Multi-Objective Optimisation of a Capacitated and Dynamic Multi-Item Inventory System using Physical (-related) Metaheuristics

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)
Naturwissenschaftliche Fakultät II - Physik
Universität Regensburg

vorgelegt von
Markus Albert Zizler
aus Steinberg am See
Februar 2008

Das Promotionsgesuch wurde eingereicht am 1. November 2007.

Diese Arbeit wurde angleitet von Prof. Dr. Ingo Morgenstern.

Prüfungsausschuss:

Vorsitzender: Prof. Dr. Jascha Repp (Physik)

1. Gutachter: Prof. Dr. Ingo Morgenstern (Physik)
2. Gutachter: Prof. Dr. Rainer Gömmel (Wirtschaftswissenschaften)

Weiterer Prüfer: Prof. Dr. Tilo Wettig (Physik)

Termin des Promotionskolloquiums: 19. Februar 2008

Contents

Preface **5**

1 General Introduction **9**
1.1 History of Operations Research (OR) 10
1.2 OR-Process 12
1.3 Combinatorial Optimisation 15
1.3.1 Basic Terms 15
1.3.2 Complexity 18
1.3.3 Multi-Objective Optimisation 21
1.4 (Meta-)Heuristics 22
1.5 Standard Methods and Problems of OR 27
1.5.1 Simplex Algorithm 27
1.5.2 Branch & Bound - BB 28
1.5.3 Traveling Salesman Problem - TSP 31
1.5.4 Different Problems 32
1.6 Simulation as Method of Optimisation 34

2 Physical Optimisation **37**
2.1 Spin Glasses 38
2.1.1 Magnetism 38
2.1.2 Theoretical / Experimental Results 39
2.1.3 Mathematical Spin Glass Models 44
2.2 Monte-Carlo-Methods 48
2.2.1 Statistical Physics 48
2.2.2 Simple Sampling 49
2.2.3 Importance Sampling 50
2.3 Optimisation Algorithms 53
2.3.1 Simulated Annealing - SA 53
2.3.2 Threshold Accepting -TA 55
2.3.3 Great Deluge Algorithm - GDA 56
2.3.4 Cooling Scheme 56

3 Different Metaheuristics **59**
3.1 Genetic Algorithms - GA 60
3.1.1 Biological Background 60
3.1.2 Algorithmic Realisation 62
3.1.3 Genetic Operations 68
3.1.4 Miscellaneous 72
3.2 Ant Colony Algorithms 75
3.3 Tabu Search - TS 78

4 Theory of Inventory Control **83**
4.1 Introduction 84
4.2 Single-Item-Models 87
4.2.1 Deterministic Models 87
4.2.2 Stochastic Models 92
4.3 Multi-Item-Inventories 95
4.3.1 Flaccidities of Single-Item-Models 95
4.3.2 Multi-Item-Models 96
4.4 Forecasting 99
4.4.1 Different Types of Forecasting Methods 100
4.4.2 Monitoring Forecast Systems 105
4.4.3 (Auto-)Correlation 108

5 Physical Optimisation and Forecasting **111**
5.1 Short Term Forecast 111
5.1.1 Model with Simple Deviation 111
5.1.2 Model with Value at Risk 113
5.1.3 Application to Grades of Soccer Players 114
5.2 Medium Term Forecast 118

6 Optimisation of an Inventory System **121**
6.1 Implementation of an Inventory Problem 121
6.1.1 Variables of the Inventory System 121
6.1.2 Hamiltonian 122
6.1.3 Standard Parameter Configuration 124
6.1.4 Standard Configuration + Stochastic Lead Time 125
6.1.5 Standard Configuration + Capacity Restriction 125
6.2 Inventory Optimisation - Part I 126
6.2.1 (s,Q) - Level Inventory Policy 126
6.2.2 (t,S) - Cycle Inventory Policy 131
6.2.3 (s,S) - Level Inventory Policy 132
6.2.4 Application of the Different Policies to Future Periods . . . 133

6.2.5 Sales Figures of a Steel Company . . . 134
6.3 Inventory Optimisation - Part II . . . 136
6.3.1 Implementation of further Parameters . . . 136
6.3.2 Simulation Results . . . 139
6.4 Physical Structures in Inventory Control . . . 144
6.4.1 Equivalence of the Systems . . . 144
6.4.2 Optimisation Methods . . . 145
6.4.3 Equivalence of the System Variables . . . 145
6.4.4 Differences . . . 147

7 Physical Optimisation by Comparison 149
7.1 Genetic Algorithm . . . 149
7.1.1 Implementation . . . 149
7.1.2 Simulation Results . . . 149
7.1.3 A new Optimisation Algorithm ? . . . 152
7.2 Results of the Research Community . . . 153
7.2.1 Overview . . . 153
7.2.2 Different Papers . . . 154
7.2.3 Mathematical Methods . . . 159
7.2.4 Delineation . . . 160

8 Summary 161
8.1 Forecasting . . . 161
8.2 Inventory Optimisation . . . 162
8.3 Conclusion . . . 164

Bibliography 165

Index 173

List of Figures 174

List of Tables 175

Acknowledgements 177

Preface

There are many optimisations in nature and in the world of physics. Light for example tries to find the way with the shortest time; in mechanics the body movement follows the restrictions of extremal principles. In biology those individuals survive that adapt most efficiently to their environment. Human beings optimise, too: strategies in production, in the service sector or in personal affairs are just series of optimisation actions under restrictions. But there is a fundamental difference between optimisation in nature and in human society: nature knows the best solution automatically; whereas human beings have to make some calculations at first. Optimisations have a great relevance in mathematics, engineering, economy, informatics and a lot of other areas: the optimal workload of production units, the arrangement of electronic circuits on a chip or the cheap laying of water pipes are only a few examples to mention in this respect.

The list can easiliy be extended. There is nearly no area in production and service that is not involved. In a competitive economic system optimisations are not only important, but even necessary, especially if there is much money involved. It is the basic rule of a well functioning economy to reach the best performance with a minimum of ressources.

Nowadays the economic world is characterised by diversity and complexity of items as well as dynamic and international markets. Therefore the competition is getting stronger and ressources like energy, raw materials, inventory and production capacities have to be used wisely. Conditions for being able to cope are the following: high customer service and quality standards, flexibility of production, short production times and especially low costs in all areas.

As a reaction to these conditions, a structural adjustment is necessary. Beside strategic concepts like *lean production*, *lean management*, *outsourcing* and *enterprise ressource planning*, *process optimisation* in the context of *business reengineering* is gaining more and more influence. Former function oriented organisation forms are replaced by process oriented concepts. Thus the effectivity of single processes is going to be increased, because the administration effort between different departments of a company can be extremely reduced. But the high dependency of the subsystems causes also a high level of complexity which

cannot be understood by a single manager. Standard methods for supporting the manager in finding the optimum of the parameters have a restricted performance to special problems. Therefore new concepts are necessary for a further optimisation of business processes. One of those relatively new concepts are physical optimisation algorithms, meanwhile known in science and practice.

In physics there are many complex systems to optimise. The laws of thermodynamics state that every material near temperature zero is going to take the state with the lowest energy: the so called **ground state**. At low temperatures therefore all atoms of the most solid state bodies should arrange regularly in three space dimensions; these ordered and ideal solid state bodies are called crystals.

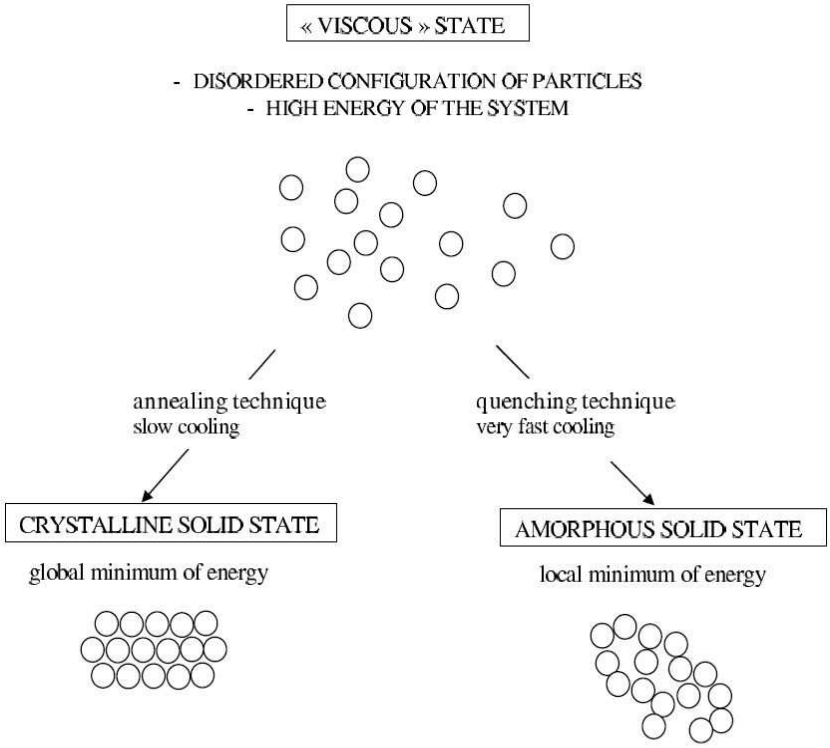

Figure 1: Possible outcomes of the annealing process

But in reality there is no ordered structure. One reason for this is that the atoms lose their energy too fast (*quenching*) to be arranged in an energetically ideal position when the solid state body is formed out of the melting; the system doesn't reach the ground state.

In order to reach the ground state, the solid state body has to be heated over the melting point and then cooled down very slowly in thermal equilibrium; thus the system itself is going to find the optimal state. This proceeding is called *annealing* and was reproduced from Metropolis et al. on the computer.

The capability of the algorithm simulated annealing was shown in simulating the cooling procedure of crystals, whose ground state was known. Because of this the algorithm then was used for systems (e.g. spin glasses), whose ground state was not known. Later Kirkpatrick [KGV83] proposed to use this kind of simulation for economic optimisation problems. Therefore he transferred the relevant economic variables to the physical equivalents.

Theoretical solid state physics is the basis of these physical optimisation algorithms. The assignment of economic variables to physical ones makes it possible to use the natural organisation process as global optimisation strategy. Thereby the parameters are interpreted as physical degrees of freedom and the cost function as energy. This logic represents an all-purpose optimisation algorithm for complex and correlated economic problems which can be applied to many problems, for example route planning or inventory control.

The ambition of this work is, to apply physical optimisation algorithms to the economic problem of inventory control. In a first and introductive step a "physical" forecast of the future demand shall be provided; the results are compared to standard methods of forecasting. The second and main part is to opimise the process of inventory control itself. Thus the physical algorithm tries to find the optimum way of ordering items for the inventory under widely realistic restrictions and constraints.

In chapter 1 a general introduction is given to operations research (OR), its standard methods and the connection to business informatics; besides a short overview about combinatorial optimisation (inventory control has combinatorial complexity) and metaheuristics (physical algorithms belong to this class of optimisation algorithms) is given. Chapter 2 initially gives information about the physical background from which the algorithms are derived; then the theory of the physical optimisation algorithms itself is described. After that, other metaheuristics which don't have a physical background, but work in a similiar way shall be explained (Chapter 3). Those are genetic algorithms, evolution strategies, tabu search and ant colony optimisation. The relevant theory of inventory control is described in chapter 4: single-item- and multi-item-models as well as the basics of forecasting. The results of the optimisation and simulation are stated in the chapters 5, 6 and 7.

Thereby the inventory problem is not just optimised with physical algorithms and compared to other methods (especially a genetic algorithm), but also regarded as physical system; thus the similarities between spin glasses and inventory control are worked out, too.

