candidates changing into a steepest ascent algorithm when the optimum solution is approached.

The advantage of this method is that in early stages of evolution the selection pressure is low which decreases the change of early convergence. While in later stages of evolution the tournament size increases which increases the chance of convergence, but that is not a problem at later stages since only minor optimizations are needed at that point.

The maximum tournament size is set at 50% of the factors/values. The reason for choosing 50% is to prevent extreme convergence on fitter candidates. It also saves near 50% computation time compared to the evaluation of almost the entire set of factors/values.

5.2.3. Pseudo Code

The pseudo code of the tournament selection method for factor determination is displayed below. The pseudo code assumes a global variable called "Candidate Solution". Another assumption is that "Factor List" is a list containing all possible values for a type, the only type. The "Tournament Size Factor Determination" is the amount of factors to be evaluated in order to choose the best factor for improvement. This size is between 1% and 50% of the total amount of factors, depending on the fitness of the candidate.

```
Mutate Candidate Solution (Tournament Size Factor Determination)
{
        Number List = Generate Numbers (Tournament Size Factor Determination),
                                        0 ~ Candidate Solution. Length);
        Worst Factor = -1;
        Worst Factor Violation Level = 0;
        For (int I = 0; I < Tournament Size Factor Determination); I++)
        {
                Temp = Factor Violation Level (Candidate Solution1 [Number List [I]]);
                If (Temp > Worst Factor Violation Level)
                {
                        Worst Factor = Number List [I];
                        Worst Factor Violation Level = Temp;
                }
        }
        //The size of the tournament for values is relative to the size of the tournament for factors
        Tournament Size Value Determination = Tournament Size Factor Determination
                                                * Factor List. Length
                                                / Candidate Solution. Length;
        Candidate Solution[Worst Factor] = Mutate Factor (Worst Factor,
                                                Tournament Size Value Determination));
}
```

The "Generate Numbers" function is a unique number generator with (for example) the following pseudo code. Note that this function can iterate endlessly when the boundaries permit less different numbers than the amount needed.

```
Number List Generate Numbers (Amount, Boundaries)
{
            Number List = new Empty List;
            While (Amount > 0)
            {
                        Temp = Random Number (Boundaries);
                        If (Not Number List. Contains (Temp))
                        {
                                    Number List. Add Number (Temp);
                                    Amount--;
                        }
            }
            Return Number List;
}
```

The pseudo code for the “Mutate Factor” function is:

```
Factor Mutate Factor (Worst Factor, Tournament Size Value Determination)
{
            Number List = Generate Numbers (Tournament Size Value Determination,
                                                            0 ~ Factor List. Length);
            Best Factor = Factor List. Get Index (Candidate Solution [Worst Factor]);
            Best Factor Violation Level = Factor Violation Level (Candidate Solution [Worst Factor]);
            For (int I = 0; I < Tournament Size Value Determination; I++)
            {
                        Temp = Factor Violation Level (Factor List [Number List [I]]);
                        If (Temp < Best Factor Violation Level)
                        {
                                    Best Factor = Number List [I];
                                    Best Factor Violation Level = Temp;
                        }
            }

            Return Factor List [Best Factor];
}
```

Note that the selection of the best factor happens inside this function while the functionality is defined separately for the algorithm.

6 Replacement Strategy Determination

The key step in evolution is the selection of the surviving candidate solutions. The main objectives of the selection are keeping the population size limited, improving the average fitness of the population and maintaining diversity. Before the replacement strategy will be determined, all objectives will be defined.

6.1. Population size

Each time the evolutionary operators are used there are a number of new candidate solutions available. A direct result of this is that the population size exceeds the boundary set by the user. The only relevance for this objective is that the amount of candidate solutions is reduced to the level defined by the user.

6.2. Improving fitness

The core functionality of the evolutionary operators is generating new candidate solutions. Since these operators do not always generate better solutions then their parents, it is relevant that the replacement strategy selects candidates who have a higher average fitness than the previous population. A well-known algorithm for improving fitness value is Elitism. The use of Elitism by other researchers can be found in [Őzcan, 2005], [Monfroglio, 1996], [Burke, 2004] and [Alkan, 2003].

6.3. Maintaining diversity

The problem with Elitism is that the population tends to lose diversity. This behavior comes from the fact that most candidates with a high fitness value come from other candidates with high fitness values. The problem with losing diversity is that the chance increases that the algorithm strands in local optima. Avoiding stranding in local optima is the main reason to use genetic algorithms in the first place. Since algorithms like hill climber and steepest ascent are faster than genetic algorithms but have (depending on the problem area) a significant chance run into local optima; these algorithms are not preferable for making timetables because of the significant amount of local optima in the search space. Research to the application of genetic algorithms has been done in [Mitchell, 1994].

It is hard for a computer to determine the similarity between different candidate solutions. By selecting between the child solutions and parent solutions, the advantage is that you know these solutions are alike. One can choose to keep the best of them that takes reasonable good care of similar solutions. The tradeoff between computation time and convergence is quite acceptable.

6.4. Final Determination

The replacement strategy needs to implement the following properties:

- Respect the population size
- Keep the best known solution
- Improve average fitness
- Maintain diversity

In order to keep the algorithm fast and effective a variant of tournament selection is a valid choice. The following algorithm manages to comply with previously defined properties.

Take for each crossover operation all parent solutions and all child solutions (which are mutated at this point). Keep the best solutions of this group. The amount of solutions kept must be the same as the amount of parent solutions. This process is visualized below. It also includes the random operator, which makes different variants of the selected candidate solutions. The random operator also passes the unmodified solution to the selection operator. Note that this process happens for all crossover operations.

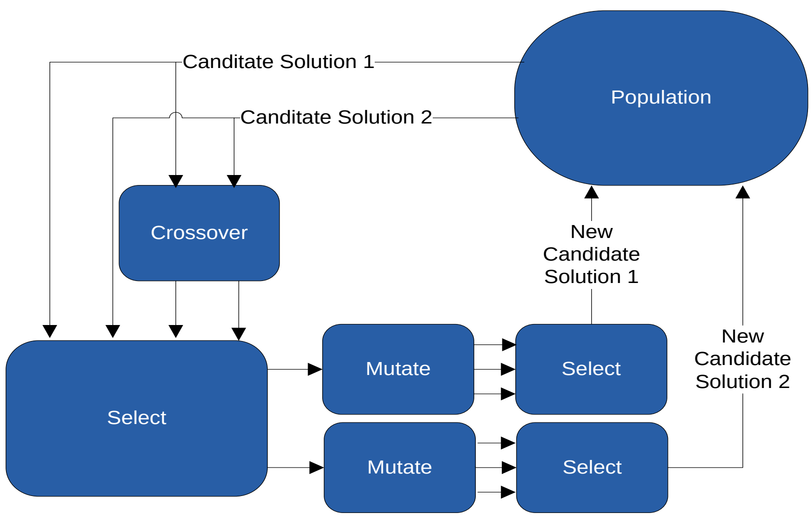

Figure 11 – Integration of the Selection Operators between the Evolutionary Operators

Since the part of the population that gets 'crossovered' generates the same amount of candidates as it removes from the population, the population size stays the same; the parents are exchanged by the best of their offspring (or themselves if they have better fitness). The part of the population that is not used for crossover is preserved for the next iteration.

This algorithm keeps the best-known solution because it is the best one of the crossover operation group. If a child replaced it, the child should be better than the previous best solution. If it was not used in crossover, it is per definition preserved. Note that the best solution is not always on the same spot; if you want to determine the best candidate you have to search the entire population.

This algorithm improves the average fitness because only the best ones of each crossover group are maintained in the population.

The mutation operator determines the parameter to adjust by random, and then it determines by tournament selection the replacement value. The tournament size is determined by the fitness value of the candidate before it is mutated. The reason to choose the factor by random is the increased computation cost of tournament selection which makes that option too expensive. The selection after the mutation is incorporated in the mutation operator. See chapter 5.2.2 for a more detailed description.

The diversity is reasonably guaranteed because the algorithm prefers to release lesser variants of a 'bloodline'; in most cases the parent or the child is being discarded, leaving the case that the crossover chance was very small which could lead to discarding of one parent and his most alike child.

6.4.1. Local Optimum Problem

A potential problem with all selection methods (except for random selection) is the chance of running into local optima. The advantage of this algorithm is that a certain amount of diversity is preserved. The practical implementation will show whether this measure is sufficient.

A solution for this problem is to randomize a part of the population while keeping the best possibilities when an optimum is detected. However, it is not possible to check whether it is a local optimum or a global optimum, so the user must choose the maximum computation time or iteration amount. Otherwise, this process will never end. This problem will be handled in the next chapter.

6.4.2. Population Trace Example

To give an impression of how the algorithm can operate on a population a trace is displayed below. Each box is a candidate solution.

The numbers inside each candidate have the following meaning:

<Identifier>: <Fitness Value>
(<Inherited From List>)

Each row is a population on a certain time. The arrows indicate the impact they have on their offspring.

- The green arrow indicates a copy of the original into the new population
- The blue arrow indicates a mayor influence in the child
- The black arrow indicates a minor influence in the child
- The red arrow indicates no influence in the child

Note that all candidates, which are selected to be maintained, can be mutated. However, mutation is visible in this figure.

The combinations of incoming arrows have the following meaning:

- Green – The green candidate is copied into the new population
- Green, Red – The green candidate is copied into the new population
- Blue, Black – The new candidate has mayor influence of the blue candidate and minor influence of the black candidate
- Blue, Red – The new candidate is a mutated version of the blue candidate

The identifier expresses the name of his predecessor plus a letter making it unique. This way you can see from the identifier who his most relevant parent is. The inherited from list indicates which parents had minor influence in the candidate.

The Fitness value is the violation rate subtracted from zero.

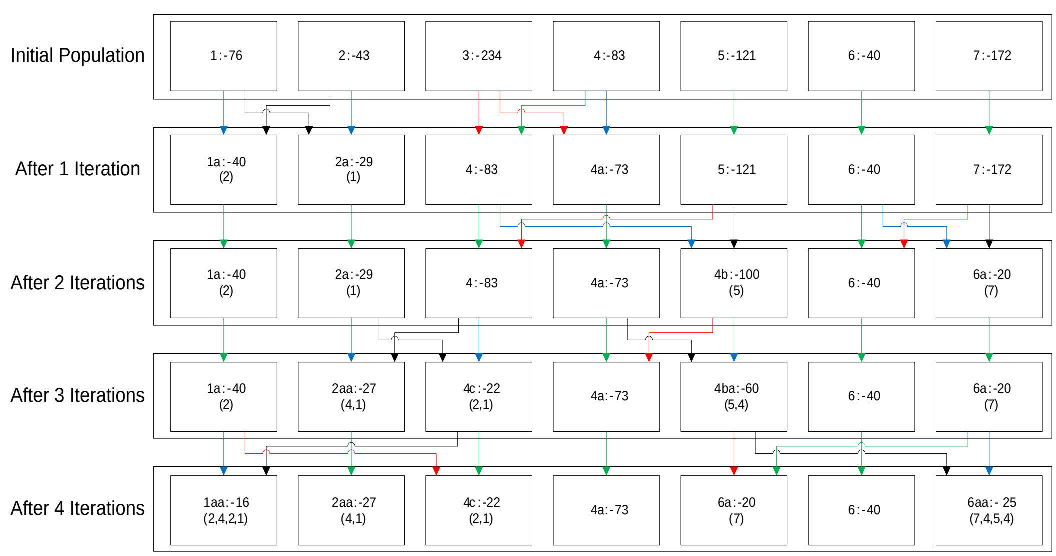

Figure 12 – Example of a Population Trace through evolution

7 Termination Criteria

Every Genetic Algorithm has the property that it is a continuous process. The process will end naturally when the global optimum has been found. In the timetable case, this optimum is a timetable that has the lowest violation rate of all timetables. The ideal case would be that no restrictions are violated. However, this is not always feasible due to too high expectations with too few resources. When the amount of restrictions increases, the chance of a violation free timetable decreases. Therefore, it is not feasible to set only the restraint to halt on obtaining a violation free timetable.

There are three options to end the process artificially: time constraints, iteration constraints and the relative improvement stagnation constraints.

The user can decide which constraints to use, however it is not advisable to use only the stagnation constraint and enable the local optimum fix.

7.1. Time Contraint

The user can set the maximum amount of time the algorithm is allowed to calculate before it needs to give a timetable. The advantage of this method is that the user will know how long it is going to take before he will get his timetable. A minor disadvantage is that it is not determined what to do when the algorithm is halfway an iteration. Another minor disadvantage is that when the algorithm gets little computation power during the execution time the resulting timetable will be less fit than a timetable being generated with much computation power during generation.