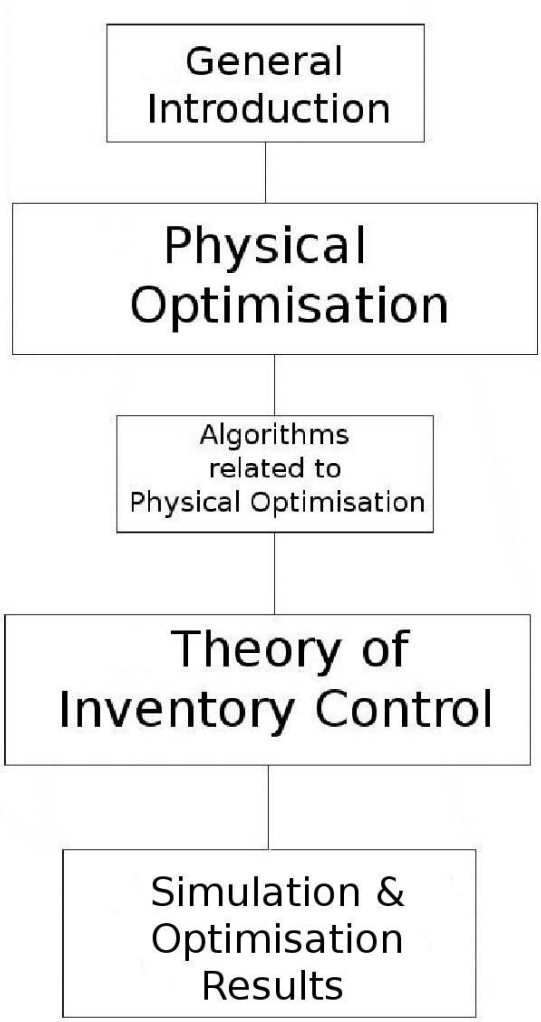

Figure 2: Structure of the dissertation

Chapter 1

General Introduction

This work is interdisciplinary and aligned to the less researched area between economic and natural science. On the one hand inventory control is a classical economic problem and on the other hand physical optimisation methods are attached to the name giving discipline. By now the application of these methods is relatively established in operations research and therefore a historic overview is given in 1.1; the OR-process is described in 1.2. In 1.3 the basic features of combinatorial optimisation are illustrated. 1.4 deals with the main features of (meta-)heuristics as a special kind of optimisation methods. In 1.5 standard optimisation methods and different established problems are presented. And at last in 1.6 a short introduction to simulation as method of optimisation is given.

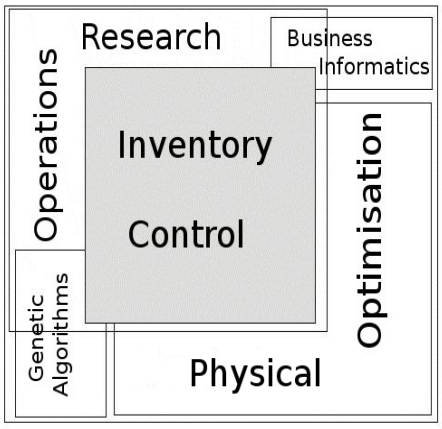

Figure 1.1: Classification of the dissertation

In Figure 1.1 the dissertation is classified in terms of different research areas.

One central subject of this dissertation is inventory control and thus it is in the center. But it is not the only main part: operations research and physical optimisation are applied to inventory control and thus there is a big overlap. If a strength classification of the dissertation is necessary, it would be assigned to OR, because physical optimisation and genetic algorithms are already established in OR. Beside there is also a link to business informatics, because the programmed optimisation algorithms could be implemented in a real inventory as a tool of optimisation or analysis. Nonetheless the dissertation is also a physical one, because mostly physical related algorithms are used and the results are analysed physically.

1.1 History of Operations Research (OR)

In a simple sense, OR is the use of general scientific methods for the study of any problem. The technology was developed from physicists, mathematicians, statisticians and biologists; thus OR is a conglomerate of different scientific branches.

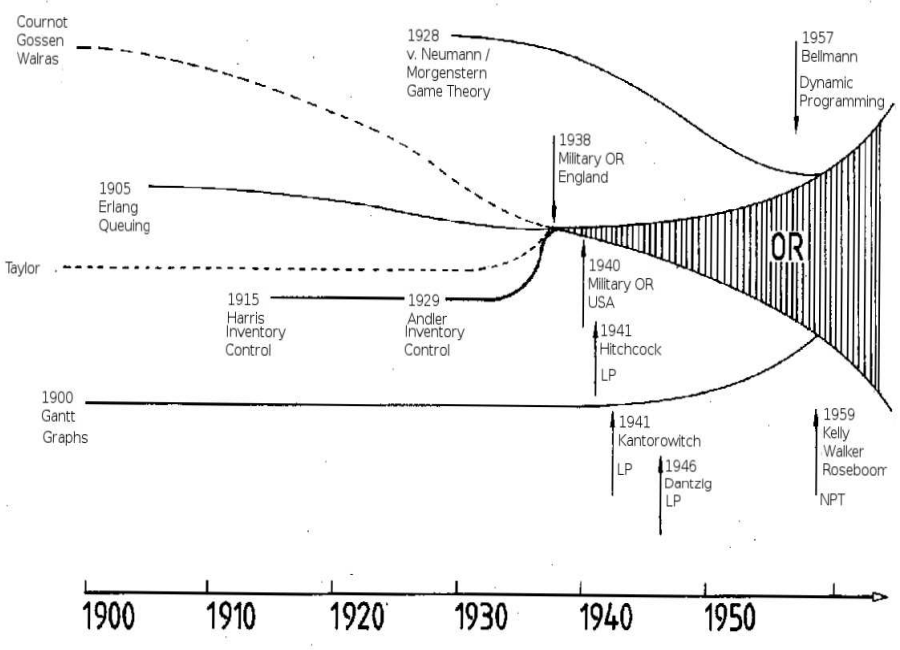

Figure 1.2: History of OR

The birth of OR was in the early forties. At first the new methods were used

as instruments for project planning of convoy optimisation in submarine combat and in the development of radar. During the fifties and sixties those methods were universally used. The enthusiasm of the seventies was followed by the disillusion of the eighties, because not each decision problem could be transformed in a good mathematical model. But in the nineties there was a re-animation in OR due to the progress in informatics and data processing. Beside others OR is used in following areas:

- Network analysis as planning and controlling instrument in aircraft / shipbuilding, coverage projects, etc.
- Linear optimisation of material flows, reload problems, production / finance / investment planning
- Inventory control
- Stowage problems with pallettes and containers in trucks, trains and ships.
- Planning of tours and modelling of tariffs

Thereby **linear programming (LP)** is the basic method of OR. Reasons for this lie in the early development of software packages on a commercial level. Already in 1970 all the essential theoretical knowledge for an effective treatment of LP problems was available. Beside pure linear problems, LP was used for problems with a *partly* linear structure:

- **quadratic programms** with a quadratic objective function and linear restrictions
- **quotient programms** whose objective function is a fraction of linear expressions; the restrictions are also linear
- **separable programms** are non-linear problems, which can be linearised in parts
- **stochastic programming** with random variables as model parameters

The great family of combinatorial optimisation problems cannot be treated with differential calculus. The coordination of machines to locations or applicants to jobs for example belongs to this group. The so called **dynamic programming (DP)** tries to find a solution for such problems. DP seperates the difficult global problem into parts, which are easier to solve. Beside DP the often used method of **branch & bound** works in the same way. **Heuristics** are another method to solve combinatorial optimisation problems. Those are methods, which find a good solution for most problem instances without being able to give a proof for this. OR also contains methods, which primarily do not optimise:

- **game theory** tries to find a solution for conflict situations; thus it has great value in explaining human behaviour
- **queueing theory** deals with stochastic processes. Queue systems help to find the right dimension of cashpoints in a warehouse or counters in an airport

Beside the described classical methods of OR there are other application areas, which can be associated with OR or related areas: *fuzzy decision models* allow an element of a set to be *between* zero (no) and one (yes). This means that an element does not need to belong clearly to a set; rather the element can partially belong to the set. This onset reflects human behaviour better than an inflexible yes or no. Practically those methods are used in investment and finance. Another big part of OR are *metaheuristics* to which biological and physical optimisation algorithms belong. The performance of modern computers enables the algorithmical imitation of intelligent behaviour. In literature these systems are summarised under the concept of **computational intelligence (CI)**. This name points out the relation to the research area of artificial intelligence (AI). Another aspect of CI is that these methods are strongly orientated at numerical mathematics and can only be realised with computer simulations. In contrast, methods of AI like expert systems have their focus on knowledge administration. But the transition between both research areas can be quite smooth. The methods of CI are often characterised as intelligent, because they have special attributes: they are flexible, discovering, explaining and able to learn and adjust. Not each method of CI shows all mentioned characteristics. Every technique has its own strength and weakness; thus it has to be proved, whether it can be used in a special application field. Concerning this dissertation it shall be tested, in what way physical (and genetical) algorithms can be used to optimise an inventory system.

1.2 OR-Process

Practical operations research demands many different activities. The totality of those activities is called "OR-process". This process consists of three parts:

1. Construction of one or several models
2. Implementation of optimisation methods
3. Transfer of the results to reality

In the first part the problem has to be identified, analysed and formulated for the construction of one or several models. Secondly, the model is optimised with

different algorithms: standard algorithms for example are available for models of linear optimisation and so called heuristics for combinatorial optimisation problems.

Optimisation models should lead to *optimal* solutions as decision proposals. Therefore *clear* objectives have to be fixed at first and the full scope of possible decisions has to be integrated in the model. In general, optimisation models consist of one objective function and at least one restriction, mostly in form of an inequation. Normally there are several restrictions, but rarely there is more than one objective function. In a narrow sense, optimisation models can only be used, if there are no external decision alternatives and when there is only one possible development; but that is true for just a few cases. In most decision situations there are several external alternatives and different possible developments of the environment. For decision preparation each alternative should be evaluated for every environmental possibility. **Simulation models** can perform such tasks: they simulate the different decision constellations. But this is an optimisation only in the sense that the most promising alternative is chosen; it is no optimisation within the model. For several possible developments each one should be weighted with the expected probability in order to get a clear evaluation of the different decision alternatives. The determination of the best alternative is usually complicated by the fact that several decision criteria have to be considered at the same time. Thus the advantage of an alternative concerning one criterion perhaps is balanced by the disadvantage of another one. But there is a field inbetween the opposite models of simulation and optimisation. Take as an example the decision problem between different investments in production machines. Thereby for each investment the optimum could be calculated with linear programming. This would not be an optimisation of the whole decision problem, but one of the external decision between alternatives. The optimisation model would have a "simulation model" above.

Thirdly, the results of the models have to be transferred to reality. Sometimes it is possible to use directly the results of optimisation in reality; in case of a simulation model above an optimisation model the results have to be interpreted by the decision maker. The process of optimisation, simulation and interpretation is always executed and supported by a computer and thus there is a strong connection between OR and **business informatics**. Obviously, this science has its roots in informatics and economics. It deals with planning, development, management and the efficient use of information and communication systems. Those systems are used for the support of business processes and decision making in companies and public administration. *Management support systems* (MSS) play a major role in information management. MSS are computer systems, which collect information from internal and external sources. Further on the information is formally prepared for the management. Concrete examples for MSS are manage-

ment information systems (MIS), decision support systems (DSS) and execution information systems (EIS). A symbolic illustration is shown in Figure 1.3. In

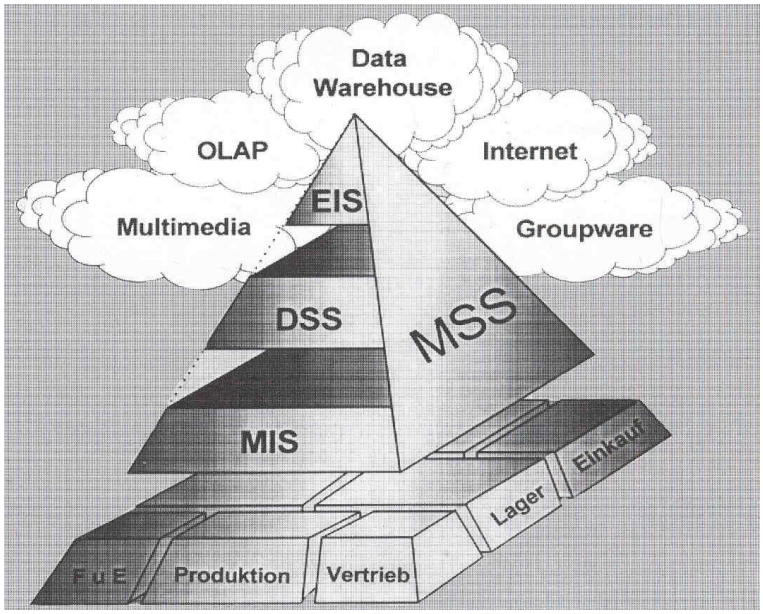

Figure 1.3: Classification of management support systems [GG98]

association with the system pyramid of a company, the categories MIS, DSS and EIS in the upper part of the pyramid are attached to the systems of controlling and planning. The clouds are possible extensions, which shall not be discussed here. The lower building stones are the departments of a company for which the MSS can be used.

DSS are interactive computing systems, which support the manager in his decisions through models, methods and problem relevant data. DSS are especially used for badly structured situations, where it is hard to find a solution for a given problem. And DSS have a broad area of application: in all levels of management and all phases of the decision process. The results of this work can be integrated in an existing DSS. A manager of an inventory for example could use this system to calculate the inventory policy for the future periods. The manager can apply the calculation directly or he can use the system for analysis.

1.3 Combinatorial Optimisation

1.3.1 Basic Terms

Each day decision makers are confronted with problems of growing complexity. The problem to be solved is often expressed as an **optimisation problem**.

In principle an optimisation problem can be described as follows [DD04]: Maximise (or minimise) the function $\mathcal{H}(x)$ under the following restrictions

$$g_i(x) \quad \begin{cases} \leq 0 \\ = 0 \\ \geq 0 \end{cases} \quad \text{with} \quad \text{i=1,...,N} \quad \text{and} \quad x \in \Gamma \tag{1.1}$$

where x is a possible configuration in the configuration space Γ, $g_i(x)$ are the contraints and $\mathcal{H}(x)$ is the **objective** function, which shall be optimised. It is a map from the set of feasible solutions (configurations) x into the set of real numbers:

$$\begin{aligned} \mathcal{H} \quad : \quad & \Gamma \longmapsto \mathbb{R} \\ & x \longrightarrow \mathcal{H}(x) \end{aligned} \tag{1.2}$$

Regularly the total costs of a system shall be minimised. A maximisation problem can be changed into a minimisation problem by multiplication with -1. A **combinatorial** optimisation problem is defined just like a normal optimisation problem. $\mathcal{H}(x)$ is again the objective function which shall be optimised. But this time the configuration space Γ is finite and consists of discrete elements. A **continuous** optimisation problem has a configuration space which is not discrete.

The restrictions of combinatorial optimisation problems are difficult to handle. At first those configurations have to be banned, which do not fulfil the restrictions. Thereby the search space is divided into small islands, which the system cannot leave if it is stranded. Thus the optimum is reached just per chance. The second possibility is to accept unfulfilled restrictions and to use the so called **virtual costs** (penalties), if a restriction is not fulfilled. A penalty function $\mathcal{H}_P$ is a map

$$\begin{aligned} \mathcal{H}_P \quad : \quad & \Gamma \longmapsto \mathbb{R}_+ \\ & x \longrightarrow \mathcal{H}_P(x) \end{aligned} \tag{1.3}$$

with $x \in \Gamma$ and

$$\mathcal{H}_P(x) = \lambda \cdot g(x) \quad \begin{cases} = 0 & x \text{ fulfils the restriction} \\ > 0 & \text{else} \end{cases} \tag{1.4}$$

$\lambda \in \mathbb{R}$ is a parameter, which has to be fixed. For each restriction a function can be defined and integrated into the objective function. If λ is very high the restrictions

have to be fulfilled, because the objective function (which shall be minimised) has higher values. For $\lambda = 0$ no restrictions are considered. A solution is **valid**, if all restrictions are fulfilled. One can distinguish between **hard** and **weak** penalties. Hard penalties do not allow to break a restriction; weak penalties allow a small non-fulfilment. In route planning for example a truck can be slightly overloaded.

A **configuration** is a possible solution of a problem, which doesn't need to fulfil all restrictions. A configuration is an element of the configuration space, which is formed by all configurations. Because of many degrees of freedom the space is called *high dimensional*. The set contains elements which do not solve the problem, because they do not fulfil the restrictions. The **solution space** is the set of all valid combinations of the system parameters. Each element of the set solves the problem and fulfils the restrictions. The solution space is a subspace of the configuration space; its elements only differ in quality.

It is common to describe the step from x to $x' = A(x)$ as **move**. $A(x)$ is the operator which changes the current configuration and depends on the shape of the underlying configuration space. The number of all moves, starting from a solution x, is restricted; not every solution x' can be reached from x. The possible moves are characterised by M_x. In principle those sets can be chosen freely for a given problem; but $x' = A(x)$ should be valid for a $m \in M_x$. When for all x of the solution space Z the sets M_x are given, a concept of neighbourhood can be defined on the set Z. Thereby a problem P with the solution space Z shall be given:

- If M_x is the set of moves, which can be excuted on x in Z, then the **neighbourhood** of x can be defined like following:

$$N_M(x) := \{x' \in Z \mid \exists \quad m \in M_x : x' = A(x)\} \tag{1.5}$$

- The union of all neighbourhoods $N_M(x)$, $x \in Z$, is called **neighbourhood structure** $\mathcal{N}$.

- If $x' \in N(x) \Leftrightarrow x \in N(x')$ is valid, a **symmetric** neighbourhood structure is given .

- Let $x, y \in Z$. The sequence of solutions $x_1, \ldots, x_k$ is called **solution path** from x to y, if the following is valid:

$$x_1 \in N(x), y \in N(x_k) \quad \wedge \quad x_{i+1} \in N(x_i) \quad \forall \quad i = 1, \ldots, k-1 \tag{1.6}$$

- A neighbourhood structure $\mathcal{N}$ is called **coherent**, if there is a path from x to y for all $x, y \in Z$.

If the operator A always produces valid solutions, it generates a solution path starting with x_0. Then the operator should find the best x' from $N(x)$:

$$\mathcal{H}(x') = \min_{y \in N(x)} \mathcal{H}(y) \tag{1.7}$$

Depending on the neighbourhood structure, $N(x)$ can be very big; this means that the subproblem itself has a great computation time. In such cases, the minimum x' of a subset $\bar{N}(x) \subseteq N(x)$ can be taken as substitute. Basically for $\bar{N}(x) \geq 2$ the following subsidiary optimisation problem is to solve:

$$\min\{\mathcal{H}(y) | y \in \bar{N}(x) \subseteq N(x)\} \tag{1.8}$$

Therewith the operator $A(x)$ itself can be formulated as an algorithm: Produce a subset $\bar{N}(x)$ of neighbourhood solutions $N(x)$ and find a x' due to 1.7. Concerning the objective function f, x^0 is called local minimum in the solution space Z and the neighbourhood N, if

$$\mathcal{H}(x^0) \leq \mathcal{H}(x) \qquad \forall \quad x \in N(x^0) \tag{1.9}$$

With opposite sign, x^0 would be a local maximum; in both cases it is a local optimum. The position of the **local optimum** is not only characterised by the objective function and the solution space; the chosen concept of neighbourhood plays an important role as well.

With the concept of neighbourhood the idea of a **local** and **global** minimum (maximum) can be formulated. A solution $x_{min} \in \Gamma$ is a **global minimum**, if for all solutions x in the solution space Γ : $\mathcal{H}(x_{min}) \leq \mathcal{H}(x)$ holds. $x_{max} \in \Gamma$ is called *global maximum*, if for all solutions x in the solution space Γ: $\mathcal{H}(x_{max}) \geq \mathcal{H}(x)$ holds. A solution $x \in \Gamma$ is a *local minimum*, if : $\mathcal{H}(x_{min}) \leq \mathcal{H}(x') \qquad \forall \quad x' \in N$. $x_{max} \in \Gamma$ is called *local maximum*, if: $\mathcal{H}(x_{max}) \geq \mathcal{H}(x') \qquad \forall \quad x' \in N$.

The structure of the configuration space is independent of the neighbourhood structure. If the different configurations are defined by the neighbourhood structure $\mathcal{N}$, the so called **search space** $\mathcal{D}$ is given. During optimisation one "walks" through the search space step by step. The more moves there are in $\mathcal{D}$, the more paths exist between two points of the search space and thus it is easier to leave local optima on the way to the global optimum.

During optimisation a "walk" from one point of the search space to another is made. If each point of the phase space is assigned to the equivalent energy $\mathcal{H}(x)$, one gets the so called **hill-valley-landscape** [Mo87] as illustration of the **energy landscape**. In Figure 1.4 there are just two dimensions of the normally high dimensional phase space represented. For a small number of different moves it is easy to see, that mostly just local minima are found and not the global optimum. A great number of moves makes it possible to bypass an energy barrier; the system doesn't get stuck in a local minimum.

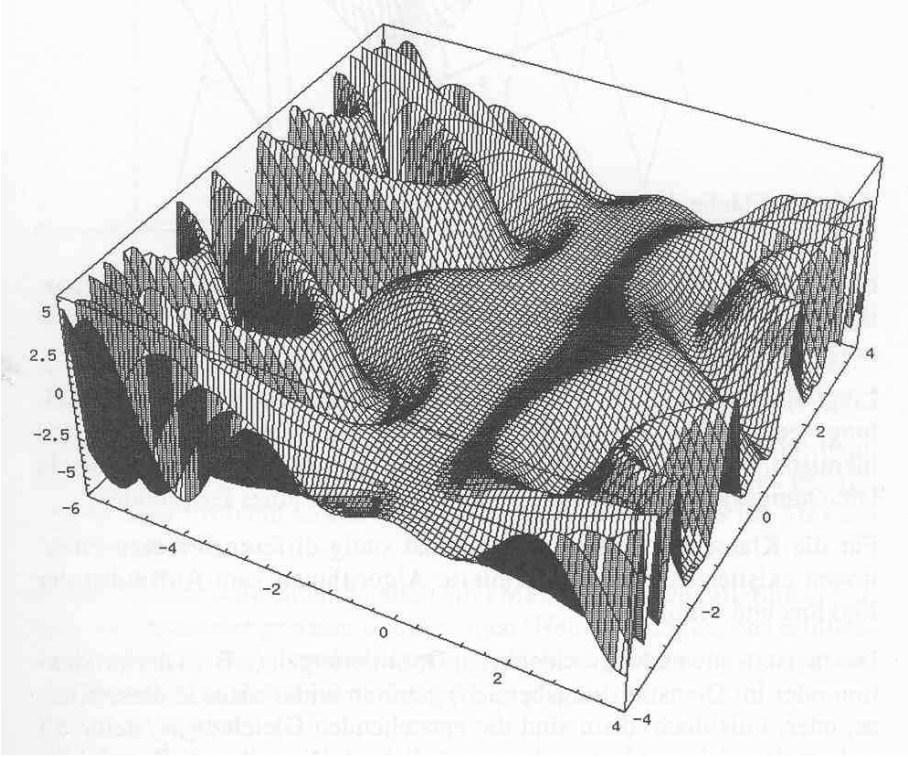

Figure 1.4: Energy landscape

1.3.2 Complexity

An important idea of OR is the **complexity** of optimisation problems. The complexity depends on the chosen methods to solve the problem; thus the concept of "algorithm" and "problem" has to be defined.

A **problem** P consists of an infinite number of problem specifications $p \in P$ with the same structure. In general, the set of all values, which defines the concrete specification of a problem, is called **input**; the concrete specification with numerical values is an **instance** of the problem. A method which is able to solve each problem specification is an **algorithm**. The best algorithm would be an efficient one. The efficiency evaluation of an algorithm depends on the ressources a program uses to execute the algorithm. A **program** is a concrete scheme of calculation steps, which is necessary for the implementation on a computer. In this context the computing time of such a program plays an important role; it depends on many variables and is therefore difficult to determine exactly. Because of that the basic computation operations are counted: arithmetic operations, comparisons and saving operations are assumed to be elementary computation steps. For simplification all those steps shall have the same duration. But there is

no sense in calculating the number of necessary computation steps for an instance; moreover it is interesting to measure the necessary computation time for solving any problem specification.

When the input of a specification is described as a sequence of symbols, the length of those sequence determines the **input size**. The value depends on the type of codification; therefore it is enough to know the dimension of a specification p. The dimension can be called $|p|$. The input size of a TSP specification with n locations for example is $|p| = n$.

If $r_A(p)$ is the minimum number of necessary computation operations to execute the program of an algorithm A, the maximum number of operations for a problem specification of the size n is given by:

$$\sup_{|p|=n} \{r_A(p); p \in P\} \tag{1.10}$$

In mathematics, the **supremum** or least upper bound of a set S of real numbers is denoted by $\sup_S$ and is defined to be the smallest real number that is greater than or equal to every number in S. It is enough to estimate the order of the upper bound of this expression. Thus some mathematical concepts have to be introduced at first in Table 1.1.

$g(n)$ is any, non-negative function over the definition space $\mathbb{N}$: $g : \mathbb{N} \longrightarrow \mathbb{R}$
1. Another non-negative function $f(n)$ is of the **order** of $g(n)$, if there is a $c \in \mathbb{R}$ and $n_0 \in \mathbb{N}$, so that $f(n) \leq c \cdot g(n) \quad \forall \quad n \geq n_0$
2. The number of all functions with the order $g(n)$ is called $\mathcal{O}(g(n))$; $\mathcal{O}$ is the **Landau** or **complexity function**.
3. Instead of $f(n) \in \mathcal{O}(g(n))$ it can be written $f(n) = \mathcal{O}(g(n))$.