The disadvantage is that the algorithm could be on a point on which the fitness improvement per iteration is very high. So the algorithm would be forced to stop functioning while much progress can be retrieved by adding a small amount of time. The problem is however that the algorithm cannot be sure to have a high fitness improvement the next few steps.

7.2. Iteration Constraint

The user can set the maximum amount of evolutionary iterations to be calculated before the algorithm needs to give a timetable. The advantage over time constraints is that it is a bit easier to make the algorithm stop after a certain amount of iterations than after a certain amount of time. For instance what should the algorithm do when it is halfway an iteration when the time elapses?

Another advantage is that it is guaranteed that the algorithm gets a certain amount of computation time.

7.3. Relative Improvement Stagnation Constraint

It is desirable to give a result when an optimum is reached. When the algorithm is not improving the average solution any more, the algorithm can stop. Since the chance of gaining better alternatives is low. The advantage is that the algorithm will stop on a time relative fitness improvement is low. The disadvantage is that it is not known in advance, what the calculation time will be.

However, it is possible to have run into a local optimum, which can be a very undesirable timetable. In this case, you want to gain variation in the population again without losing the good candidates that are already calculated. A solution is to replace a part of the population with new random candidates.

7.4. Mixed Termination Constraint

Due to the various advantages and disadvantages of the different constraints, it is useful to combine them into a single constraint with less influence of the disadvantages.

Let the algorithm work for the duration given by the user, except if the minimum amount of iterations has not been met (useful for computation on computers on which the computation power available to the algorithm is not guaranteed).

The algorithm has the following routine:

1. Make an initial population of candidate solutions.
2. Start the algorithm to evolve the candidates.
3. Continue until the relative improvement is stagnated.
4. If the time (and iteration minimum) has been met, stop the algorithm and output the best-calculated timetable. The routine stops here.
5. Replace a part of the population with new random candidates while keeping the best candidates.
6. Continue with step 2.

The advantage of this algorithm is that all disadvantages of the separate constraints are eliminated. The disadvantage is that the time constraint is likely to be exceeded with this mixed termination constraint. However, it is not desirable to halt the algorithm when much improvement is expected, so taking more time should be in the interest of the user. It is up to the user to make an estimation how complex the scheduling problem is when determining the expected computation time.

The algorithm is expected to have a reduction of restriction violations during the execution of the algorithm. The expected estimated graph of the minimum, average and maximum restriction violation is displayed in the figure below.

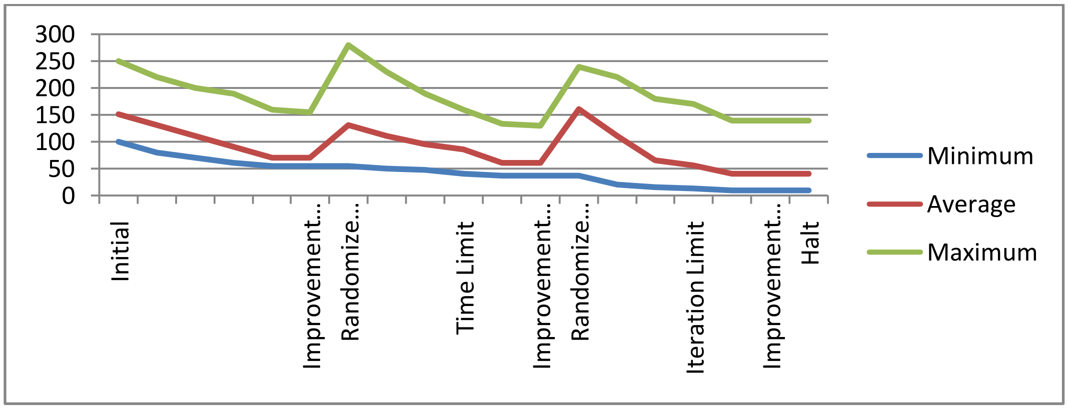

Figure 13 – Fitness Improvement during execution using the Mixed Termination Constraint

The larger the population size, the higher the chance that the randomization process will pull the population entirely out of the local optimum. The reason for this is that the only way to achieve fitness devaluation is with the crossover operator.

The crossover function makes it possible to generate a less fit child who is still better than the other parent is and other child. This is the only way to escape a tight local optimum. The more fitness decreases are necessary to escape the local optimum, the tighter the local optimum is.

The larger the population is the more often an individual from the local optimum can crossover so that a less fit child is maintained. The result is that a larger population can slip out of tighter local optima than smaller populations.

The higher the fitness of the candidates in the local optimum is, the smaller the chance that they will slip out the local optimum. This is a direct result from the 100% chance of mutation in the algorithm. When the fitness is high, a lot of effort is done to mutate with high fitness improvement. The result of this is that the chance of undoing the effect of the previously performed crossover action.

7.4.1. Dynamic interface

Since the algorithm has at any point in the evolution process a best-known solution, it is possible to stop the algorithm and output the current best timetable. This way the user can even put no termination criteria in the algorithm and stop at the time he desires his timetable. This way the user is free to run the algorithm according to his termination criteria.

Part II: Practical Evaluation

In this part, the Memetic Algorithm designed in the previous part is put to the test by a practical evaluation.

First, an introduction to the concrete problem at Info Support is given in chapter 8. Then a concrete solution to the scheduling problem is designed in chapter 9, followed by an evaluation of the implementation in chapter 10.

In chapter 11, the general algorithm is evaluated on the overall performance of the algorithm on scheduling problems.

8 Introduction Concrete Problem

Info Support is a medium sized IT company with its own Knowledge Center. The position of the Knowledge Center within the organization is visualized in the figure below.

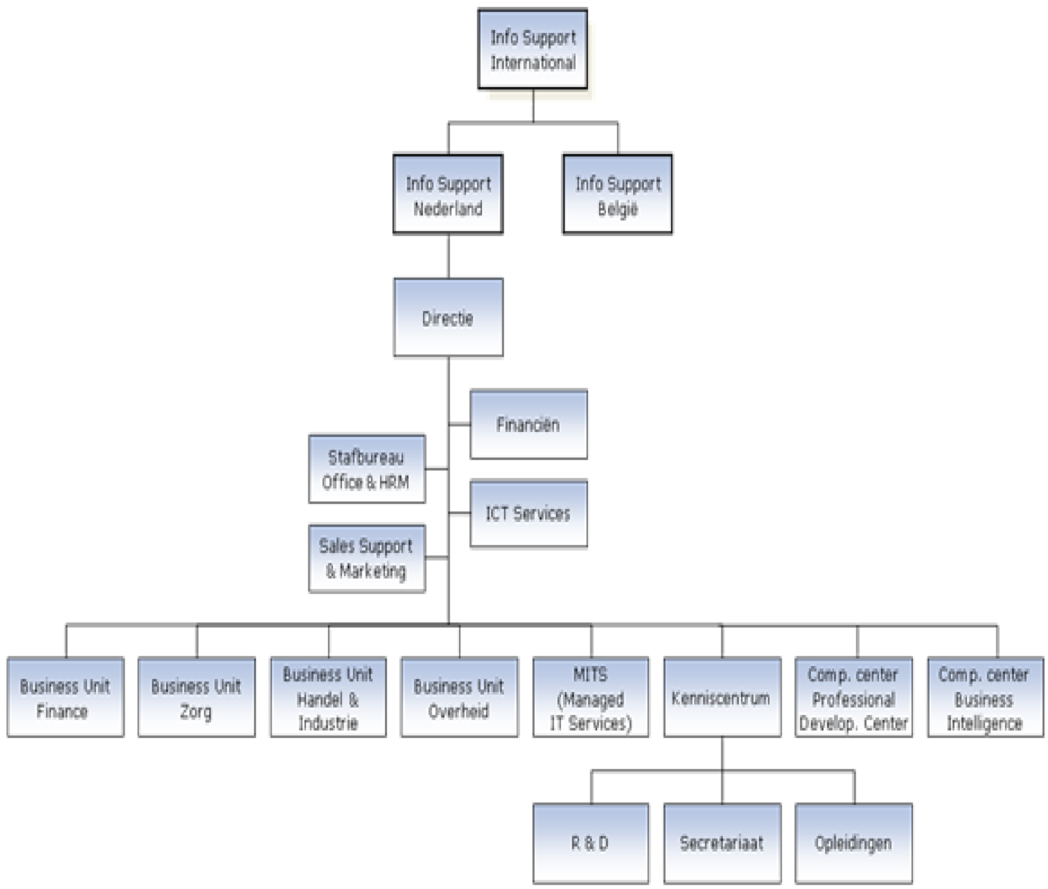

Figure 14 – Overview of Info Supports hierarchy

This center provides training for IT professionals from other companies and for their own consultants to enable them to keep their technological knowhow up to date. Now the timetables for the courses are scheduled by hand. However, Info Support would like to see this process automated with a timetable generator.

8.1. Timetable Usage

A customer can register for training on a specific place and date. The available dates are present in a training calendar, which gives an overview for all training courses. An example is presented in the table below.

	LEV. CODE	IS. CODE	AANTAL DAGEN	TARIEF	STARTDATUM & LOCATIE: UTRECHT (NL)=U VEENENDAAL (NL)=V					
					JULI	AUG.	SEPT.	OKT.	NOV.	DEC.
ENDEAVOUR ONTWIKKELSTRAAT										
Endeavour Essentials		ENEST	1 Dag	€ 550,-	11 V	21 V			20 U	
Endeavour Requirements		ENREQ	1 Dag	€ 550,-	25 V			2 V, 16 U		18 V
Endeavour Analysis		ENLYS	1 Dag	€ 550,-	15, 29 V				4 V	
Endeavour Java Development		ENDEVJ	3 Dagen	€ 1650,-				6, 9 V		15, 18 V
Endeavour .NET Development		ENDEVN	4 Dagen	€ 2200,-		26 V		20 V		8 V
Endeavour Test		ENTEST	1 Dag	€ 550,-				1 V		4 V
Endeavour .NET Integrator		ENINTN	1 Dag	€ 550,-			Bel voor trainingdata: 0318 50 11 19			
Endeavour Project Management		ENPM	2 Dagen	€ 1100,-			Bel voor trainingdata: 0318 50 11 19			

Figure 15 – Example of a partial training calendar

For the non-Dutch readers a short translation of the columns present in the table above. From left to right the names are training name, training code, number of days, cost and starting date. After the date pops up a 'V' or a 'U', this represents the location (Veenendaal/Utrecht)

The courses are scheduled, and then the customers can sign in the desired scheduled courses. An employee of the knowledge center does the scheduling by hand. This is a time consuming task, which can be done by a computer. What you see in the figure is only a part of the planning. There is also a schedule for teachers. Therefore, a timetable does not only contain trainings and dates, but also a teacher per training.

8.2. Concrete Assignment

Info Support would like the creation of the entire timetable to be automated, because it takes a lot of time making each schedule. Besides that, it can very well occur that a computer finds better schedules than the manual scheduler makes.

Because the courses are scheduled before they know the customer's demand, it is valuable to know how to determine which courses should be scheduled. This problem is not going to be solved by an algorithm; this is a management problem concerning market prediction. This problem is going to be address in the next part of this thesis.

9 Concrete Design

9.1. Functional Requirements

The main functionality of the program is that it gives a suitable timetable based on requirements from the user.

A timetable consists of courses, teachers, locations and times. The attributes of these parts are a subset of the general set described in Part I. This subset is the relevant part for Info Support.

Another relevant functional requirement is the weight of the restrictions.

9.1.1. Courses

For each course the course identifier (IS. Code), the duration, the prerequisite courses and the times to be scheduled are relevant attributes.

- The identifier is used in other input requirements provided by the user.
- The duration of the course is relevant in the scheduling because a location (and teacher) is busy for the duration of the course. A course must be planned within a workweek.
- The prerequisite courses are a collection of identifiers of courses that are to be scheduled before the current course.
- For the timetable period, each course has an amount of times to be scheduled. The courses should be planned as widely spread in time as possible.
- The desired location must be set. If there is no preference for location, the course must be evenly divided between locations.

Other course information is not relevant for the generation of the timetable. This information will be entirely excluded from the algorithm.

9.1.2. Teachers

A teacher consists of an identifier, contract hours, deployment rate, course competences, days off, location preference, tolerance period for teaching activity and variation between courses desire.

- The identifier is not used in other attributes; however, it is needed for identification in the timetable representation.
- Contract hours are the amount of hours a teacher is available at work.
- The deployment rate is the percentage of the contract hours the teacher ought to be scheduled for teaching.