Table 1.1: Definition 1

This definition means that the function $f(n)$ is bounded by $g(n)$, for n sufficiently large. So the function $f(n)$ is of the order $g(n)$, if the following is valid:

$$\exists \quad c \in \mathbb{R} \qquad \lim_{n \to \infty} \frac{f(n)}{g(n)} = c. \tag{1.11}$$

With this arrangements the measure of necessary computation operations for the solution of a problem specification with input size n can be defined:

1. Let $r_A(p)$ be the number of necessary computation operations of an algorithm A to solve $p \in P$. Function $R_A(n)$ with

   $$\sup_{|p|=n}\{r_A(p); \quad p \in P\} \in \mathcal{O}(R_A(n)) \quad \forall \quad n$$

   is called complexity of an algorithm A. It gives an upper estimation for the maximum number of computation steps of an algorithm A for a problem specification with input length n.

2. If $R_A(n)$ is bounded by a polynom, the algorithm is called **polynomial**; otherwise it is called **non-polynomial**.

3. For two algorithms A and B with the complexities $R_A(n)$ and $R_B(n)$ A is more **efficient** than B, if following is valid:

   $$R_A(n) \in \mathcal{O}(R_B(n)) \quad \wedge \quad R_B(n) \notin \mathcal{O}(R_A(n)) \quad \forall \quad n$$

Table 1.2: Definition 2

More precisely $R_A(n)$ is called *maximum computation time*; the **worst case analysis** is orientated at this time measure. The disadvantage of this standard method is lacking representativeness with respect to practical problems. Therefore the **average case analysis** has gained significance lately. But in order to find the average effort of a problem, the probability distribution of all possible problem specifications has to be known. If just a finite number of exemplary problems is taken, the representativeness of this random sample has to be guaranteed.

Those results can be directly transferred to problems. The complexity of the most efficient known algorithm to solve a problem defines the problem complexity in a *weak* sense. The difficulty of this definition is easy to see: the validity of a statement on complexity depends on the number of all *known* algorithms for a special problem and is thus of temporary character. That is interesting for the practitioner, but in theory this measure is just an upper bound for the complexity of the problem. But when there is evidence that no algorithm is more efficient than the known, one speaks of problem complexity in a *strength* sense.

The number of problems, which can be solved in polynomial time, has a special position. If there is a deterministic polynomial algorithm for the solution of a problem, it is called **polynomial limited**. The number of all polynomial limited problems is characterised by $\mathcal{P}$. A problem $P' \notin \mathcal{P}$ is named non-polynomial limited. In informatics a distinction is drawn between deterministic and non-deterministic problems. $\mathcal{NP}$ is the set of all problems, which can be solved with non-deterministic algorithms in polynomial time. An algorithm is non-deterministic, when there is no certainty about the next step. Each problem of $\mathcal{P}$ is obviously an element of $\mathcal{NP}$; but not vice versa. It is uncertain whether the formalism of $\mathcal{NP}$ is necessary, because nobody could *prove* a problem to be element of $\mathcal{NP}$ and not of $\mathcal{P}$. If there would be a proof for $\mathcal{P} \neq \mathcal{NP}$, the search for an efficient solution could be dismissed.

If a problem p is such that every problem in $\mathcal{NP}$ is polynomially transformable to p, it is **$\mathcal{NP}$-hard**. If in addition problem p itself belongs to $\mathcal{NP}$, p is said to be **$\mathcal{NP}$-complete**. The concept of *transformability* means following: Suppose there is a problem p_1 which can be solved by an algorithm A. If every instance of another problem p_2 can be transformed into an instance of p_1 in polynomial time, then algorithm A can be used to solve p_2. $\mathcal{NP}$-complete problems are the "hardest" of all problems in $\mathcal{NP}$. If a polynomial algorithm for any $\mathcal{NP}$-complete problem would have been found, a polynomial algorithm for all problems of $\mathcal{NP}$ would be available and $\mathcal{P} = \mathcal{NP}$ would be proved.

But all attempts to prove $\mathcal{P} = \mathcal{NP}$ theoretically have failed so far. And because no exact polynomial algorithm has been found for any problem in $\mathcal{NP}$, there is strong circumstantial evidence that $\mathcal{P} \neq \mathcal{NP}$. Therefore the use of heuristics has considerable justification.

Besides complexity there is another argument for favouring heuristics [Re95]: the best solution of an optimisation *model* is not automatically the best solution for the underlying real-world problem. Of course there is never a truly exact model, but heuristics are usually more flexible and capable of coping with more complicated (realistic) objective functions and constraints than exact algorithms.

1.3.3 Multi-Objective Optimisation

Most problems in reality don't have a single objective. Normally, multiple objectives have to be considered for an adequate solution of the complete problem. **Multiobjective** (or multicriteria) optimisation is the process of optimising several conflicting objectives with different constraints at the same time. Multiobjective optimisation problems can be found wherever optimal solutions are demanded in the presence of trade-offs between conflicting objectives. In inventory control for example there is a trade-off between storage and order costs: the lower the order costs (few orders with high quantity), the higher the storage costs

(high stock due to a high order quantity). Usually there is no single solution to multiobjective problems; instead there are many different alternative solutions. This diversity eliminates simple decisions; the decision has to be based upon the complex context of the situation. In mathematical terms, the multiobjective problem can be written as:

$$\begin{aligned} \max_x \quad \mathcal{H}(x) &= (\mathcal{H}_1(x), \ldots, \mathcal{H}_N(x))^T \\ \text{with} & \\ f(x) &\geq 0 \\ g(x) &= 0 \\ x_u \geq \quad x &\geq x_l \end{aligned} \tag{1.12}$$

where $\mathcal{H}_i$ is the i-th objective function, f and g are the (in-)equality constraints; x is the vector of optimisation variables, which is restricted by x_u as the upper bound and x_l as the lower one. The solution of this problem is a set of so called **pareto** points. Pareto solutions are those for which improvement in one objective is only possible with the worsening of at least another objective. The solution to a multiobjective problem is a (possibly infinite) set of pareto points. A solution $\mathcal{H}^*$ is termed pareto-optimal, if there is no other feasible solution $\vec{Z}$ such that ${\mathcal{H}_i}^* \leq \mathcal{H}_i$ for all $i \in \{1, \ldots, n\}$ and ${\mathcal{H}_i}^* < \mathcal{H}_i$ for at least one $j \in \{1, \ldots, n\}$.

In traditional multiobjective optimisation the different objectives are aggregated to a single (scalar) function, which can be treated by techniques like genetic algorithms, random walk, simulated annealing, etc. Mostly heuristics are used for optimisation, because often at least one objective is of combinatorial nature and thus linear methods like **multiple objective linear programming** (MOLP) can only be implemented in special cases. In this dissertation the traditional way is chosen and in the majority of cases the optimisation is done with simulated annealing.

1.4 (Meta-)Heuristics

A naive approach for solving an instance of a combinatorial optimisation problem is simply to list all possible solutions, evaluate their objective functions and pick the best. It is immediatly obvious that this approach of **complete enumeration** is likely to be inefficient, because of the vast number of solutions to any problem of reasonable size. This point can be easily illustrated for the TSP. If a computer can list all solutions of a 20 city problem in 1 hour, it will need 17.5 hours for 21 cities and 6 centuries for 25. The reason for this increase of computation time lies in the exponential increase of possible solutions: $(N-1)!$, where N is the number of cities. In the early days of operations research, the emphasis

was mostly on finding the optimal solution to a problem. Therefore various exact algorithms were devised which would find the optimal solution much more efficiently than complete enumeration. The most famous example is the simplex algorithm for linear programming problems. At first such algorithms were capable of solving small instances of a problem, but not able to find optimal solutions to larger instances of a problem in a reasonable amount of computation time. As computing power increased, it became possible to solve larger problems; the researchers became interested in how the solution times varied with the size of the problem. In some cases the computing effort could be shown to grow as a low-order polynomial in the size of the problem.

Some combinatorial problems can be solved with **linear programming (LP)** by introducing integer variables taking the values 0 or 1 in order to produce an **integer programming (IP)** formulation. Exact methods like branch & bound or dynamic programming find an optimal solution in a finite number of steps. But that does not mean that a practical problem can be solved in acceptable computation time. The computation effort for $\mathcal{NP}$ problems rises strongly with the input size. In spite of the fast development of hardware technology, realistic problems of this class cannot be solved exactly.

Algorithms, which find a good solution in relatively short computation time, are called **heuristics** (heureka [greek]= i have found). The problem here is that there is no guarantee of optimality; in many cases it is not clear how close a particular solution is to optimality. In some cases it is possible to analyse heuristic procedures explicitely and find theoretical results bearing on their average or worst-case performance. However, analysis of general performance in this way is often difficult, and in any case may provide little help in evaluating the performance of a heuristic in a particular instance. Some heuristics try to find a valid start solution for an optimisation problem P:

$$\min \mathcal{H}(x) \quad \text{with} \quad x \in Z \tag{1.13}$$

with $\mathcal{H}$ as *objective function* and Z as acceptance area. This area is often not given explicitly, but implicitly by restrictions; therefore the determination of any element of Z is not trivial. In the following it shall be assumed that the minimum is positive and exists in Z. An example is the **next-neighbour** heuristic, which is used in route planning. In this heuristic the neighbour with the smallest distance is visited next. Another proceeding has the **Vogel approximation method**, which is used in the area of transport optimisation. The basic idea is to move those transport quantities with the lowest unity costs at first and to pay attention to the fact that alternative transports from the same supplier or to the same customer would be much more expensive. While the next-neighbour heuristic finds a valid solution more or less independently from the objective function,

the Vogel approximation method makes more effort to use the objective function when searching for a start configuration. This qualitative difference is easy to see and can be quantified by the **performance** of a heuristic. Here, the solution of a heuristic is connected with the optimal solution. For a given minimisation problem P with an objective function $\mathcal{H}$ the performance $Per_H(n)$ of a heuristic H with the instance size n is the lowest number with:

$$Per_H(n) \geq \frac{\mathcal{H}(x_H(p))}{\mathcal{H}(x^*(p))}, \qquad \forall \quad p \in P \quad \text{with} \quad |p| = n \tag{1.14}$$

$x_H(p)$ is the solution found by heuristic H and $x^*(p)$ is the optimal solution of a specification p from P. Then the performance of the heuristic H for the problem P is defined by

$$Per_H = \lim_{n \to \infty} Per_H(n) \tag{1.15}$$

In many cases the discovery of a good start configuration for a given problem contains some difficulties in relation to the performance and the computation complexity. A way out offer other heuristics, which improve the start configuration step by step. If there is a known start configuration $x \in Z$, the **operator** $A(x)$ generates a sequence of valid solutions, whose objective value is continuously reduced in every iteration. If there is no improvement possible, the method stops. The sequence of solutions only depends on the operator $A(x)$. This operator should produce a better solution than x; if that cannot be realised, the solution is excellent in the solution space; one speaks of a **local optimium**. The definition of a local optimum of a function $\mathbb{R}^n \longrightarrow \mathbb{R}$ is strongly connected with the concept of neighbourhood (see subsection 1.3.1). Heuristics can be classified into several broad categories: greedy construction methods, neighbourhood search routines, relaxation techniques, partial enumeration and so on. But many heuristics are problem-specific; therefore a method which works for one problem may not be appropriate to solve a different one. Furthermore a "classical" heuristic mostly gets trapped in a local minimum. In order to improve the effectiveness of the method, it can be applied several times with different initial conditions; at the end the best result is chosen. But this increases the computation time without any guarantee to obtain the optimal configuration, especially when the number of local minima grows exponentially with the size of the problem. To overcome the obstacle of local minima, a temporary degradation seems promising. A mechanism for controlling the degradations makes it possible to avoid the divergence of the process; a local minimum can be left and other valleys are explored concerning their optimality. Therefore techniques like **metaheuristics** are preferable, because they can leave local minima and are applicable far more generally. The most famous metaheuristics are:

- simulated annealing (SA)

- genetic algorithms (GA)
- tabu search (TS)
- ant colony algorithms (ACA)

Each of those is actually a family of methods. Examples for less widespread metaheuristics are: noising method, distributed search, Alienor method, particle swarm optimisation, artificial immune systems, etc. The metaheuristics can be applied to all kinds of discrete problems and can also be adapted to continuous problems. Some features appear in most metaheuristics, for example *diversification* to explore regions of the search space and *intensification* to go into some promising regions; another common feature is the use of memory to archive the best solutions. And to some extent they can deal with the stochastic explosion of possibilities. But metaheuristics also share some disadvantages: difficulties in tuning numerous parameters and long computation times.