- The course competences are represented as a list of courses the teacher can give. Each course has numerical indication of how experienced the teacher is with the course. In addition, an override check can be made to ensure the teacher is scheduled for a certain course. In this case, a specified period (in days) before the actual course is to be kept free from teaching.
- The days off are registered with the teacher. Each start and end of days off is inserted in a list.
- Preference for location. A teacher can give a preference for Veenendaal or Utrecht.
- Tolerance period for teaching activity. A teacher can give the size of the timeframe in which he wants his hours to be evenly spread out. A small timeframe ensures not to be scheduled for days in a row, while a large timeframe allows it. The smaller the timeframe is, the harder the scheduling process is.
- Variation between courses desire. This is a numerical value giving the divergence of courses in the scheduling period. The higher the number is, the larger the penalty of scheduling few sorts of courses for the teacher will be. Note that a teacher who can give only one course will more easily violate this rule, so by giving a large value it will by definition give a low fitness value to the timetable.

Other teacher information is not relevant for the generation of the timetable. The only attribute which can be useful when finalizing the timetable is the entire teacher name. This name is excluded from the algorithm.

9.1.3. Locations

Info Support has two locations where courses can be given (Veenendaal and Utrecht). On these locations, multiple rooms are available for giving courses. The determination in which room the courses are given is postponed until the actual course registrations are known. Based on these requirements the following attributes are determined:

- Location ID (V for Veenendaal, U for Utrecht)
- Room capacity on location – The overscheduling of a location on a specific time may not exceed certain limits.

9.1.4. Time

The courses are scheduled with an amount of days; the size of a time block is 1 day. The timetable period is set on a quart year. A day is represented in DD/MM/YYYY. The algorithm gets the start date and end date as parameters.

9.1.5. Weight of restrictions

The previously mentioned restrictions must be weighted in order to control the output of the algorithm better. Some requirements are less important than other requirements to be achieved in the final timetable.

It is also necessary to be able to give a violation level based penalty for a requirement. For example, a certain teacher should be scheduled for three days a week. Whether the teacher is scheduled for four days or for zero days, it is both a violation of the restriction. However, the four days scheduled can be better than zero days.

It must be possible to give hard upper and lower bounds and soft upper and soft lower bounds for a restriction. With these bounds, it is possible to determine the violation level of restrictions gradually.

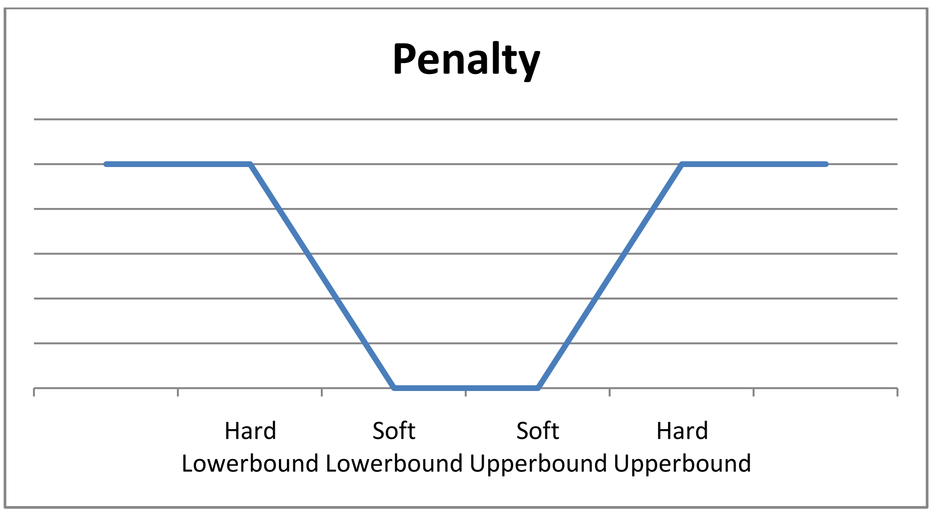

Figure 16 – Visualization of restriction penalties by bounds

9.2. Technical Requirements

The main reason for this research is to make a scheduling program using Memetic Algorithms in order to validate how a MA performs on generating timetables; the incorporation of MA in the program is the main requirement.

9.3. Functional Design

The program will consist of all required input files and the executing program. When the calculation is done, a timetable is saved to an xml file. Besides the timetable, a quality report is also saved.

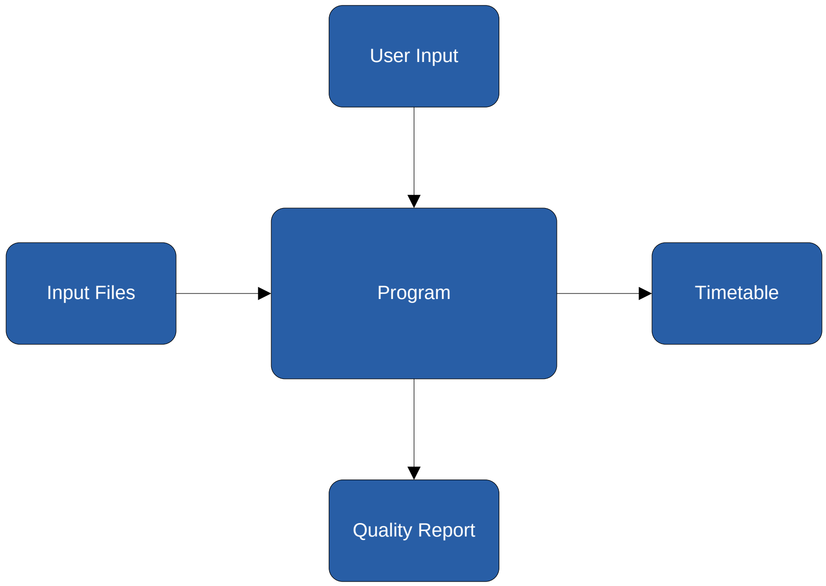

Figure 17 – Black box representation of the program

9.3.1. Input files

The input files will consist of objects in xml version 1.0 with the utf-8 encoding. This is the standard representation used by the XML-Serializer available in C# .NET framework 2.0 ~ 3.5. These files contain all the information needed to generate a timetable. The following information is available in separate input files:

- Course information
- Teacher information
- Location information
- Time information
- Weight of Restrictions
- Algorithm and interface settings
- Previous timetable (optional)

A basic editor for the input files will be provided with the program. The main purpose of this editor is to avoid the manual generation of the xml input. Making the input files is not the goal of this assignment, but they are needed in order to test the functionality of the program.

The program does not check on impossible timetables. For instance if all teachers are not available for any courses the algorithm will create a timetable with a low fitness value. The check on impossible timetables is a welcome addition to the program and therefore is mentioned in the future research chapter.

9.3.2. Interface

The program is going to be a windows Forms application. Some basic settings will be adjustable inside the program, but the idea is that all settings are retrieved from the input files. Changed settings can be saved back to the input files. The only settings that can be adjusted inside the program are:

- Termination criteria (minimum time, minimum iterations, stagnation boundary)
- Probabilities (crossover chance, tournament size % to be used)
- Population size
- Percentage of population to be used for crossover
- Choice to use a previous timetable for more nuanced scheduling. File is selectable.

The program gives an indication during execution what the current worst fitness, average fitness and best fitness is. In addition, time and iteration is displayed. This can be disabled for performance increase. It is possible to stop the algorithm at any time during the execution; the best-known solution can be exported to an xml file.

9.3.3. Quality Report

When a timetable is generated, a quality report is generated containing all violations of the generated timetable. This document is a tree representation with different levels of violation. The following hierarchy is represented in the quality report.

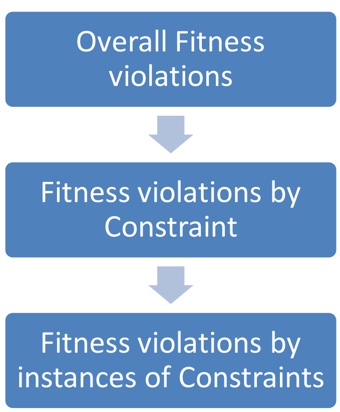

Figure 18 – Hierarchy of the tree represented in the Quality Report

This report is useful for the user to verify whether the violations are within an acceptable level. This report gives a review to the user whether the weights of the violations are in proportion. It also gives hints about the impossibility of the combination of teacher (availability) and courses.

9.3.4. Multi Threading

The program uses multi threading on a limited scale because the purpose of the algorithm is making timetables and not making a perfect multithreaded program. The purpose of multithreading is to make the algorithm able to use more of the computers computation power. The limitation of computation power for a regular single threaded algorithm comes from the fact that most computers are multi cored nowadays. A single thread program can use only one core, so that limits the maximum computation power available to the program. The amount of threads is limited to the amount of crossover operations each iteration plus the main thread and interface thread.

The only record that can be adjusted by multiple threads at the same time is the variable containing the worst seen timetable by the algorithm. This rate of independence between threads follows directly from the fact that all crossover operators operate on different candidate solutions. This simplifies the implementation of multithreading significant.

In the figure below, the part of the algorithm that is getting its own thread is marked by the orange circle. Since multiple crossover operations are performed each iteration they can all be executed at the same time.

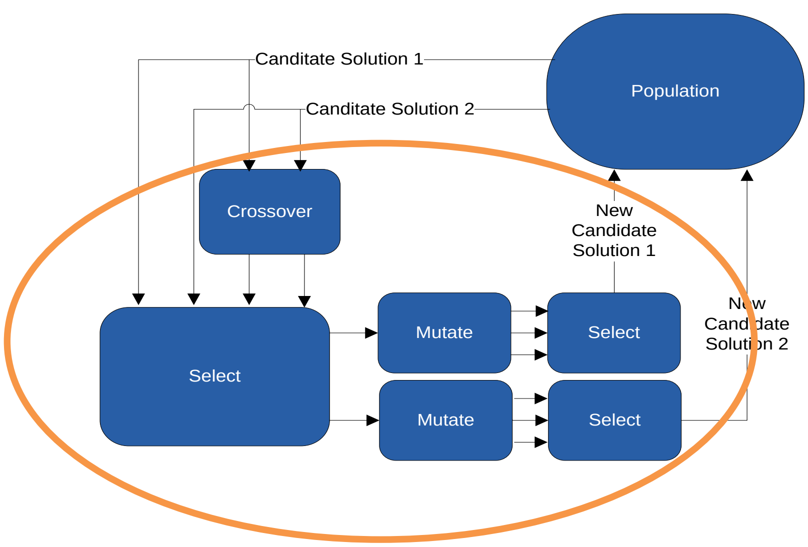

Figure 19 – The Memetic Algorithm with the thread part highlighted

10 Result Evaluation

Before the resulting algorithm can be tested, a test plan has to be designed. This plan is needed for testing the algorithm on performance versus time. With this plan, specific tests can be carried out to measure the quality of the algorithm and the difference between different settings.

10.1. Test Plan

The main objective of the algorithm is delivering a sufficiently good timetable within certain time limits. The quality of the timetable is generally measured versus the best attainable timetable within the given boundaries. This is a real concern, since one can only determine what the best timetable is when all possible timetables have been generated. The limited duration of this experiment does not allow calculating the best timetable brute force. That is why the decision is made to let the devised algorithm for a significant longer time than it regularly has when it is in use.

10.1.1. Testing Quality

As mentioned it is necessary to know in what fitness range the best timetable is located. Since the goal for Info Support is attaining a sufficiently good timetable, it is not necessary to have the perfect timetable. However, it is needed that certain quality is attained. The desired quality has a relation with the execution time of the algorithm. Since the algorithm needs exponentially more time to improve the quality, there is a boundary of time versus quality improvement. This boundary expresses the moment that makes the user want to terminate the algorithm. The following test will be used to measure quality improvement over a time way longer than regular use:

Time limit:	None
Iteration limit:	None
Stagnation boundary:	With and without
Crossover chance:	10% per factor
Crossover amount	10 candidates
Population size	1000 candidates

Note that these are two tests. The first test tries to avoid local optima. The second test does not avoid local optima by randomizing the population.

10.1.2. Testing Crossover chances

A standard issue with Genetic Algorithms is the uncertainty what chances are best in the algorithm. The crossover operator is going to be tested with 10%, 20%, 30%

and 40% crossover chance. Note that this chance is the chance for each factor to be chosen from either the first parent or the second parent. That is why chances above 50% have the same effect as chances below 50% and do not need testing. Again, these tests are done with and without stagnation boundary.

Time limit:	None
Iteration limit:	None
Stagnation boundary:	With and without
Crossover chance:	10%, 20%, 30% and 40% per factor
Crossover amount	10 candidates
Population size	1000 candidates

These eight tests are compared to each other.