In the current state of research it is generally impossible to envisage the effectiveness of a given method for a special problem. Moreover, the current tendency is the emergence of so called **hybrid methods**, which benefit from the specific advantages of each metaheuristic by combining them in a new method. Finally a basic advantage is their use for all kinds of extensions:

- **multiobjective** optimisation: several contradictory objectives are optimised simultaneously
- **multimodal** optimisation: a whole set of local optima is determined
- **dynamic** optimisation: the objective function is temporarily varied

A classification of mono-objective optimisation methods is given in Figure 1.5. Thereby combinatorial and continuous optimisations are differentiated. For combinatorial optimisation several methods can be used: "specialised" heuristics, entirely dedicated to the considered problem and metaheuristics. For continuous optimisation the linear case (which is solved with linear programming) is separated from the non-linear one, where the framework for difficult optimisation can be found. Thus a pragmatic solution can be to resort to the repeated application of a local method; those methods mostly exploit the gradients of the objective function. If the number of local minima is very high, the recourse to a global method is essential. The traditional methods of global optimisation require restrictive mathematical properties of the objective function and thus metaheuristics are a better alternative. There are metaheuristics "of neighbourhood", which make progress by considering only one solution at a time (SA, TS, etc.) and "distributed" ones, which handle a complete population of solutions (GA and others).

In the presence of a concrete optimisation problem it is difficult to choose an "efficient" method able to produce an "optimal" solution at the cost of a "reasonable" computation time. So far theory is not of great help, because the convergence theorems are often non-existent or just applicable under very restrictive assumptions. Moreover the theoretically optimal adjustment of the various parameters is often inapplicable in practice, because it induces a prohibitive computing cost. Consequently the choice of a "good" method and the adjustment of the parameters depends on the know-how and the experience of the user.

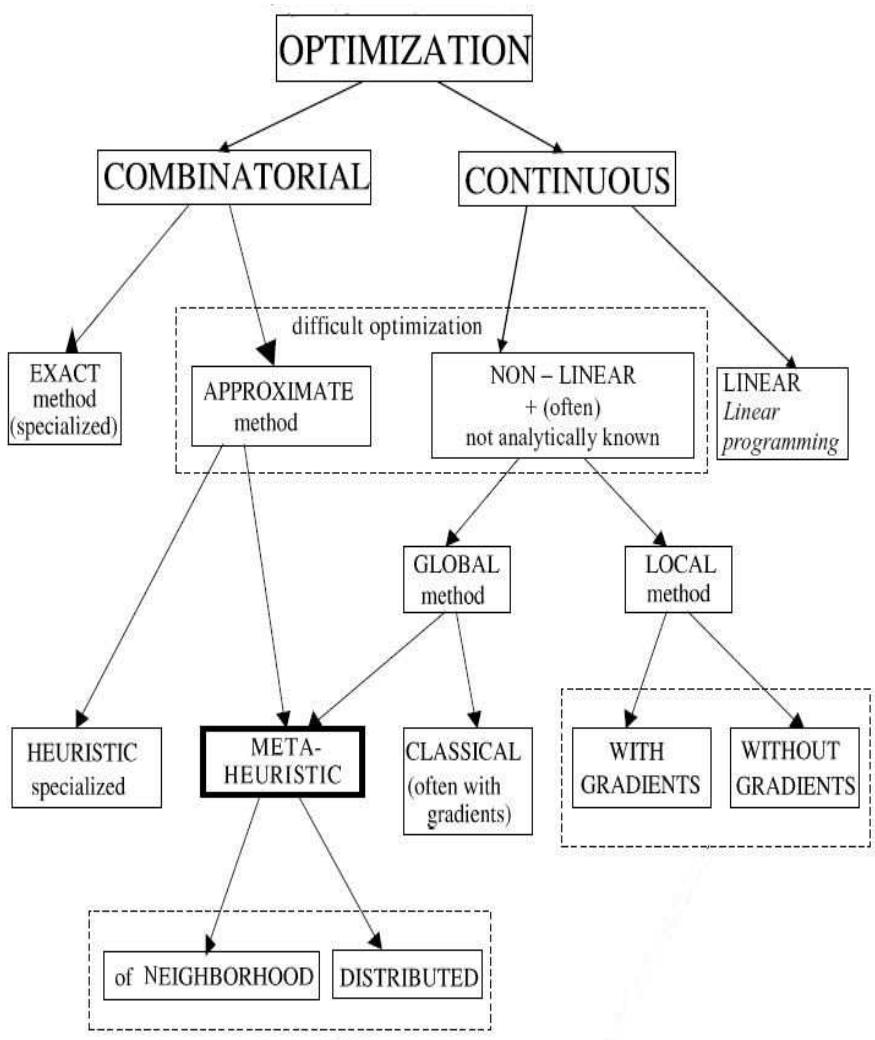

Figure 1.5: Classification of mono-objective optimisation methods [CS03]

1.5 Standard Methods and Problems of OR

1.5.1 Simplex Algorithm

The simplex algorithm is a method of mathematical optimisation; it was developed in 1947 by Dantzig [Co85]. This algorithm solves a problem exactly after a finite number of steps or identifies its insolubility. In some theoretical exceptions there can be cycles, which prevent the finding of the optimal solution. The name is derived from the fact, that the equations describe a simplex, whose edge is used to find the solution. Methods of **linear optimisation** or linear programming are the most important tools of OR. The optimisation of a linear function occurs in many economical problems, for example in production planning. Therefore the mathematical model can have a lot of different forms: the objective function has to be maximised or minimised, the restrictions are (in-) equations. In order to have a unified solution method, it makes sense to develop a standard form, into which all linear optimisation problems can be transformed. This idea leads to the **standard equation** form:

$$\begin{aligned} & \max \quad \mathcal{H} = c^T x + b_0 \\ \text{with} & \\ & Ax = b \\ x \geq 0, \qquad & x, c \in \mathbb{R}^n, b \in \mathbb{R}^m \end{aligned} \tag{1.16}$$

It is assumed that $\mathbf{A}$ is a $m \times n$ matrix with $m < n$ and $\text{rank}(\mathbf{A}) = m$. The main advantage is the standardisation; another one is that the objective function $\mathcal{H}$ can be handled like a restriction. When the problem is transformed into the standard form, two aspects have to be considered:

1. Minimisation problems are transformed in maximisation problems by multiplication with -1.

2. Inequations are transformed to equations by the introduction of so called **slack variables**.

The main task of linear optimisation is to find the optimal solution. The first problem is that the set of all possible solutions consists of an infinite number of points; it can even be unlimited. So one point has to be selected from the infinite set. The decisive idea is to restrict the possible solutions to the so called geometrical edges of the solution set. When the linear optimisation problem (LOP) has an optimal solution, it is at least in one of the edges. So just the edges (basic solutions) have to be checked in order to find the optimum. Starting from

one edge, an adjacent edge can be located in order to get a better solution; this is continued until the optimum edge is reached. According to the common notation a so called pivot format to Equation 1.16 looks like following:

$\mathcal{H}$	$x_1 \cdots x_m$	$x_{m+1} \cdots x_n$	x_B
1	$0 \cdots 0$	$y_{0,m+1} \cdots y_{0,n}$	y_{00}
0	1	$y_{1,m+1} \cdots y_{1,n}$	y_{10}
$\vdots$	$\ddots$	$\vdots \quad \vdots$	$\vdots$
0	1	$y_{m,m+1} \cdots y_{m,n}$	y_{m0}

Table 1.3: Pivot Format

There $x_{m+1}, \ldots, x_n$ are the slack variables, which are necessary to transform the inequations into equations; y_{00} is the value of the objective function, which is equivalent to b_0. The basic solution is:

$$x = (x_1, \ldots, x_m, \ldots, x_n)^T = (y_{10}, \ldots, y_{m0}, 0, \ldots, 0)^T \tag{1.17}$$

If the **criterion line** of the pivot format is not negative ($y_{0,m+1}, \ldots, y_{0,n} \geq 0$), the solution is optimal. The most often used method to solve LOPs is the simplex method. It is built on the **Gauss - Jordan** algorithm, which is used to solve linear equation problems. The complete algorithm is described in Table 1.4

1.5.2 Branch & Bound - BB

Branch and bound (BB) is a general algorithmic method for finding optimal solutions of various optimisation problems, especially in discrete and combinatorial optimisation. It is basically an enumeration approach in a fashion that prunes the nonpromising search space. The method was first proposed by A. H. Land and A. G. Doig in 1960 for linear programming. The general idea may be described in terms of finding the minimal or maximal value of a function $\mathcal{H}(x)$ over a set of admissible values of the argument x. Let $P(Z^0)$ describe the following combinatorial optimisation problem:

$$\min \mathcal{H}(x) \qquad x \in Z^0, \quad Z^0 \quad \textit{finite}. \tag{1.18}$$

The optimal solution of the problem is $x^*(Z^0)$ and the optimum value of the objective function is $\mathcal{H}(x^*(Z^0))$. In principle this problem can be solved by calculation of all permitted solutions. The optimum can be found by comparison: at first $\mathcal{H}(x)$ is calculated for all $x \in Z^0$. Then x^* is the optimal solution, if $\mathcal{H}(x^*) \leq \mathcal{H}(x)$ for all $x \in Z^0$.

S1 Test of optimality
Is $y_{0j} < 0$ for a j (x_j not free) or $y_{0j} \neq 0$ for a $j(x_j$ free) then go to **S2**,
else: STOP! Optimality.

S2 Select a column $j_0 \in \{1, \ldots, n\}$ with
$y_{0,j_0} = \min\{y_{0,j} | j \in \{1, \ldots, n\}\} < 0$
Go to **S3**.

S3 Is there a $i_0 \in \{1, \ldots, m\}$ with $y_{i_0,j_0} > 0$?
If not, the objective function has no upper bound.
STOP.
Otherwise go to **S4**.

S4 Select a row i_0 with $y_{i_0,j_0} > 0$ and

$$\frac{y_{i_0,0}}{y_{i_0,j_0}} = \min\{\frac{y_{i,0}}{y_{i,j_0}} | y_{i,j_0} > 0\}.$$

Make a Pivot step with y_{i_0,j_0}.
Go to **S1**.

Table 1.4: Simplex - Algorithm

A **complete enumeration** of all permitted solutions is only possible for problems with a very small set Z^0. Therefore it is better to divide the solution set in smaller parts and to prove for some that they do not contain the optimum. In this method not every solution has to be considered explicitly; therefore this methods are characterised as **implicit enumeration**. A famous representant of those methods is **branch & bound**: instead of a complete problem $P(Z^0)$ a **relaxed** problem $P(Z)$ with a bigger set $Z \supseteq Z^0$ is examined. That makes sense, because the new problem $P(Z)$ is easier to solve, if Z is well selected. If $x^*(Z) \in Z^0$ is valid for the solution of the relaxed problem $P(Z)$, the optimum solution of the original problem $P(Z^0)$ has been found. In the other case $\mathcal{H}(x^*(Z))$ is a *lower bound* for the value of the objective function belonging to $x(Z^0)$; that is true because of $Z \supseteq Z^0$. The main component of branch & bound is the **branching** of a solution and the **bounding** by calculating bounds. Branching means the splitting of the problem $P(Z)$ into several subproblems $P(Z_i)$ by splitting Z into

several subsets Z_i with $\bigcup_i Z_i = Z$. Because of $Z_i \subseteq Z$ for the subproblems $P(Z_i)$ is valid: $\mathcal{H}^*(Z_i) \geq \mathcal{H}(Z) \quad \forall \quad i$. If the optimum solution $x^*(Z_i)$ of a subproblem $P(Z_i)$ is allowed for the problem $P(Z^0)$, then $x^*(Z_i)$ is also the optimum solution for the problem $P(Z^0 \cap Z_i)$. Because of $(Z^0 \cap Z_i) \subseteq Z^0$ follows $\mathcal{H}^*(Z^0 \cap Z_i) \geq \mathcal{H}^*(Z^0)$; so $\mathcal{H}^*(Z^0 \cap Z_i)$ is an upper bound for $\mathcal{H}^*(Z^0)$. If the branching is continued with all problems $P(Z_i)$, one gets a tree of problems with $P(Z)$ as root.

	Let $P(Z)$ be the relaxed problem
	$\mathcal{F} := \infty$
	$\mathcal{Q} := \{P(Z)\}$
(IT_{BB})	While $\mathcal{Q} \neq \emptyset$
	Take an element $P \in \mathcal{Q}$
	Solve P
	If $\mathcal{H} < \mathcal{F}$
	if x allowed
	$x^* := x$
	$\mathcal{F} := \mathcal{H}$
	else
	Generate subproblems P_i
	$\mathcal{Q} := \mathcal{Q} \cup \{P_i\}$
	Go to (IT_{BB})
x^*	is the optimal solution with the objective value $\mathcal{F}$

Table 1.5: Branch & Bound

Bounding means the blocking of a subproblem $P(Z_i)$ for further branching, because a branch is only useful, if the optimum solution $x^*(Z^0)$ can be in Z_i. If $\mathcal{F}$ is the smallest upper barrier found so far, several conditions can be drawn:

- Is $\mathcal{H}^*(Z_i) \geq \mathcal{F}$, a further branching does not lead to a better result; $P(Z_i)$ is not considered further.
- If $\mathcal{H}^*(Z_i) < \mathcal{F}$, Z_i is branched.

The method stops, if there is no problem left to be split; the solution with the value $\mathcal{F}$ is the optimum of $P(Z^0)$. The formalisation of BB is shown in Table 1.5. This approach is used for a number of NP-hard problems, such as: knapsack problem, integer programming, nonlinear programming, traveling salesman problem (TSP).

1.5.3 Traveling Salesman Problem - TSP

This concept summarises everything in literature what is connected with optimising the way of persons or transport vehicles. Postmen, traveling salesman, garbage/supply trucks search for the best tour. The special problem of the traveling salesman is to visit n-1 customers starting from a special point and going back to this point at the end. Searched for is the shortest time or the lowest cost-expensive tour. An exact definition is given in Table 1.6:

$D = (V, E; d)$ shall be an evaluated and directed graph with the vertex set V ($|V| = n$), the edge set $E = V \times V$ and the evaluation $d : E \to [0, \infty)$.

1. $(\nu_1, \ldots, \nu_l)$ is called a **tour** including the places $\nu_1, \ldots, \nu_l$,
   if following is valid: $\nu_i \in V$ $(1 \leq i \leq l \leq n+1)$
   and $\nu_i \neq \nu_j$ for $i \neq j$, $\quad 1 \leq i, j \leq l-1$

2. A tour is called $(\nu_1, \ldots, \nu_l)$
   cdot **open**, if $\nu_1 \neq \nu_l$
   cdot **closed**, if $\nu_1 = \nu_l$
   cdot **complete**, if every place of V is included in $(\nu_1, \ldots, \nu_l)$
   cdot **component** tour, if $(\nu_1, \ldots, \nu_l)$ does not contain every place of V
   cdot **round trip**, if it is closed and complete.

3. The **length** of a tour $(\nu_1, \ldots, \nu_l)$ is defined by $\sum_{i=1}^{l-1} d(\nu_i, \nu_{i+1})$.

4. The problem to determine a tour of minimum length over V is characterised as taveling salesman problem (TSP). If $d(\nu_i, \nu_j) = d(\nu_j, \nu_i)$
   for all $1 \leq i, j \leq n$, the TSP is **symmetric**.

5. If $d(\nu_i, \nu_j) + d(\nu_j, \nu_k) \geq d(\nu_i, \nu_k)$ is valid for all $1 \leq i, j, k \leq n$,
   the TSP is called **geometric**.

Table 1.6: Definition of a TSP

1.5.4 Different Problems

Minimum Flow Problem

Many goods are moved in diverse **transport systems** either within a company or on their way from the producer to the customer. Pipe systems for gases and fluids, rail and road systems for all kind of items are examples for such transport means. They are characterised by **locations** and **paths**: locations where items are produced, which flow into the system, are traded or leave the system; and paths, where the real transport takes place. In the last decade many models have been developed, which reflect the real facts in a mathematical path and help to find out, at what time, where and how many quantities have to be transported. Transport systems are mostly described by **graphs**. Graphs consist of **vertices** and **edges** (arrows). The vertices mean locations and the edges are path connections with or without one-way character. The set of locations V can be devided into three disjunct subsets V_1, V_2, V_3:

- V_1 is the set of such locations, where the items flow into the system; the locations are called **sources**.
- V_2 are pure **locations of turnover** with input = output.
- V_3 characterises the set of locations, where the transported items are taken out of the system; they are called **sinks**.

Further it is assumed that one source $i \in V_1$ can push a_i item units (or less) per time unit into the system; from location $i \in V_3$ at least b_i item units shall be taken out. x_{ij} is the **flow** from i to j; this flow has a capacity of κ_{ij} [item unit/time unit]. And each transport from i to j shall cause $c_{ij} > 0$ cash units of transport costs per item unit. The task to find the flow with the minimum costs can be formulated as linear optimisation problem:

$$\min \mathcal{H} = \sum_{i \in V} \sum_{j \in N(i)} c_{ij} x_{ij} \tag{1.19}$$

$N(i)$ is the set of all adjacent vertices of i. The restrictions are

$$\sum_{j \in N(i)} x_{ij} - \sum_{l \in N(i)} x_{li} \quad \begin{cases} \geq a_i & \forall \quad i \in V_1 \\ = 0 & \forall \quad i \in V_2 \\ \leq -b_i & \forall \quad i \in V_3 \end{cases}$$

Formula 1.19 is called **minimum flow problem**. If there are no upper bounds $x_{ij} \leq \kappa_{ij}$ one speaks of a **transshipment problem**. In literature many special cases of 1.19 are discussed; their origin lies in the restriction of some model

variables. The minimum flow problem without turnover locations and without connections between sources and sinks is called the **(capacitated) transport problem**, if there are (no) upper bounds $x_{ij} \leq \kappa_{ij}$.

Assignment Problem

A slightly varied form is the **assignment problem**:

$$\max \quad \mathcal{H} = \sum_{i \in V_1} \sum_{j \in N(i)} c_{ij} x_{ij} \tag{1.20}$$

under the restrictions

$$\begin{aligned} \sum_{j \in N(i)} x_{ij} &\leq 1 \quad \forall \quad i \in V_1 \\ \sum_{i \in N(j)} x_{ij} &\geq 1 \quad \forall \quad i \in V_3 \end{aligned} \tag{1.21}$$

with $x_{ij} = 0, 1$. Here $A = |V_1|$ applicants have to be assigned to $B = |V_3|$ jobs; c_{ij} is the aptitude of applicant i for job j.

Knapsack Problem

In the so called **knapsack problem** a set of N items is available to be packed into a knapsack with a capacity of C units. Item i has value v_i and uses up c_i units of capacity. Now the optimisation problem is to determine the subset I of items which should be packed in order to maximise

$$\max \quad \mathcal{H} \;=\; \sum_{i \in I} v_i \tag{1.22}$$

with

$$\sum_{i \in I} c_i \;\leq\; C$$

The solution is represented by the subset $I \subseteq \{1, \ldots, n\}$.

1.6 Simulation as Method of Optimisation

In simulation sections of reality are modelled as closely as possible and analysed due to their attributes, if relations of the real environment cannot be discovered.

Simulation deals with computational experiments in order to describe the behaviour of systems. An experiment is a repeatable observation under controlled restrictions, whereas several independent variables are manipulated; thus the underlying hypothesis can be tested in different situations. For realising the experiment mathematical methods are used in order to observe the dynamic behaviour of the system. The observed section of reality is mapped to a simulation model. Then the variables of the model get values and results can be shown. By a change of input the consequences for the results can be illustrated and thus the dependencies between input and output variables are identified.

In most models only a few variables are considered. For practical application those input combination is chosen that promises the best value. A result is optimal, if a change of variables does not lead to an improvement. If the variables are stochastic it is difficult to say that a result is better than another. Further difficulties are:

- The effort for constructing a simulation model is high; in order to have an acceptable model size, a special adjustment for each problem is necessary.
- In spite of a great effort for the adjustment only approximative results are possible.
- Some unimportant details can be overestimated in the model.

The solution of decision tasks with existing techniques needs a reduction of complexity. Such an advancement is not strongly necessary in simulation, but therefore the optimum is mostly not reached. Thus it is essential to generate better simulation results by a specific change of variables. At first the variables of the objective function and their interdependences have to be shown. Simulation models can show the influence of single variables. Thus it is possible ...

- ... to analyse complex relations within a system.
- ... to see the effect of different environments onto the system.
- ... that the decision maker gets a better understanding of the system.
- ... to test decisions before implementation.

After simulation the complexity of the system can be reduced, then the system is optimised anew with methods of mathematical programming. It is also possible

to find criteria of good solutions; these criteria are discussed in the area of so called *heuristic programming*.

In management decision theory there are two requirements of a decision problem: *completeness* of decision variables and an *exact formulation* of the objective for a quantitative implementation. A bad fomulation of objectives in simulation would lead to a wrong analysis of the simulation results. Such a proceeding could generate masses of unnecessary data. The precise objective formulation is especially necessary for simulation, because exact knowledge of the system. This is eminently important for the simulation of stochastic processes, which are characterised by uncertainty and lack of information; here the so called **Monte-Carlo-simulation** can be a great help for understanding complex processes.

If simulation analysis is a part of decision making, every subtask has to be connected with the main problem. The results from simulation analysis can provide essential criteria for an optimisation method; the quality of a solution has to be seen in relation with the achievement to the specific problem. The classical proceeding of a simulation is:

1. Formulation of the problem
2. Formulation of the mathematical model
3. Formulation of the computational program
4. Data collection
5. Implementation of the simulation with real data
6. Analysis of the results

The simulation process can be described formally by a mathematical function $\mathcal{H} = \mathcal{H}(\vec{x})$, where $\vec{x}$ is the input vector and $\mathcal{H}$ the output vector. Commonly one of the components of $\mathcal{H}$ is the objective function and the other components have to fulfil a list of restrictions. Because of stochastic elements and the combination of continuous and discrete input variables, $\mathcal{H}(\vec{x})$ cannot be determined exactly. This is true especially for stochastic problems, where several probability distributions are used. Thus there can be no statement about steadiness, differentiability or anything else. Because of that the simulation of optimisation problems has to abstain from special methods; only a functional coherence between variables and object function is demanded.

Chapter 2

Physical Optimisation

Physical optimisation problems are mostly characterised by the fact that analogous to spin glasses a great number of solutions exist; but there are just a few optimal solutions. The effort to find these solutions shall be as small as possible. In practice a lot of exact methods exist which find the optimum with certainty. But the size of the system is strongly restricted by the computing time; thus a mathematical exact solution is not profitable for practical problems. In this case methods are needed approaching the optimum in the best way. The concept of "physical optimisation" is only referred to commercial problems and not to the simulation of physical systems. Physical optimisation is just derived from the simulation of large systems in statistical mechanics. Kirkpatrick et. al. proposed in 1983 [KGV83] to apply this method to commercial optimisation problems. In this way physical optimisation was developed to solve specific *economic* problems. The relation between the economic problem and physics is summarised in following table:

Physical System	**Optimisation Problem**
energy	objective function
temperature	control parameter
particle coordinates	parameters of the problem
system states	feasible solutions
low energy state	"good" solution

Table 2.1: Analogy between an optimisation problem and a physical system

In this chapter the background of physical optimisation is illustrated. In section 2.1 spin glasses as disordered materials exhibiting high magnetic frustration are explained. Spin glasses have many ground states which are never explored

on experimental time scales; thus Monte-Carlo methods are used for simulating the behaviour of spin glasses (2.2). At last the different physical optimisation algorithms themselves are presented in section 2.3.

2.1 Spin Glasses

Up to the seventies the history of physics was characterised by **ordered systems**. **Unordered** systems were almost totally neglected. Research concentrated on ideal structures like perfect crystals, because it was easier to find theories describing the physical properties. But those ideal structures can only be realised unter laboratory conditions. Because of this physicists started to observe unordered systems.

So the attempt was made to find out something about the behaviour of impure crystals. Therefore magnetic atoms of small concentrations were injected into a unmagnetic material, in order to examine the magnetic interaction; for example iron atoms in a gold crystal ($Au_{1-x}Fe_x$ with x as concentration) can be observed. At an iron concentration between 1 and 12 % in the gold crystal, the characterictic behaviour of a spin glass was observed. The abstract causes of this behaviour can also be found in economic systems; therefore spin glasses have a great significance for the optimisation of economic problems.

On the one hand the expression "spin glass" refers to the so called **spin** from quantum mechanics, which is responsible for magnetic effects. On the other hand the word **glass** is pointing to an unordered system; common window glass for example has no ordered structure like diamonds; the atoms are arranged unregularly. The main properties of spin glasses are **competition** and **coincidence** of the magnetic interactions. For a better understanding of the so called spin glass phase, some basics of the system will be explained at first. After that, a few experimental research results will be presented and the basic effects of the system described. At the end of this chapter different models of spin glasses are presented, which were used for computer simulation. This simulation models were the starting point for physical optimisation methods.