10.1.3. Population Size and Crossover Amount

Another relevant part in the algorithm is the population size and the amount of candidates to be adjusted at a single time. The population size has a relation with the capacity of avoiding local optima and the speed of improvement of the population. The smaller the population is, the faster the algorithm will be. The larger the population is, the smaller the chance of having a population with small divergence will be. The crossover amount is a setting with minor impact. The smaller the amount of candidates to be used by crossover is, the larger the overhead of shuffling the population. However, this shuffling does not cost much computation power. The advantage of not evolving the entire population at once is that the possibility arises that candidates created at multiple iteration steps can evolve together. Note that the duration of an iteration round is strongly dependent of the amount of candidates to be used for crossover and the improvement per iteration is limited by this amount because a part of the population stays the same.

The different tests to be executed are deliberately significantly different in amount. This is chosen because there can be difference between one timetable and another, so it is not relevant to determine the best population size exactly for this test set.

Time limit:	None
Iteration limit:	320000, 160000, 80000, 40000, 20000, 10000, 5000, 2500 and 1250 iterations
Stagnation boundary:	With
Crossover chance:	50% per factor
Crossover amount	100% of the population
Population size	2, 4, 8, 16, 32, 64, 128, 256 and 512 candidates

10.2. Test Results

All developed tests have been executed. The results are evaluated in this paragraph.

10.2.1. Quality Test Results

The quality test showed a nice curve over time indicating the expected improvement degeneration over time. As expected, the version without stagnation prevention showed a more rapid improvement in early generation. This is explained by the constant randomizing of the population, which limits the evolution by the replaced candidates. However, it is possible that this shuffling brings new candidates with more potential. Therefore, in the end it is useful to use some randomizing of the population. However, this has to be used with regulation in order to keep the algorithm fast.

In the figure below, the concrete test results are recorded. The red line is so thick because the amount of iterations is very large and the population stagnated so many times that the separate improvement curves are merged into one big line.

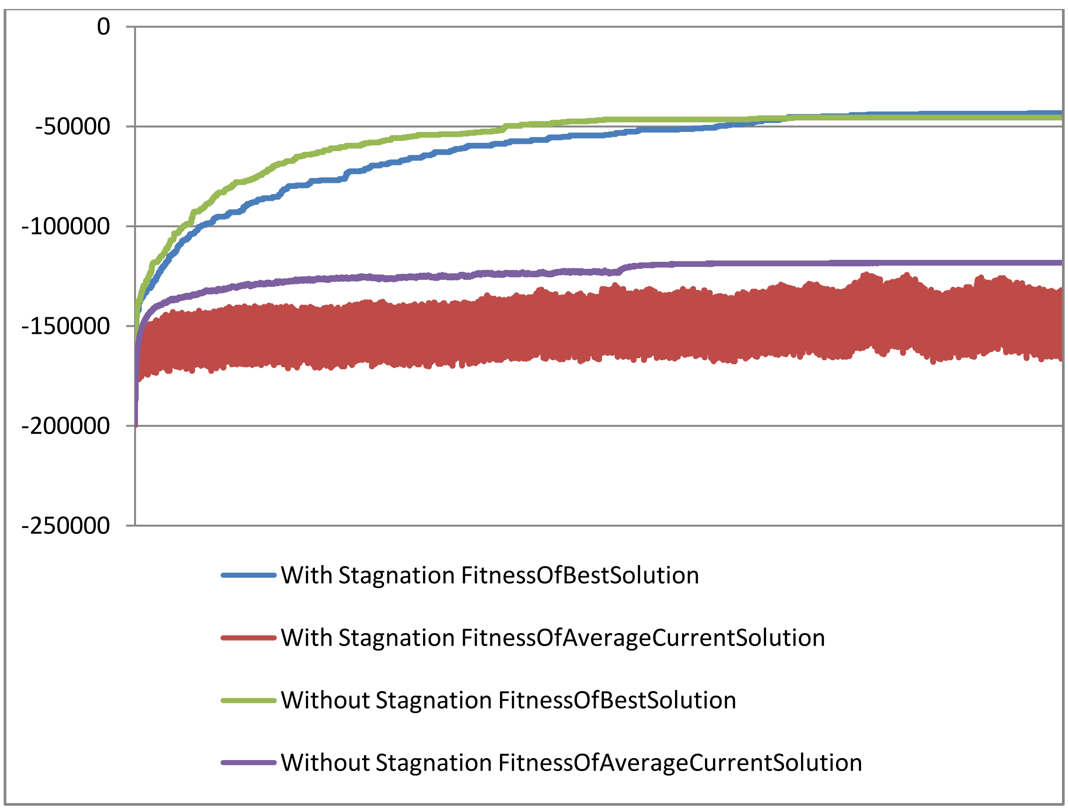

Figure 20 – Test results quality over time

The concrete test values have been:

Time limit:	Not relevant due to irregular load balancing on test machines
Iteration limit:	32000
Stagnation boundary:	With and without
Crossover chance:	10% per factor
Crossover amount	10 candidates
Population size	1000 candidates

The concrete test results have been:

Best candidate without stagnation prevention:	-45416
Best candidate with stagnation prevention:	-43271
Equal performance iteration step:	22432
Quality at that point:	-45876

These results indicate that there is such a thing as a local optimum for the algorithm. Otherwise, the best candidate generated without stagnation prevention cannot be worse than the best candidate generated with stagnation prevention can. Note that the best candidate found with stagnation prevention perhaps is not the optimal candidate because the line is not flat for a long time compared to the time the previous best candidate has been found.

10.2.2. Crossover Chance Test Results

Although the main reason for this test was to test what crossover chance performed best it is relevant to note that the conclusion formed on the previous test about the stagnation still holds with this test. The behavior is much alike the previous test. Note that the tests with stagnation prevention have not recorded in the figure below. This information made the figure crowded, which did not contribute to the comprehension of the figure.

The main conclusion to be drawn from the figure is that the crossover chance does not influence the progression that much. The differences between crossover chances are so small that the deviation of the algorithm can be the cause of the differences between the best candidates.

When you look at the average fitness values, you can see that the lower crossover chances have an average higher fitness value. A higher average fitness value appears to be better. However, the best-found candidate indicates minor difference between the crossover rates. The cause of having higher average fitness values with lower crossover chances is explainable by the near duplication of the best parent in crossover. Because of this, the better candidates will have nearly the same offspring while other the other candidates disappear from the population. This near duplication of better candidates brings a risk with them to steer the population to a local optimum. When the individuals stay more differentiated this risk is smaller.

In the figure below, the concrete test results are recorded:

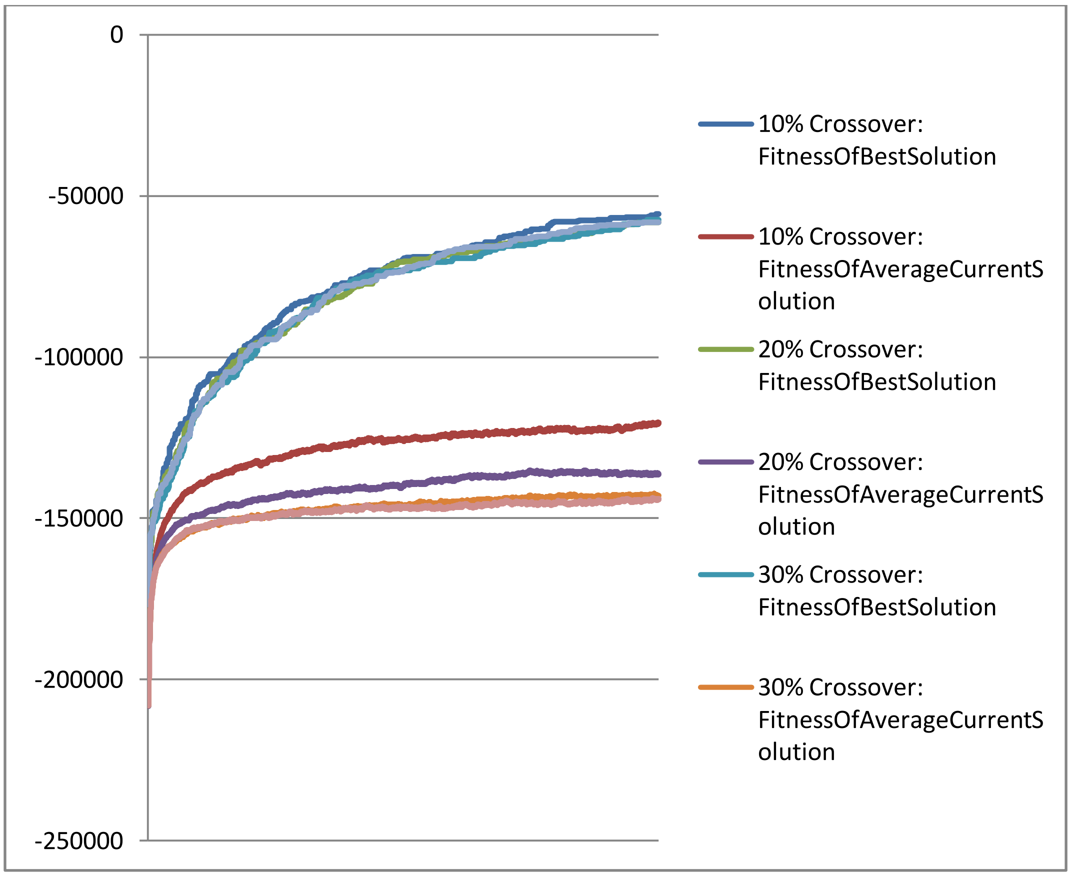

Figure 21 – Test results crossover chances

The concrete test values have been:

Time limit:	Not relevant due to irregular load balancing on test machines
Iteration limit:	<>
Stagnation boundary:	With and without
Crossover chance:	10%, 20%, 30% and 40% per factor
Crossover amount	10 candidates
Population size	1000 candidates

10.2.3. Population Size Test Results

The population test results showed a remarkable preference for small populations. As you can see in the figure, the smaller populations perform better than the larger populations. In the first figure, the tests are displayed with results versus computation time. They all have executed 320.000 crossover operations. Here the relation between performance and population size is clearly visible. The smaller the population size, the better the performance is. Note that a very small population has a higher susceptibility for stagnation; the population of four found a better solution than the population of two. This result supports the stated relation between the population chance and the susceptibility for running into local optima.

The following results have been attained by the test:

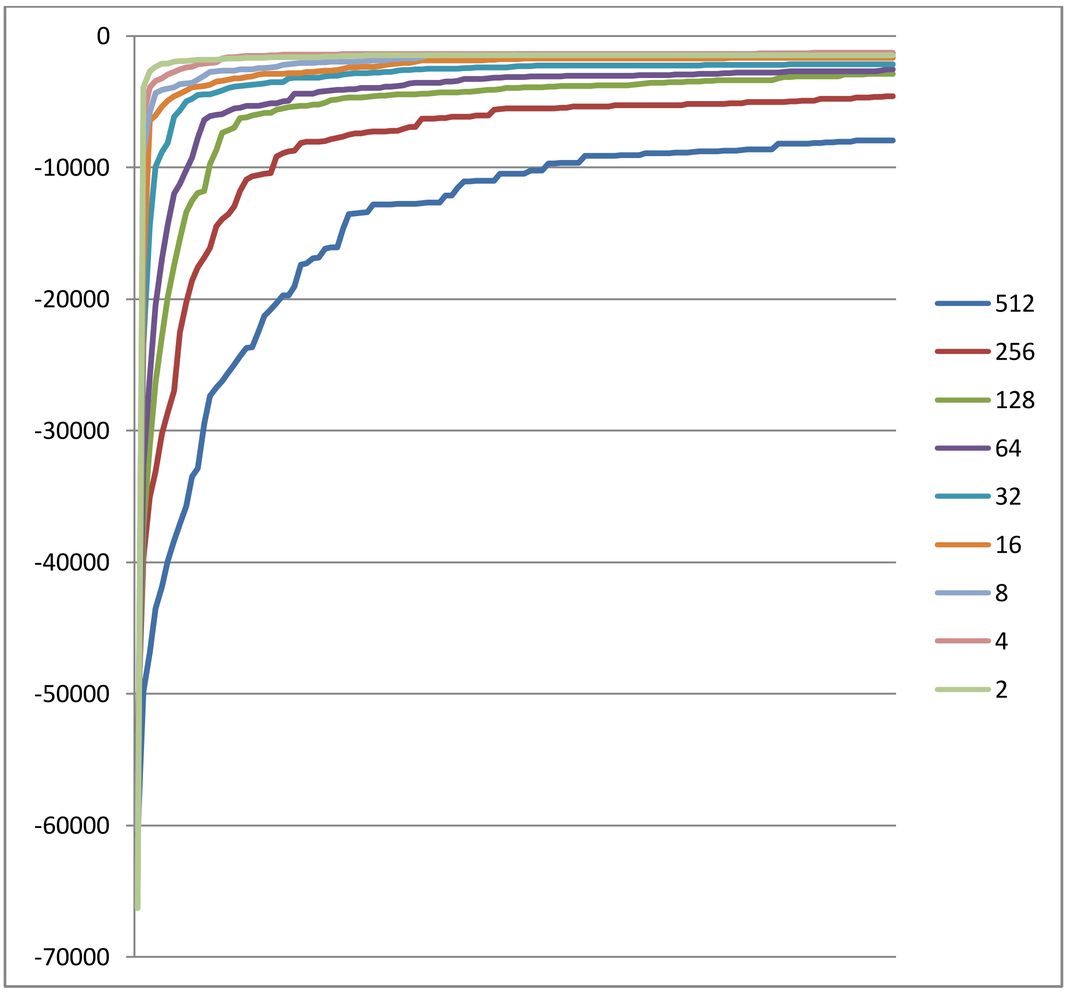

Figure 22 – Test results population size with crossover limitation

The concrete test values have been:

Time limit:	Total 320.000 Crossover operations
Iteration limit:	320.000, 160.000, 80.000, 40.000, 20.000, 10.000, 5000, 2500 and 1250 iterations
Stagnation boundary:	With and without
Crossover chance:	50% per factor
Crossover amount	100% of the population
Population size	2, 4, 8, 16, 32, 64, 128, 256 and 512 candidates

With the same test results, it is also possible to determine the performance per iteration for each population size. This gives insight in the difference between the population sizes. It is quite remarkable that larger populations with much more diversity do not perform better than smaller populations. Note that the larger populations use much more resources than the smaller populations to achieve the same results.