2.1.1 Magnetism

The simplest theory of magnetism says that special atoms behave like bar magnets. They produce magnetic fields and are influenced by external magnetic fields; the atoms interact with one another. Direction and strength of the magnetic effects can be described by the **magnetic moment** or the spin. The spin is produced by the charged particles of which an atom consists. If one puts a material of magnetic atoms in an external magnetic field, the spins will align in a

specific direction. In some materials internal effects can also lead to such a orientation. For one of these internal effects all spins align in the same direction. This orientation is especially responsible for the strong magnetic properties of iron; the effect is known as **ferromagnetism** and is caused by the exchange interaction of the metal atoms by the overlap of the electron sheath of adjacent atoms. Another effect is **anti-ferromagnetism**: adjacent spins point to different directions. The reason for this is again the overlap of the electron sheath. The total magnetic

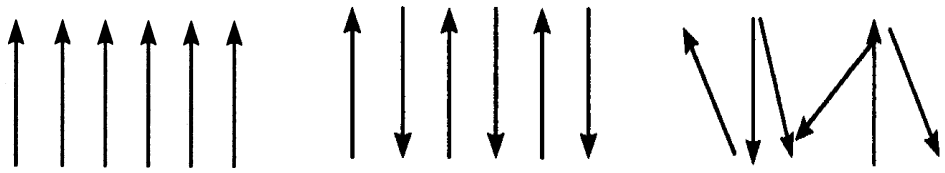

Figure 2.1: Schematic description of a ferromagnet (left), an anti-ferromagnet (middle) and a paramagnet (right)

energy of a ferromagnet has its minimum, when *all* spins point into the same direction. Energy is needed, when a spin shall turn into the opposite direction. If heat energy is added to the system, the order is influenced. For a temperature over the **Curie-point**, the direction of the singular spins changes because of the thermal movement. The ferromagnetic order of the system disappears and the material gets **paramagnetic**. This radical change of magnetic properties is called a **phase transition**. The spins are statistically distributed in all directions; the magnetisation disappears.

Spin glasses have both: ferromagnetic and anti-ferromagnetic interactions, which compete with on another [St93]. That is the main characteristic of spin glasses and it is a new form of magnetic order. Meanwhile the **spin glass behaviour** has been found in a lot of metals, semiconductors and insulators.

2.1.2 Theoretical / Experimental Results

RKKY - Interaction

The spin glass state differs intrinsically from normal magnetic systems. Famous examples of **metallic** spin glasses are copper with manganese (Cu_xMn_{1-x}) and iron with gold ($Au_{1-x}Fe_x$). An often examined, isolating spin glass is (EuS), which is magnetically thinned with unmagnetic (Sr)-ions ($Eu_xSr_{1-x}S$); EuS itself is ferromagnetic. But there are two competing interactions: a negative coupling between *adjacent* Eu-ions and a positive bond between the atoms, which are

one row further away. Beside the temperature the concentration x is responsible for the magnetic behaviour.

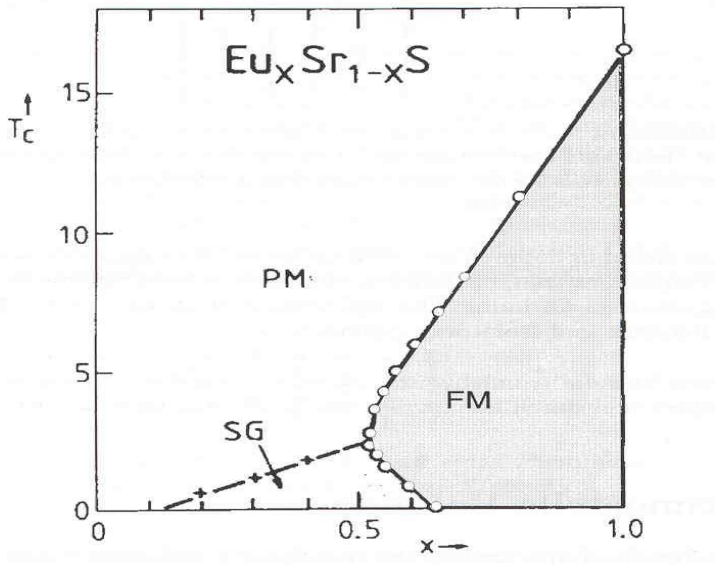

Figure 2.2: Magnetic phase diagram of $Eu_xSr_{1-x}S$

Figure 2.2 shows a **phase diagram** with a direct transition from the magnetic phase to the **spin glass phase** at a concentration x of the Eu^{2+}-ions between 13% and 51 %. Depending on the concentration x and the temperature T there exists a ferromagnetic (FM), a paramagnetic (PM) and a spin glass phase (SG). For x-values between 51 and 65 % at first there is a transition from the paramagnetic phase to the ferromagnetic phase, when the temperature is lowered. Because of the competing interaction the ferromagnetic order is highly disturbed; but the spin glass phase exists only for deep temperatures [Ko93].

A theoretical explanation for the positive and the negative bonds is given by the RKKY-interaction, named after Rudermann, Kittel, Kasuya and Yosida. This exchange interaction is based on the "polarisation" of the conduction electrons. Each charged particle has a magnetic moment (spin) and so does the electron. The polarised electrons itself influence the magnetic moments of the atoms, and so there is an intercation between the atoms themselves. For the strength of the bond J_{ij} it is valid:

$$J_{ij} \propto \frac{\cos(2\vec{k_F} \cdot \vec{r_{ij}})}{r_{ij}^3} \tag{2.1}$$

There $\vec{k_F}$ is the Fermi-wavevector. For positive values of $J_{ij}(r)$ the interaction is ferromagnetic; negative values cause a negative interaction. The RKKY-

interaction reaches many atoms in the neighbourhood and shows oscillatory behaviour. Depending on the distance of the atoms there is a ferromagnetic or an anti-ferromagnetic bond of the spins (Fig. 2.3, left side). Because of the competing interaction and a statistical distribution of the atoms in the crystal there are spin glass effects. The atoms can be imagined in the middle of a concentric sphere with decreasing strength of interaction further away. From shell to shell of the sphere, ferromagnetic and antiferromagnetic behaviour alternates. A spin glass can develop, when electrons and atoms interact. The electrons carry the interaction between the atoms, whose spins can turn up or down under the influence of the surrounding electrons.

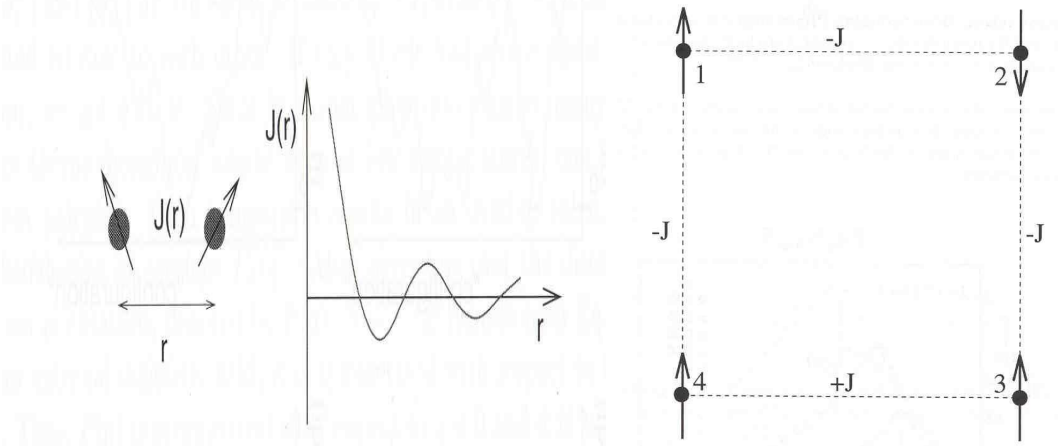

Figure 2.3: Schematic plot of the RKKY-interaction (left); Tag of four atoms (right)

Frustration

Approximately one half of the atomic pairs interacts ferromagnetic, the other half anti-ferromagnetic. This **dual** behaviour makes it possible that the spin of one atom cannot satisfy the interactions with all other atoms.

For illustration one can imagine a tag of four atoms which have an equal distance to one another (Fig. 2.3 on the right side). The interactions have the same amount, but for each pair of atoms the interaction is positive or negative. For an odd number of positive (negative) couplings in the tag not all interactions can be satisfied at the same time. Every configuration of spins at least cannot satisfy one of the bonds; the system is **frustrated**. This frustration effect causes that there are several low energy states and thus different configurations of spins with the same minimum energy. One speaks of degenerated energy states. Such effects are also characteristic for combinatorial optimisation problems: the costs

are going to be interpreted as energy and thus one gets several equal energy states.

Phase Transition

Because of the degeneracy of the lowest energy states it can be asked, whether the spin glass is a new state of matter or just a very slow paramagnet. At a real phase transition the final state has a characteristic order as long as the temperature does not change. The spin glass phase could be a clearly distinct phase, whose magnetic order remains at low temperatures. But the spin glass could also be a paramagnet with a very slow magnetic behaviour; thus it just seems to be a statistical phase. If it could be observed that one or several spins change their orientation at low temperatures, then this would be a proof for paramagnetic behaviour. For this the spin glass must be observed over a very long period of time.

Susceptibility, Heat Capacity, Magnetisation

In the lab one can search for hints of a phase transition. Those hints would be sudden changes in the magnetic and thermal characteristics at a critical temperature. A lot of experiments show the **spin glass phase**, for example measurements of the alternating field capacity. Susceptibility χ_{ac} gives information about the reaction of the spinsystem on a very weak, external alternating magnetic field.

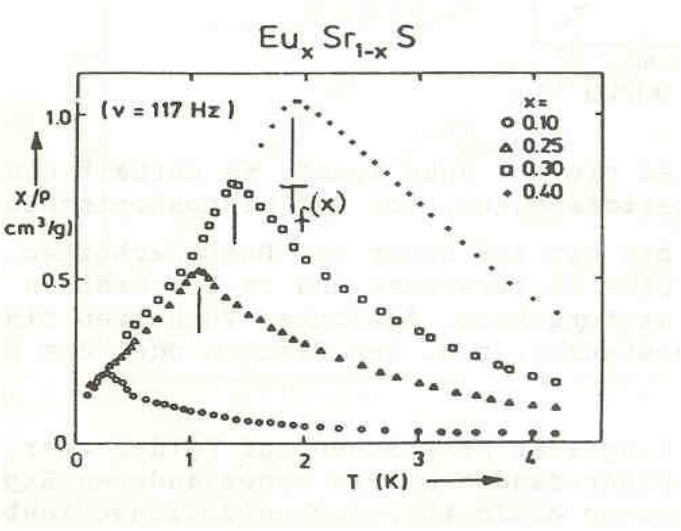

Figure 2.4: Alternating magnetic field susceptibility of $Eu_xSr_{1-x}S$

Figure 2.4 shows that χ_{ac} has a sharp peak at the **freezing temperature**

T_f. But this peak is rounded off even for small additional fields; moreover it depends on the frequency and the concentration of the used materials. So spin glasses have a peak in susceptibility χ at a temperature T_f and that indicates a phase transition. The heat capacity C on the contrary has a wide maximum at a temperature higher than T_f. So what happens at the temperature T_f ?

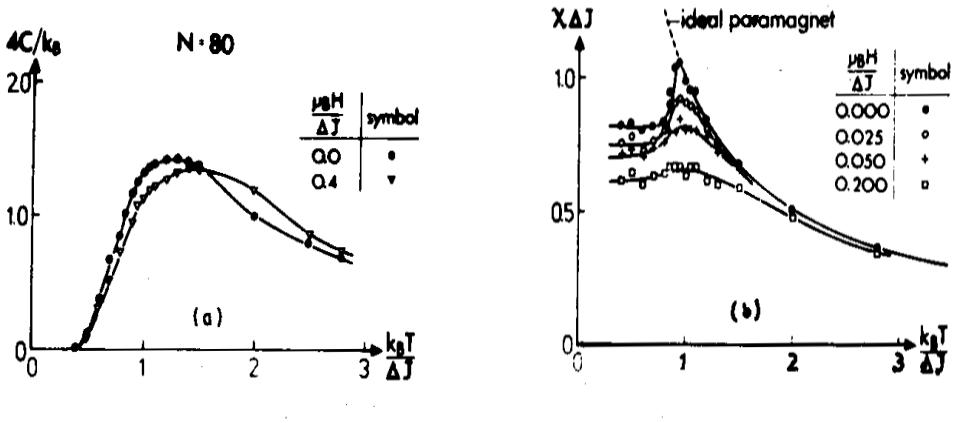

Figure 2.5: Heat capacity and susceptibility for different magnetising forces

At first, a phase transition into an antiferromagnetic order was assumed, but a suddenly appearing order should have shown up in the heat capacity. But this contradicts the fact that the specific heat is strictly monotonic increasing at T_f and has a wide maximum foremost above T_f. Furthermore scattering experiments with neutrons show that there is no periodic order. Neither a homogenous magnetisation nor a antiferromagnetic structure can be observed.