The following results have been attained by the test:

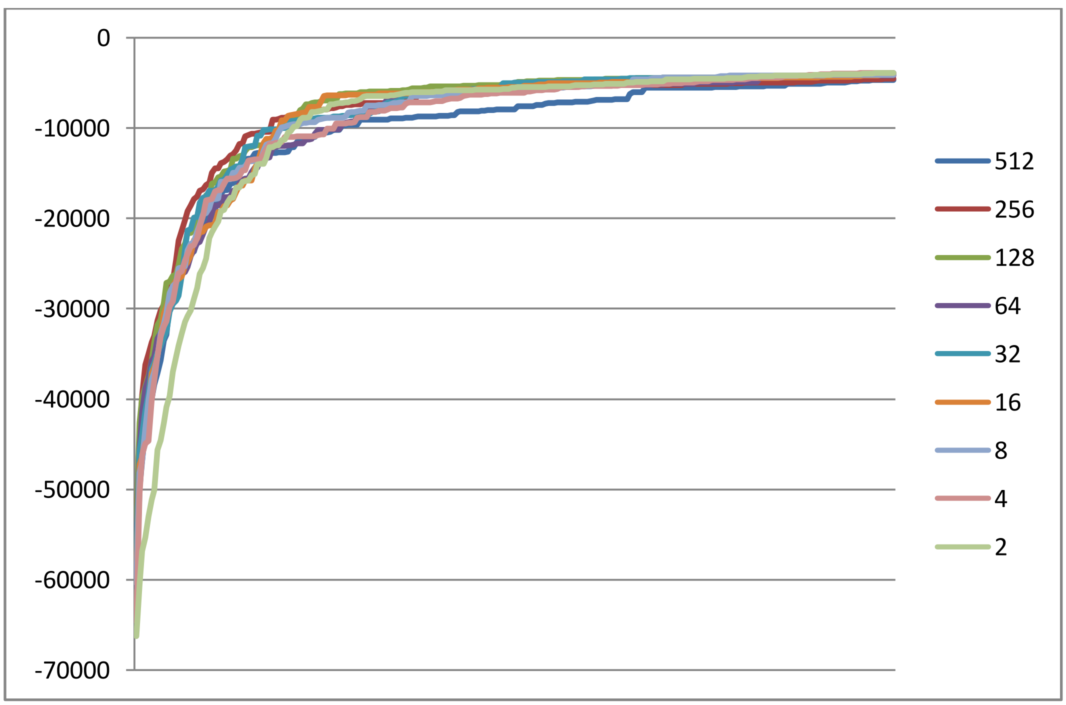

Figure 23 – Test results population size with equal amount of iterations

The concrete test values have been:

Time limit:	Not relevant due to irregular load balancing on test machines
Iteration limit:	2500 iterations
Stagnation boundary:	With and without
Crossover chance:	50% per factor
Crossover amount	100% of the population
Population size	2, 4, 8, 16, 32, 64, 128, 256 and 512 candidates

10.3. Test Conclusions

The algorithm on itself is susceptible to running into local optima. However, it is easy to eliminate this problem by using randomization when the population quality stagnates. The chosen strategy is to randomize 2/3 of the population when the average fitness value of the population is worse than 100 iterations ago. The best candidate solution is always maintained. The reason not to randomize the entire population (minus the best solution) is to keep a random set of candidates to keep some of the effort of the previous iterations. The advantage is more performance; the potential disadvantage is that the odds are higher that the algorithm does not escape the local optimum.

The crossover chance showed a remarkable small impact on the algorithm, except that the average population has a much higher quality with a lower crossover chance. My advice is to use a higher crossover chance to improve the odds of having a diverse population. You will lose a limited amount of performance but with more certainty of not running into local optima.

The population size tests showed the most remarkable results. The population should be very small in order to keep the algorithm fast. The chance of running into local optima appears to be very slim. A population of two is not advisable because of the increased chance of staying in local optima. It is reasonable to say that other small population sizes suffer from the same problem. The test machine is a dual core processor that performs best when running six crossover threads at the same time. This makes the choice for a population of 12 candidates a practical choice, since this is the smallest population utilizing six crossover threads at the same time. Note that it is not possible to compute more threads at the same time than half of the population. All results of the previous iteration have to be finished and the population has to be shuffled before the next iteration can start.

11 Overall Performance & Conclusions

There are multiple ways to generate a timetable automatically. The most trivial method of scheduling is creating all timetables possible based on the available dates, courses and teachers and evaluate them all. After that, pick the timetable that suits the requirements of the user best. This method has the advantage that it is certain that when the algorithm finishes the best solution is found. No other algorithm can give this certainty with a problem of this rate of complexity.

The downside of simply trying all alternatives is that the computation cost is astronomically large. The test set used with the tests of the developed algorithm made 9.5×10^{310} timetables possible to generate. With a standard computer (Pentium 4 dual core 3.0 MHz), it is possible to evaluate about 30 timetables each second. This is equal to 9.4×10^{8} timetables a year. This is a clear indication of the sheer impossibility to evaluate all timetables. The developed algorithm finds a decent timetable in 24 hours. After that period, only minor optimizations are found.

There are more solutions made to cope with the complexity of timetable generation. Some examples can be found in [Alkan, 2003], [Monfroglio, 1996] and [Őzcan, 2005]. However, the other proposed algorithms all work on slightly different timetable problems. This difference makes it impossible to use test sets used by other genetic algorithm designers and compare them directly.

Part III: Market Determination

In the previous part, the algorithm for timetable generation has been evaluated. This algorithm depends on concrete demands for courses in a specific period. Therefore, the need for courses has to be determined prior to the timetable generation. This determination is a market determination problem. In general, these kinds of problems are about product development. In the case of the Knowledge Center, the product is the course.

The courses to be scheduled are determined by the Sales Manager Learning Solutions. Since the timetable is made before the actual demand is known, it is necessary to have an indication what the actual demand will be like. The currently used methods to determine this demand are defined in chapter 12. In this chapter, the view of the Sales Manager Learning Solutions is elaborated. The currently used method consists of different input factors with different effectiveness.

The factors that are identified are evaluated in chapter 13. In this chapter, the views of other field experts are elaborated. Since not all relevant factors are present in the current determination process, there is room to define these as well. These factors are described in chapter 14. Here the views of other field experts are also elaborated, now with an eye on missing factors. The overall advice to Info Support can be found in chapter 15.

12 Defining current determination process

In this chapter, the determination of the demand for courses is elaborated. This information is obtained from the Sales Manager Learning Solutions. The Sales Manager Learning Solutions is responsible for the market determination of the courses on offer. In a later chapter, this vision is compared with perceptions of other people at Info Support.

The customer demand for the courses at Info Support is currently determined in three main areas: Technology Suppliers, Internal Demand and External Demand. In the pie chart below, these sources are displayed. The three defined sources are elaborated in the following paragraphs.

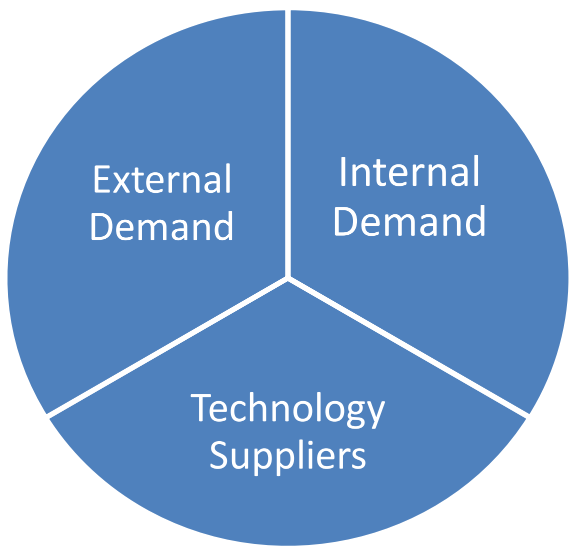

Figure 24 – Pie chart concerning the main sources for the determination of the courses on offer

12.1. Technology Suppliers

The technology suppliers are the companies who make the volume software used by developers to make software products. The consultants at Info Support need to keep their knowhow of these newest products and technologies up to date.

Info Support works tightly with volume software vendors to develop courses for new products. The primary source of new volume software for Info Support is Microsoft. Microsoft has a learning division, which has the task to develop course materials and exams. This material is used by Info Support to give Microsoft certificated courses. However, this division does not make courses for all Microsoft products. This leaves an opportunity for Info Support to take this role. The main reason Microsoft is not making courses for all software is because of the limited communication between the learning division and the product development divisions and the lack of resources of the learning division. Another reason is that they are a profit/loss center, which makes them focus on high volume products and not on niche products.

Since the learning division and the product development divisions of Microsoft are separated from each other, the product development divisions do not really care who provides courses for their programs. The reason for this is that they are rewarded for making successful products and not for letting the learning division sell enough courses. They want their products to be used. However, most products they sell need training before they can be used. Therefore, products can be sold better when there is a possibility to obtain information needed to be able to work with the products.

Because of this information need, the product development division desires courses for their products. Their main concern is not helping the learning division, but improving their own results. Therefore, if a partner wants to make a course for their program, they will help with the creation of the course content. They will even buy some lectures from the developed course because they want their staff to be up to date too. This helps the partner with some certainty of regaining their investment. Info Support takes this role as partner in making courses for new products.

There are two main advantages of working together with a volume software vendor like Microsoft. The first advantage is that Microsoft is capable of generating demand. Customers from the Knowledge Center ask for specific Microsoft Courses on a regular basis. No other software vendors who can match this market penetration and fame. The second advantage is that they have a large marketing department. This department is able to make market predictions that are hard to make for a small marketing department like the one from Info Support. One way of obtaining this market information is by attending conferences and meetings organized by Microsoft.

By attending conferences, not only market information is obtained from Microsoft, but also from competing firms. The advantage of this is that new technologies are spotted early on. This makes it easier to adjust the strategy compared to later stages of the technology cycle. The downside of the collaboration between competing firms is that competing firms get to know your information too. It is important to keep this in mind.

Microsoft defines numbers for specific courses that cover certain knowledge. The advantage of these numbers is that customers can see that a course is covering certain knowledge. A problem with these numbers is that competing firms give other courses with the same course number causing that the customer sees the courses as equal. The courses from Info Support tend to be of a higher quality but also with a higher price than the competition. It is hard to explain to the customers who compare courses on price, location and date that these courses are implemented differently causing a higher quality. However, this is not particularly relevant for the market determination on itself.

12.2. Internal Demand

The core business of Info Support is professional tailored software development. Therefore, the most important customer of the Knowledge Center is Info Support itself. The internal demand for courses is determined via three channels: course evaluation forms, via teacher interaction and via hierarchical feedback.

12.2.1. Course Evaluation Forms

After the courses are completed, the students are given the opportunity to give feedback about the course they followed. In the appendices, the course evaluation form is available. This is an analogue version of the course evaluation form, while the course evaluation form given to the students is a digital version. However, the questions are the same.

This form asks questions about the quality of the course and it gives the participant the opportunity to tell which information he would like to obtain via a course, which was not part of the course he followed. This second part is relevant for the market determination.

12.2.2. Teacher Interaction

During a course, the participants can ask questions and give remarks to the teacher that can be relevant for determining the market for courses. Each teacher is part of a team of teachers who operate in the same area of expertise. They have periodical meetings in which they decide which courses are desirable to develop or enhance. In these meetings, the comments they hear during their teaching are used in the decision process. With this procedure, the comments given by the students are used to improve the courses.

12.2.3. Hierarchical Feedback

The head of the Knowledge Center is responsible for the final curriculum. To obtain information from within the company, he has meetings with the Business Unit Managers to obtain the demand from the development divisions of Info Support. The Business Unit Managers obtain their information from progress interviews with their software developers.

The developers, who want to develop their software development skills in an area in which no courses are present at Info Support, are a source of internal demand. Another source of information is feedback from software development teams who lack knowledge needed for their projects. Their needs are also communicated with their Business Unit Managers.

The Sales Manager Learning Solutions decides together with the head of the Knowledge Center which new courses should be developed.

12.3. External Demand

The external demand is the desire of consultants from competing firms. These consultants give feedback via the following channels: course evaluation forms, via teacher interaction and via customer relations. Since the course evaluation forms and the teacher interaction are equal to the ones described in the Internal Demand paragraph, they are not elaborated in this paragraph.

12.3.1. Customer Relations

The Sales Manager Learning Solutions keeps contact with the most important customers to obtain feedback about the courses on offer. Other customers are only visited when there is a specific reason for the visit. The reason not to visit all customers is the sheer amount of customers, which is much too large to be physically possible to visit by one person in a regular basis.

It is not reasonable to expect from the customer that he put his own resources in the course demand determination for Info Support when there is no direct benefit for him. It is possible to give the customer benefit for helping determining the demand. For example by giving rewards for useful ideas. However, another possibility is to obtain the information directly from the developers. This is what the Sales Manager Learning Solutions does. He gets his information by communicating with the team leaders working at the customer. He determines what knowledge they are missing in their software development projects.

It is also possible for him to get information from existing customers of Info Support, but who are new to the Knowledge Center. These customers already bought products or services from Info Support. These products can generate a need for courses, which can be fulfilled by the Knowledge Center. The main reason will be to sell more courses. However, new customers can also bring new demands for other courses. It is always relevant to get more demand information from the customers.

12.4. Overlap between internal and external demand

The internal and external demands have in common that they both have course evaluation forms and teacher interaction. This is logical since the developers from both groups go to the same courses and fill out the same course evaluation forms.

The teacher groups evaluate both direct teacher interaction and the course evaluation forms. This makes the teacher groups a significant source of information for the Sales Manager Learning Solutions. In the pie chart below the significance of the teacher groups is visualized.

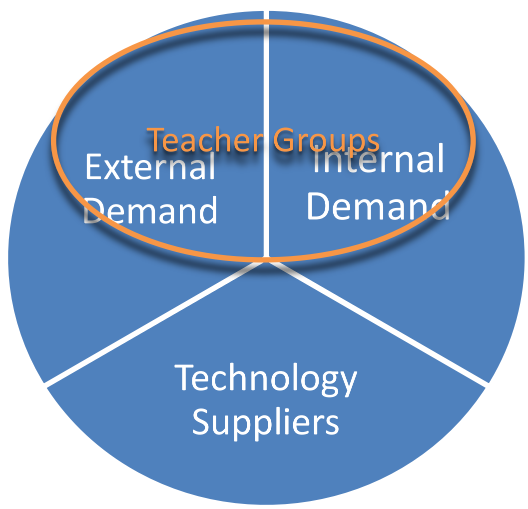

Figure 25 – Pie chart visualizing the impact of teacher groups on the internal and external part

12.5. Market Domain

There are clear boundaries within which markets Info Support wants to give courses. Their aim is to be the first to give courses on new products and on products commonly used by internal developers. Large competitors typically dominate products that exist longer and prove to have a descent market perspective. Info Support does not aim for being active in mass markets. The reason for this is that they lack the volume to make decent profit in those markets.

There is a shift in the market domain to courses in soft skills. Traditionally, this kind of courses is not in the curriculum of Info Support. However, these courses are needed for the improvement of the maturity of the software development process. Since this evolution is recently formally initiated in the company, the desire to have these courses within the company rises.

13 Quality of the current determination process

In this chapter, the current method for market determination is evaluated. The aim is to find weak spots in the used methods and solutions to make the determination process better.

13.1. Internal Demand

The internal demand is determined via course evaluation forms, which in practice provide little feedback for the need of new courses according to the head of the Knowledge Center. Another method being used is the feedback via the business units. However, the disadvantage of this method is that people do not feel encouraged to give feedback with this mechanism.

In the current situation, three different persons handle the responsibility for the entire curriculum from Info Support. The Human Resources department is concerning the soft skills courses, the Professional Development Center is concerning specific Endeavour courses and all other courses are part of the Knowledge Center's curriculum. This responsibility is shifting to a centralized management in which the Knowledge Center covers all areas of knowledge.

13.1.1. Course suggestions and consequences

With the right attitude, it is possible for a consultant from Info Support to come to an improvement of the curriculum. For example, the Microsoft Technology Manager from Info Support saw a knowledge gap in the curriculum, which made him develop his own course. After he developed the course, he initially did the teaching himself. When the course was mature, he taught the teachers how to give the course. Now the course is part of the curriculum of the Knowledge Center. Although this method proved successful for him, you cannot assume that every colleague with a great idea is capable of setting up a course himself.

According to the Java Technology Manager, the need for new knowledge by individuals in general does not lead to new Java courses. The main reasons are the expected lack for a market for the courses and the lack of teachers creating the courses. A Java teacher who also makes courses agrees with him that there is not enough time for him to make all courses he desires. The reason for this is that the courses are being made in the spare time between teachings. The colleagues in need for the knowledge not available internally are trained externally when no courses are going to be made internally.

13.1.2. Competence Centers

The technology managers are chair of their Competence Center. A Competence Center is a group of people who all have a shared competence. Info Support has the following Competence Centers: Architecture, Infrastructural Software Services, Java, Microsoft Application Development, Project Management, Requirements & Analysis, Business Intelligence & Data Warehousing and a new Competence Center Testing. These Competence Centers do not correspond to the Business Units. A Competence center has members of all disciplines: managers, consultants, teachers, etcetera. These people come from all Business Units. The figure below gives a visual representation of how the competence centers are positioned within the company.

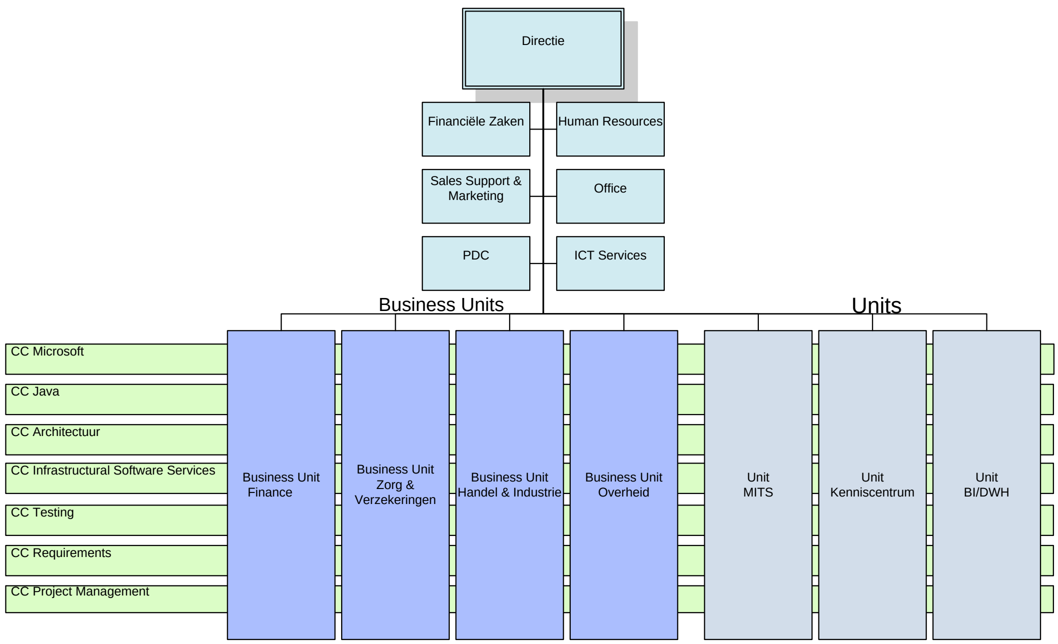

Figure 26 – Overview Competence Centers at Info Support

The advantage of using Competence Centers is that all people within that center have roughly the same area of expertise and come from different parts of the company. The advantage of making the Competence Center responsible for the courses on offer is that the developers are more committed to improve the curriculum. This view corresponds with [Mohr, 2005], which states that combining the knowledge of different people working at different function, technology, hierarchy, business, and geography is one of the key strategy points to become better at innovation.

A solution to the distance between the divisions and the Knowledge Center proposed by the Microsoft Technology Manager is to use the different Competence Centers to keep the accompanying curricula complete and up-to-date. Each Competence Center has its own area of expertise and should use the knowledge they possess to keep the accompanying courses up-to-date and make all needed knowledge available. A teacher who is responsible for new courses of the Java department is also part of the Competence Center Java. He states that this Competence Center is already having this policy of working together to keep informed of new technologies.

According to the Java Technology Manager, the Competence Center determines which part of the total supply of technology is used within the company. The curriculum should be matched with this selection, because the selected technology creates a high need for courses that supply the selected technologies. There are also different knowledge profiles defined which comply with functions within the company. The knowledge profiles are then evaluated on whether all courses are given by the Knowledge Center. The chosen technologies that are going to be part of every standard software development project are also being put into their software factory Endeavour.

The Competence Center Requirements Analysis is also involved in the education according to a Functional Designer from that Competence Center. Consultants from this competence center give all requirements analyses courses. In general, these consultants work in the field, except when they give courses. Most of the courses they give are the courses for the Professional Development Center to help users of the Endeavour software factory with the requirements analysis parts. They develop these courses themselves under the supervision of the Professional Development Center and the Knowledge Center. The rest of the courses they give use externally made material.

According to [Mohr, 2005], it is relevant to look around to see new trends appear. The Competence Center can help with this by making the participants aware of new problems at competing firms. The advantage of using the Competence Center for this is that a significant part of the participators is consultant. Consultants often work together with competitors in the same development teams. This way the external need is directly observable by the internal consultants, which enables proven techniques like empathic design to be used. Even though empathic design is originally meant for product design [Mohr, 2005], it is still applicable on the development of courses.

13.1.3. Empathic Design

According to [Mohr, 2005], empathic design is a good method to determine the real customer needs. Like a product, a course needs to fulfill certain demand from the consumer. By asking the consumer what he desires you often do not get the product the consumer is going to buy. That is why empathic design is used to observe behavior rather than asking directly. This is also the case with courses; people do not know what they miss, whether it is a tangible product or a course.

The courses from the Professional Development Center are gradually being integrated in software projects. The basic courses Endeavour are still presented in standard courses. Info Support is the developer of a software factory called Endeavour. This software factory is sold to other software developers to help them to bring structure in their software development process.

The more advanced Endeavour courses are shifting to just-in-time courses. These courses are supplying the needed information just before needed in the Endeavour projects. An advantage of this method is the direct feedback possible from the students and the direct link between theory and practice. This way the teacher can see how the students are using the knowledge provided by the course. This can lead to adjustment of the course or even the birth of new courses.

This trend within the Endeavour courses could be applied in the entire curriculum. However, the problem with this method is that it is not feasible to implement this system for small groups of students. It is not wise to put a teacher on small groups because the teacher needs the same amount of time for smaller and for larger groups. The groups will be smaller because external students often follow courses in small groups. An extra downside is that the teacher needs to travel to the external software development teams, which has its own down sides.

As mentioned before the Requirements Analysis Competence Center gives all courses by consultants who work in the field most of the time. The advantage of this is that they can keep track of new technologies they come across. The technological feedback is of the same order of magnitude in the teachers from the field and the just-in-time courses.

13.2. Resource Problems

At the Java Department, according to the Java Technology Manager, the problem is not the determination which technologies are available. The most important constraint they experience is the profitability of courses for new technologies. Although Info Support has five different price ranges for courses, the cost for a course cannot be raised infinitely. The reason for this is that the clients do not want to pay any price for the course even if the course has relevant specialized knowledge. Because the amount of Java programmers within Info Support is not very large, the internal market for courses is not that large either. Because of this lack of market, most technologies are not used for new course material.

The Microsoft Department is larger in both internal and external clients. Because of the larger market, the Microsoft courses are more likely to become profitable than the Java courses. This advantage combined with the advantage that Microsoft is more cooperative than the Java technology suppliers are gives the Microsoft courses an edge on being implemented.

13.2.1. Causes for low sales

According to the Java Technology Manager, there are four main causes for the lack of attendees. The first cause is that the reputation of Info Support is not good enough with their Java curriculum. If Info Support builds up a better reputation they can sell courses based on their reputation. Now they are one of the many suppliers, without having a unique selling point. To solve this anonymity, the quality of the Java courses needs to be communicated better to the external market.

Another cause is that the technology suppliers (like Sun) offer their own courses on some of the new technologies they develop. People tend to go to the supplier of the technology, because they perceive the supplier as being supplier of quality courses. The reason for this perception is that the supplier should know the technology best and was brilliant enough to develop the technology in the first place. It is hard to compete with a supplier who has this advantage. This cause exists because the Java department does not have the reputation of providing more value than the technology supplier provides.

There are more autodidact developers in the Java community than in the Microsoft community, which makes the target audience smaller. This limits the amount of courses they can sell. It is hard to tackle this problem. The only way to get autodidact developers to courses is to offer courses in which they can obtain valuable knowhow they cannot obtain from the internet.

The marketing is quite small for the Java courses they offer. The sales department focuses on Microsoft material since the offer of Java courses is not very large. However, in order to make the Java curriculum bigger the amount of attendees must increase too. Therefore, the departments are waiting on each other in order to grow. This situation can only be solved by intervention from the company itself.

13.3. Technology Suppliers

Because of the intense collaboration with Microsoft, it is possible to miss opportunities to give courses for products made by other software vendors. That is why it is relevant to keep an eye on other software vendors, who also generate a need for courses for specific products. It is not only relevant to get more opportunities, but also to have a larger spread in course material. When the spread is better, the dependence on Microsoft will be smaller and there will be more potential customers when the courses on offer have more variety. Note that Info Support also has a Java department, but the Sales Manager Learning Solutions has the tendency to think mainly about Microsoft products. It is vital that he considers the Java part of the curriculum more with customer visits, in order to increase the market share of Info Supports Java courses.

13.3.1. The open Java market

According to a Java teacher and initiator of new courses, the Java department has multiple suppliers to obtain new technology. The Java community is much more distributed than the Microsoft community is. There are conferences all over the world for Java technologies. These conferences are open to all who have new Java technology to share. This helps to let smaller suppliers show their products and technologies. This openness is a characteristic of the Java community.

It is an advantage that smaller suppliers get their share of attention, because the innovative products usually come from smaller companies. The disadvantage of the large spread in technology is that the users can use multiple technologies, which can lead to the necessity to know different technologies as a developer. In general, Info Support chooses a limited subset of the comparable technologies available.

The Java Technology Manager from Info Support is also chair of the Dutch Java group. Most Dutch Java developers are a member of this group. This group organizes every year two conferences in the Netherlands. Therefore, Info Support is one of the first companies to see the new Java technologies. Another advantage Info Support has is that the Java Technology Manager is also one of the few certified Java Champions. Java Champion is a title Sun gives to a limited amount of people. There are about a hundred people worldwide and only two in the Netherlands. The advantage of being a Java Champion is that new technologies from Sun are first announced to the Java Champions and later to the rest of the Java market.

13.4. Course Evaluation Forms

The current course evaluation form can be found in the appendices. This is the written version of the evaluation; the students fill out a digital version with the same questions.

13.4.1. Learned material utilization

The overall course evaluation form is not focusing on concrete relevant information. It only questions quality and not what the arguments are which contribute to the quality. The only information you can get from this is whether the quality is ok. You will not know what causes the quality to be suboptimal.

Take question one for example. The students are asked what the reason is for them to follow the course. However, they only ask which category the reason is in (solving a problem, preparation, developing new skills and knowledge, etc…). It is useful to know what problems they like to solve as specific as possible. If you have this information, you can see whether the student got the information he needed and whether his choice for that course was the right choice. You can also find out whether his desired knowledge is available at all at the knowledge center.

At question five, the satisfaction level is asked. This is important information. However, it is more important what did not meet the expectations and contributed to the satisfaction level. This is information relevant in determining better courses. It is also possible that the current students can be divided into separate groups who desire different courses. This way the coverage of the course material can be enlarged.

The downside of these questions is that people tend to give short answers to open questions in evaluation forms. It is useful to let the teachers ask the students directly what they expect to retrieve from a course. By asking the students directly what they expect, the answer will be more elaborate than the same question in an evaluation form.

13.4.2. Delayed Evaluation

The relevance of a course cannot be fully determined right after the end of the course; the relevance will not be clear until the learned material has been put to the test. That is why it is useful to ask the student what the relevance of the course was when he used the material in his job. At this point, the student knows better whether he missed information in the course. The current form asks for this information in question three. This question should be asked on a later moment. Besides, it misses the arguments responsible for the grade.

Since it is impossible to obtain this directly after the course, it is necessary to obtain this information on a later moment. Since it is not very likely they are filling out another evaluation form some time after attaining the course, it is more useful to obtain this information in a different way.

In the case of external students, it is possible to obtain this information when the Sales Manager Learning Solutions is visiting the customers. He can see what the former students are doing with the information they obtained by attending the course and determine whether the course was right for the student. Note that there is a difference between a satisfied customer and a best served customer. The second one got the information that contributed most to his targets. Of course, this is within the boundaries of the knowledge available within Info Support.

In the case of internal students, it is possible to give direct feedback to the knowledge center or by attaining it via the hierarchy of the company. It is much easier to obtain information from internal students because they work for the same team.

13.4.3. Suggestions

Ask the student for what specific task he uses the obtained knowledge. By getting a concrete answer from the student what the usage will be, it is possible to determine whether the course is supplying the right information for his goal.

Ask the student what knowledge he desired to obtain but did not get in the course he followed. This way it becomes clear what the student expects and does not get. This missing knowledge indicates that the information should be in the course, or there should be another more suitable course, which supplies this knowledge.

Ask the student after he has used the obtained knowledge again whether the knowledge helped to fulfill the described task. This way it becomes clear whether certain goals can indeed be achieved by the courses.

At that same time, ask the student whether the missing knowledge mentioned in the course evaluation form was indeed needed for his task and whether other knowledge was missing which he did not notice when he filled out the evaluation form.

14 Completeness of current determination process

In the first chapter of this part, the current methods are described to obtain demand from the market. In this chapter, the missing information in the current market determination is determined.

14.1. Creativity Techniques

Info Support is going the right way by using Empathic Design style approaches within the Competence Centers. A useful addition to this method is to use creativity techniques. These techniques are used to improve the idea generation. They all have in common that they encourage original and divergent thinking. A consultant from Info Support proposed a creativity technique he saw at another successful software development company.

Their solution to lower the threshold for ideas for new courses is to make a forum on which new ideas can be posted. Everybody from the company can post an idea and everybody can give comments on given ideas to improve the concept. Colleagues can also support these ideas to enlarge the chance the idea is used to improve the curriculum. This way there is a centralized brainstorm session in which every colleague can join. To make sure good ideas are executed, there must be a budget available for good ideas. With this method, good ideas will be noticed and executed.

[Mohr, 2005] supports this view in multiple ways. First, it is a way to reduce the threshold colleagues need to overcome to get their suggestion in the open. Second, it is also another way to express passion for discovery and novelty. By putting resources to this, the vision is more convincing than just stating it in the policy. Third, it is a method to increase the level of experimentation within the company.

14.2. External Cooperation

As elaborated before, the problem with the Java courses is that courses tend to sell mediocre. This makes it hard for the Java department to make a complete Java curriculum.

An idea from a Java teacher overcome this problem is to sell self-made courses to other companies who give courses or sell course material. There are two types of partnerships according to [Mohr, 2005], the horizontal and vertical partnerships. The vertical partnerships relevant for Info Support are partnerships with the suppliers and customers.

14.2.1. Vertical Partnerships

In the case of Info Support, the suppliers are the companies in charge of making new technology and the customers are the people who buy the courses. The customers are already involved by input via courses (evaluation and remarks during the courses) and by visits by the Sales Manager Learning Solutions.

The large suppliers of Java Technology have been approached for the realization of partnerships in the creation of course material. However, these proved to be no profitable partner in course material. The reason for this is the unrealistic financial demands they make. Therefore, besides the fact that it is not often possible to make courses for technologies published by the larger suppliers because they make their own courses. They also do not really want to work together. This is an important plus for the Microsoft market, since Microsoft is collaborating with other companies who want to make courses for their products.

14.2.2. Horizontal Partnerships

The horizontal partnership type is another viable option for partnerships. These are competing companies, who can save production cost when cooperating. [Mohr, 2005] uses the names competitive collaboration and co-opetition for this type of partnership.

Contrary to Java technology suppliers, there are suppliers of courses who are willing to cooperate in the creation of course material. This can lead to a better return on investment of the course generation, because the investment of the course generation can be split over the two companies. This makes it possible for Info Support to make the curriculum cover more technology without an increase of investment. The advantage is that internal developers can obtain the technologies easier and cheaper.

At the Microsoft department there is already a knowledge sharing with competing firms according to a teacher at Info Support. These companies get special courses for trainers to make them able to learn the material to their students. Microsoft is even directing competing firms to Info Support to learn how to give the courses.

There are also courses available to buy from those other suppliers. The advantage is that the course material is obtained relatively cheap. However, the courses they offer are limited and should therefore not be the sole source of new courses. These courses are a fundamental part of the total curriculum of Info Support.

14.3. Priority problem with Customer Relations

The Sales Manager Learning Solutions is working at the Knowledge Center, which makes him focus on the curriculum of the Knowledge Center. The knowledge center takes responsibility for only the courses that are of the knowledge center. However, other departments like Professional Development Center also provide courses and teachers to give them. These departments do not have their own managers for customer relations. Therefore, the improvement of the curriculum of the other departments does not have the same priority as the improvement of the curriculum from the Knowledge Center. More courses can be sold/enhanced/developed when the awareness of other departments is higher within all different departments.

Another advantage when the awareness of other departments is higher is that the knowledge center can gather more customer information. This information can be used to improve the courses given by the other departments.

It is good to see that Info Support is moving the responsibility for all courses to the Knowledge Center. This solves the problem that the courses belong to different departments.

14.3.1. Endeavour product deployment

The Endeavour software factory is currently sold to customers with only courses that supply technological knowhow. However, it is also useful to complement these courses with courses that help them organize the change in structure of their software development process. It is useful to teach the customer how to redesign their software development process so they can use the software factory in its full potential. This lack of knowledge at the customer's side is common because they use the Endeavour software factory to formalize their product development methods. This addition of knowhow makes Endeavour more valuable for the customer as a product.

14.4. Specialising general courses

As earlier explained the courses not given by Info Support because they are too large scaled are a potential source of more specialized course material. If you can identify relevant groups within the main target group, the course can be tailored to these groups to make the course more relevant for them. This way the courses that are under heavy competition can be offered with benefits for the specified target (sub-) group. Existing channels like student feedback can achieve this during courses or by feedback in the just-in-time courses mentioned as an improvement for the course integration.

14.5. Tapping into new markets

The current method for determining the demand is focusing on volume software, which creates a demand for courses. However, by focusing on volume software other important markets can be overlooked. The market for specific tailored software products can also be large when there is a large group of users. This is only the case when either the customer has many employees who use the product or when multiple organizations buy software together.

One of the largest organizations in the Netherlands is the government. The government has enough employees to make specific courses for them. There are large groups of users of the same specific software. Because of the size of these groups, this can be a profitable customer for the Knowledge Center to have. The disadvantage of this method is that the consultants from Info Support do not need this specialized information, while the policy of Info Support is that courses must have a benefit for internal consultants. This policy makes it unlikely that Info Support will enter this market.

An idea from a Functional Designer is to set up collaboration with a Scandinavian company. This company provides training in Software Process Improvement. They are large in some countries but have not been introduced in the Netherlands yet. They have very much knowledge of best practices in software development. The courses would be a nice addition to the curriculum of Info Support.

15 Conclusions

As we have seen in chapter 12, Info Support utilizes the standard known methods for demand determination. Traditionally, Info Support is working tightly together with Microsoft to find new market demands. The partnership with Microsoft proves to be a stable source to keep the curriculum adequate for the demands from the market.

Info Support is aware of the danger of relying too much on a single partner and therefore does Java development and supply Java courses. To increase the size of the java department, it is important that the Java department of the Knowledge Center is showing to the market that they offer interesting high quality courses. This can be achieved by better promotion by the Sales Manager Learning Solutions. A good method to improve the perceived quality is to give courses about new technologies before the competition can.

Besides working together with Microsoft, it is relevant to work together with other firms who give courses or create courses. This enables Info Support to make more courses profitable to include in the curriculum. In the end, internal consultants will have more knowledge available to learn internally.

Info Support is aware that the feedback via business units is not the best way to communicate internal needs and is therefore shifting to more communication via the competence centers. These competence centers are all about technology and therefore the right place to determine needed courses. In addition to the competence centers role in technology management it is wise to invest creative techniques like a forum to post ideas on to lower the threshold for the consultants to give their input for a better market determination.

The different departments who are supplying courses are going to be centralized in terms of responsibility, which improves the market research concerning other departments than the Knowledge Center. The market for soft skills is getting more important for Info Support. Therefore, the desire to make these trainings themselves arises.

The Professional Development Center is working on the integration of their courses within software projects. It is relevant to determine whether more courses in the entire curriculum are beneficial to give in this style.

Part IV

16 Future Research

16.1. Generic Timetable Representation

The best internal representation of a timetable is dependant of the type of scheduling problem and the dominant kind of restrictions within that timetable. In order to make a true generic scheduling algorithm it is necessary to give the user the choice what representation he wishes to use. In order to achieve this it is necessary to make the representation at runtime. When the representation is build, it is necessary to adapt the access functions to the chosen representation. The behavior of the access functions need to be determined at runtime too, this is makes the problem challenging.

I have chosen not to research this problem in this thesis because it does not contribute to my core research. This is due to the fact I will evaluate only one type of timetable. It will be possible to insert this extra functionality in the algorithm without much effort.

16.2. Concrete Program Additions

The following issues are no pure research problems. They are non-trivial problems nonetheless.

16.2.1. Restriction Checker

The current implementation of the MA is trying to make a timetable based on the given requirements. It does not check on mutual exclusive requirements however. A trivial example is that a teacher is given who should be giving courses but there is no course given he is allowed to give. This will always give a timetable that violates the restrictions. The delivered timetable will have a balance between low scheduled time and scheduling on courses he is not allowed to give.

16.2.2. Calculation Time Estimator

Due to the probabilistic nature of genetic algorithms, it is a challenge to give an expected calculation time for an acceptable timetable. This problem can be viewed from a mathematical point of view and from an empirical point of view. The calculation time is dependent on the size of the timetable and the tightness of the restrictions given. Take for example a timetable consisting of two courses versus a timetable consisting of twenty courses. The first one has only a few values to optimize; the second one has lots of them. The tightness of restrictions is expressing the ratio of acceptable timetables compared to the entire set of timetables.

17 Table of Figures

18 References

Alkan, A., Őzcan, E. (2003). Memetic Algorithms for Timetabling. *The 2003 Congress on Evolutionary Computation*, *3*, pp. 1796-1802.

Burke, E.K., Landa Silva, J.D. (2004). The Design of Memetic Algorithms for Scheduling and Timetabling Problems. *Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing*, , pp. 289-312.

Eiben, A.E., Smith, J.E. (2007). Introduction to Evolutionary Computing. : Springer.

Even, S., Itai, A., Shamir, A. (1975). On the complexity of time table and multicommodity flow problems. *16th Annual Symposium on Foundations of Computer Science*, , pp. 183-193.

Holland, J. (1975). Genetic algorithms in search, optimization, and machine learning.. : Michigan Press.

Mitchell, M., Holland, J.H., Forrest, S. (1994). When Will a Genetic Algorithm Outperform Hill-Climbing. In , *Advances in Neural Information Processing Systems 6*. : Morgan Kaufman. pp. 51-58.

Mohr, J., Sengupta S., Slater, S. (2005). Marketing of High-Technology Products and Innovations. : Pearson Prentice Hall.

Monfroglio, A. (1996). Timetabling through Constrained Heuristic Search and Genetic Algorithms. *Software - Practice and Experience*, *26*, pp. 251-279.

Wolpert, D.H., Macready, W.G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, *1, issue 1*, pp. 67-82.

Őzcan, E. (2005). Memetic Algorithms for Nurse Rostering. *Computer and Information Sciences - ISCIS 2005*, *3733*, pp. 482-492.

Appendice A. Course Evaluation form

METRICS THAT MATTER

Instructor-Led Training - Post-Event Survey

Naam docent: ____________ Cursus: ________________

Einddatum cursus: _________

Naam cursist:____________________________ E-mailadres cursist:______________________________

Bedrijf: ____________________________________ Functie: _________________________________

De kwaliteit van uw ervaring tijdens deze training is erg belangrijk voor ons en uw commentaar en opmerkingen vormen een wezenlijk onderdeel van onze kwaliteitsbeheersing.
Neem alstublieft een moment de tijd om ons uw commentaar te geven. Dank u.

1. Wat is voor u de belangrijkste reden dat u deze training hebt gevolgd?

☐ Oplossing van een bepaald probleem
☐ Voorbereiding op het implementeren of bijwerken van een bepaald-product
☐ Ontwikkeling van nieuwe vaardigheden en kennis (niet in verband met de implementatie van nieuwe software)
☐ Voorbereiding op een certificeringsexamen
☐ Beter inzicht in producten voordat ik nieuwe software aanschaf
☐ Voorbereiding op een verandering in mijn loopbaan
☐ Anders (geef een omschrijving) ______________________

2. Hoe beoordeelt u deze training op de volgende aspecten? Geef een cijfer op een schaal van 9 tot 1, waarbij 9 staat voor uitzonderlijk of uitstekend en 1 voor onacceptabel of slecht.

Leslokaal	**Uitzonderlijk/ uitstekend**								**Onacceptabel/ slecht**	N.v.t.
	9	8	7	6	5	4	3	2	1	O
Kwaliteit van leslokaal	O	O	O	O	O	O	O	O	O	O
Prestaties van technologie die in leslokaal wordt gebruikt (hardware/ software)	O	O	O	O	O	O	O	O	O	O

Eventuele opmerkingen over leslokaal ______________________

Docent	**Uitzonderlijk/ uitstekend**								**Onacceptabel/ slecht**	N.v.t.
	9	8	7	6	5	4	3	2	1	O
Inhoudelijke kennis van instructeur	O	O	O	O	O	O	O	O	O	O
Beantwoording van vragen door instructeur	O	O	O	O	O	O	O	O	O	O
Mate waarin instructeur praktijkervaringen en -voorbeelden aandraagt	O	O	O	O	O	O	O	O	O	O
Presentatievaardigheden van instructeur	O	O	O	O	O	O	O	O	O	O
Algehele prestaties van instructeur	O	O	O	O	O	O	O	O	O	O

Eventuele opmerkingen over instructeur ______________________

METRICS THAT MATTER

Instructor-Led Training - Post-Event Survey

Selecteer N.v.t. in het gedeelte Trainingsinhoud als tijdens uw training geen praktijkoefeningen hebben plaatsgevonden.

Trainingsinhoud	**Uitzonderlijk/ uitstekend**								**Onacceptabel/ slecht**	N.v.t.
	9	8	7	6	5	4	3	2	1	o
Duidelijkheid van trainingsinhoud	o	o	o	o	o	o	o	o	o	o
Logische opbouw van trainingsinhoud	o	o	o	o	o	o	o	o	o	o
Technische diepgang van trainingsinhoud	o	o	o	o	o	o	o	o	o	o
Effectiviteit van praktijkoefeningen als aanvulling op aangeleerde kennis en vaardigheden	o	o	o	o	o	o	o	o	o	o
Relevantie van praktijkoefeningen voor feitelijke werksituaties	o	o	o	o	o	o	o	o	o	o
Effectiviteit van praktijkoefeningen binnen elke les als aanvulling op aangeleerde kennis en vaardigheden	o	o	o	o	o	o	o	o	o	o
Tijd voor activiteiten als discussies en praktijkoefeningen (naast theorie)	o	o	o	o	o	o	o	o	o	o
Taalkundige kwaliteit van cursusmateriaal (grammatica, terminologie, stijl)	o	o	o	o	o	o	o	o	o	o

Wat zou u willen veranderen in de trainingsinhoud om uw leerervaring te verbeteren? ______________________

Eventuele overige opmerkingen bij trainingsinhoud ______________________

3. Hoe beoordeelt u deze training op de volgende aspecten? Geef een cijfer op een schaal van 9 tot 1, waarbij 9 staat voor uitzonderlijk of uitstekend en 1 voor onacceptabel of slecht.

Didactische effectiviteit	**Uitzonderlijk/ uitstekend**								**Onacceptabel/ slecht**	N.v.t.
	9	8	7	6	5	4	3	2	1	o
Verworven kennis en vaardigheden	o	o	o	o	o	o	o	o	o	o
Invloed van training op effectiviteit van uw werkzaamheden	o	o	o	o	o	o	o	o	o	o
Invloed van training op uw doorgroeimogelijkheden	o	o	o	o	o	o	o	o	o	o

4. Hoe waarschijnlijk zou u het volgende doen op basis van uw ervaring met deze training?

	Absoluut	Waarschijnlijk	Neutraal	Waarschijnlijk niet	Absoluut niet
Deze training aanbevelen bij iemand anders	○	○	○	○	○
Binnen 12 maanden nog een training volgen	○	○	○	○	○
Meer bepaalde producten en -technologieën methodes gebruiken	○	○	○	○	○

5. Hoe goed voldeed deze training aan uw verwachtingen?

Ruimschoots								Totaal niet
9	8	7	6	5	4	3	2	1
○	○	○	○	○	○	○	○	○

6. Hoe tevreden bent u over het geheel genomen met deze training?

☐ Zeer tevreden
☐ Enigszins Tevreden
☐ Enigszins Ontevreden
☐ Zeer ontevreden

7. Welke van de volgende beschrijvingen is het meest op u van toepassing?

☐ **IT-professional**: IT-manager of -medewerker die vooral actief is op een geformaliseerde IT-, IS- of MIS-afdeling.
☐ **Ontwikkelaar**: iemand die softwaretoepassingen ontwikkelt, complexe macro's samenstelt, computercode schrijft of websites maakt voor zichzelf en/of anderen.
☐ **Beleidsmaker**: iemand die op het werk een belangrijke rol speelt bij de besluitvorming voor de eigen afdeling of meerdere afdelingen, maar NIET op de IT-afdeling werkt.
☐ **Informatiewerker**: iemand die een computer of apparaat gebruikt om informatie op het werk te creëren, beheren, verwerken of veranderen.
☐ **Thuisgebruiker**: iemand die thuis een computer voor persoonlijke doeleinden gebruikt.
☐ **Anders** (geef een omschrijving) ____________________

8. Hoe lang werkt u al in de IT-branche?

☐ Minder dan 1 jaar
☐ 1-2 jaar
☐ 3-4 jaar
☐ 5-6 jaar
☐ 7-9 jaar
☐ 10 jaar of meer

9. Hoeveel PC's, inclusief laptops, zijn er ongeveer in uw hele organisatie (inclusief locaties)?

☐ Nul pc's
☐ 1-24 pc's
☐ 25-499 pc's
☐ 500+ pc's
☐ Weet niet
☐ Werk op dit moment niet / niet van toepassing

10. In welke mate gebruikt u de in deze cursus behandelde onderwerpen in uw dagelijkse werkzaamheden? Slechts één aankruisen.

☐0% ☐10% ☐20% ☐30% ☐40% ☐50% ☐60% ☐70% ☐80% ☐90% ☐100%

Instructor-Led Training - Post-Event Survey

11. Als u vindt dat deze training een positief effect heeft op uw functioneren, op welke gebieden dan het meest? Alles aankruisen dat van toepassing is.
    - ☐ kwaliteit verhogen
    - ☐ productiviteit verhogen
    - ☐ verkoop verhogen
    - ☐ mogelijkheid tot innoveren vergroten
    - ☐ verminderen van tijd om taken te doen
    - ☐ kostenverlaging
    - ☐ klanttevredenheid verhogen
    - ☐ verhoging werknemer tevredenheid

12. Hoe groot schat u het positieve effect van de cursus op uw functioneren.
    ☐0% ☐10% ☐20% ☐30% ☐40% ☐50% ☐60% ☐70% ☐80% ☐90% ☐100%

13. Hoeveel van de verbetering in uw functioneren zal een direct resultaat zijn van het volgen van deze cursus? (Als u bijvoorbeeld denkt dat de helft van uw verbetering een direct resultaat is van deze training, vul dan hier 50% in.)
    ☐0% ☐10% ☐20% ☐30% ☐40% ☐50% ☐60% ☐70% ☐80% ☐90% ☐100%

14. Wilt u op de hoogte gehouden worden van andere, aanvullende en/of vervolgcursussen?? ☐ Ja ☐ Nee

Dank u voor uw medewerking.