Another important characteristic is the influence of the observation time in the freezing temperature of the spin glasses. If $Eu_xSr_{1-x}S$ is observed over a long period of time, T_f can change up to 20 percent. This shows that spin glasses do not come to a rest. There is a great spectrum of relaxation times, from the microscopic time of 10^{-12} s to the time a spin needs to twist and up to many years. This behaviour can also be found in other incoherent systems like glasses, polymers and ceramics. Below T_f are many more or less equivalent spin configurations. The experimental realisation determines the taken states.

In order to understand the slow reaction of spin glasses at fields or other disturbances the magnetisation is measured. For the thermal equilibrium the mean magnetisation is $M = 0$. If the sample is cooled down without a magnetic field (**zero field cooling**) and then the external field is turned on for a short time, the sample remains magnetised (IRM). The same happens, when the sample is cooled down in a magnetic field (**field cooling**); after cooling the field is turned

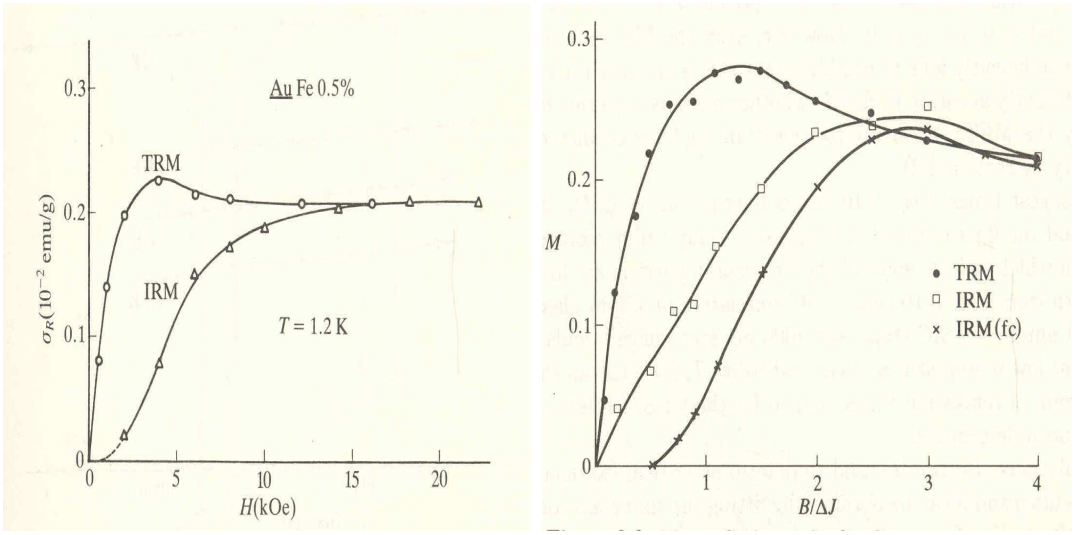

Figure 2.6: Remanent magnetisation of an AuFe-alloy (left) and a computer simulation(right)

off. The magnetisation fades away very slowly. This **remanent magnetisation** depends on the previously applied field, the temperature, the switch-on time and the rate of cooling; its existence shows that there are many stable states in a spin glass. That is the main difference between incoherent materials and pure crystals. The remanent magentisation is shown in Figure 2.6 on the left; the computer simulation on the right confirms a good synchronisation between experiment and theoretical model.

Review

The above listed phenomena can be understood with the frustration effects in a spin glass. Most of the materials with spin glass behaviour show two decisive effects: **disorder** and **competition** of the positive and negative couplings. This causes frustration and a high energy degeneracy of the system. In order to understand the characteristics of spin glasses, simplified models have been developed, which concentrate on the main mechanisms. In this way one gets a strongly idealised picture of a spin glass, which nevertheless contains all decisive physical aspects.

2.1.3 Mathematical Spin Glass Models

The theoretical description of phase transitions is very difficult. Physically and mathematically exact models often can be mastered only with a great numerical

effort. Therefore simplifying models have been developed. The simplifications are justified, if the main physical characteristics of a spin glass do not change. Thus abstract models have been developed, which are as simple as possible, but do not lose their physical content. The models can be tested by comparing the theoretical results of the simulations to the experiments.

Ising-Model

Mathematically simplified, spin glasses can be described by the Ising-model. Thus N locations in a 1-, 2- or 3- dimensional lattice are considered, where every lattice point i is associated with one spin s_i. In this model every spin has just two possibilities: $s_i = +1$ for spin-up and $s_i = -1$ for spin-down. Because of this it can be deduced that there are 2^N states in the phase space Γ. Every configuration $\sigma \in \Gamma$ of the lattice can clearly be determined by a set of variables $\sigma = s_1, s_2, \ldots, s_N$. The following Hamiltonian $\mathcal{H}$ describes the magnetic systems of the Ising-model:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j - \frac{2\pi}{h} g_S \mu_B B_0 \sum_{i=1}^{N} s_i \tag{2.2}$$

Here is:

J_{ij}	Exchange interaction between s_i and s_j
$\langle i,j \rangle$:	only adjacent spins are summed up
J_{ij}	Exchange interaction between spins s_i and s_j
B_0	external magnetic field
g_S	Lande-factor
μ_B	Bohr Magneton
h	Planck's constant

Table 2.2: Parameters of the spin glass Hamiltonian

Mostly the constants g_S, μ_B are set to unity. Then B_0 is chosen in such a way that the magnetic moment per spin equals unity. It follows:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j - B_0 \sum_{i=1}^{N} s_i \tag{2.3}$$

The first term describes the sum of the exchange energies between all spin pairs s_i and s_j. The other term considers the interaction of the spins with an external

magnetic field; the field tries to align the spins in the same direction. For a positive exchange interaction all spins are parallel in the ground state; this spin structure is called ferromagnetic. For a negative coupling J_{ij} the structure is called anti-ferromagnetic. The Ising-model was analytically solved from Ising in the year 1925 for next neighbours with $J_{ij} = J$ and $J = 0$ else. Onsager solved the two dimensional problem analytically without an external field, Yang in 1952 with $B_0 \neq 0$. But there is no analytical solution for the three dimensional Ising-model. In general it is very difficult to estimate the couplings J_{ij} theoretically. Thus it is appropriate to adjust the coupling constants to the experimental results.

Heisenberg-Model

On the basis of the Ising-model Heisenberg developed an improved 3-dimensional model in 1928. This model could benefit from further developments of quantum mechanics at that time. For isotropic ferromagnets following Hamiltonian is valid:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij} \vec{s_i} \cdot \vec{s_j} - B_z \sum_{i=1}^{N} s_i^z \tag{2.4}$$

with J_{ij}=±J and $|\vec{s_i}| = 1$, $|\vec{s_j}| = 1$. In contrast to the Ising-model the spins are considered to be 3-dimensional vectors, which can have any direction in space. The Heisenberg-model has a very universal form and contains the Ising-model as 1-dimensional case.

XY-Model

The two dimensional Heisenberg model is called XY-model and has the Hamiltonian:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij}(s_i^x s_j^x + s_i^y s_j^y) - B_x \sum_{i=1}^{N} s_i^x \tag{2.5}$$

$$\text{where again:} \qquad (s_i^{x\,2} + s_i^{y\,2}) = 1 \quad \Leftrightarrow \quad |\vec{s_i}| = 1. \tag{2.6}$$

so that $\vec{s_i}$ can be illustrated as

$$\begin{pmatrix} \cos\Phi_i \\ \sin\Phi_i \end{pmatrix}$$

and then with the addition theorem

$$\sin(\Phi_i)\sin(\Phi_j) + \cos(\Phi_i)\cos(\Phi_j) = \cos(\Phi_i - \Phi_j) \tag{2.7}$$

it can be written:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij} \cos(\Phi_i - \Phi_j) - B \sum_{i=1}^{N} \cos(\Phi_i) \tag{2.8}$$

Φ_i, Φ_j are the phases of the spins; $\Phi_i - \Phi_j$ is their phase difference. In this model the spin can turn in the XY-plane, where it has constantly an amount of 1.

Edward-Anderson-Model

This model is most analysed and also based on the Hamiltonian of the Ising-model. The spins are placed on the vertices of a cubic 3-dimensional lattice; the spins are Ising-spins with two possible settings. The range of interaction is reduced to next neighbours. The basic characteristics of disorder and competition are represented by a statistical distribution $P(J_{ij})$ of the couplings. The strength of a coupling J_{ij} depends on the distance of two spins s_i and s_j ; but it disappears when there is another spin in between. The distribution is a Gaussian one with a standard deviation Δ and an expectation value of zero:

$$P(J_{ij}) = \frac{1}{\sqrt{2\pi}\Delta} \exp\left(-\frac{J_{ij}^2}{2\Delta^2}\right) \tag{2.9}$$

The EA-model replaces the **site-disorder** by a **bond-disorder**. Site disorder means the random distribution of magnetic atoms in space and bond disorder the statistical distribution of couplings J_{ij}. The ferro- and antiferromagnetic interactions are uniformly distributed. With this model it is possible to ask for the ground state as the main physical feature. How do the spins have to align, so that the spin glass is in a state of minimum energy? Because of the frustration effects it is difficult to find the ground state. The total energy depends on the number of unsaturated bonds. Therefore those spin configuration has to be taken, which saturates most bonds.

$\pm J$-Model

Toulouse et al. found out that the behaviour of spin glasses is mainly characterised by frustration effects and therewith by the sign of J_{ij}. So the $\pm$J-model was developed. Here only the interactions between next neighbours are considered, too. The strength of the exchange interaction J_{ij} is $+J$ and $-J$ with a probability of 50% each. The Hamilton operator can be described as:

$$\mathcal{H} = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j - B_0 \sum_i s_i \tag{2.10}$$

This is a strong abstraction of the complex physical facts, but it contains the main features of spin glasses. Especially the freezing of the system in disordered ground states can be shown.

2.2 Monte-Carlo-Methods

2.2.1 Statistical Physics

In classical physics problems with limited particle number can be exactly described by the Newton equations. If one knows all physical variables, which determine the system at a defined time t_0, the state of the system can be forseen exactly for all later times t.

For complicated **many-particle-systems** this is not possible any more; thus statistic variables are used. The analysis of the mean values then makes it possible to say something about the macroscopic behaviour of the system. Because of the great number of configurations in optimisation problems, physical optimisation also uses methods of statistical physics. In this context the observed systems generally can be described as **canonical ensembles**. These are closed systems which are in thermal contact with a heat bath and energy, but no particles can be exchanged. The system is in a thermal equilibrium, when the temperature T of the system is equal to the temperature of the heat bath. If that is the case, the probability distribution of any state σ can be described by the **Boltzmann distribution** [No02].

$$P_{equ}(\sigma) = \frac{1}{Z}\exp\left(-\frac{\mathcal{H}(\sigma)}{k_B T}\right) \tag{2.11}$$

k_B is the Boltzmann constant and Z the **state sum**, which plays a major role in statistical physics and is a scale factor for the calculation of many variables. The partition factor is given by:

$$Z = \sum_{\sigma\in\Gamma} \exp(-\beta\mathcal{H}(\sigma)) \tag{2.12}$$

with $\mathcal{H}$ as Hamilton-function and $\beta = \frac{1}{k_B T}$. The mean value or thermal expectation value in a discrete system with an observable $\mathcal{A}$ can be calculated as follows:

$$\langle\mathcal{A}\rangle = \sum_{\sigma\in\Gamma} \mathcal{A}(\sigma)P_{equ}(\sigma) = \frac{\sum_{\sigma\in\Gamma} \mathcal{A}(\sigma)\exp\left(-\frac{\mathcal{H}(\sigma)}{k_B T}\right)}{\sum_{\sigma\in\Gamma}\exp\left(-\frac{\mathcal{H}(\sigma)}{k_B T}\right)} \tag{2.13}$$

For $\mathcal{A} = \mathcal{H}$ one gets the expectation value of the Hamiltonian, which can also be

expressed through the logarithmic derivation of the partition function:

$$\begin{aligned} -\frac{\partial}{\partial\beta}\ln Z &= -\frac{1}{Z}\sum_{\sigma\in\Gamma}\frac{\partial}{\partial\beta}\exp(-\beta\mathcal{H}(\sigma)) \\ &= \frac{1}{Z}\sum_{\sigma\in\Gamma}\mathcal{H}(\sigma)\exp(-\beta\mathcal{H}(\sigma)) \\ &= \langle\mathcal{H}\rangle \end{aligned} \tag{2.14}$$

From that the **heat capacity** $C(T)$ can be derived:

$$\begin{aligned} C(T) &= \frac{d\langle\mathcal{H}\rangle}{dT} \\ &= \frac{1}{k_B T^2}\left[\frac{1}{Z}\sum_{\sigma\in\Gamma}\mathcal{H}^2(\sigma)\exp(-\beta\mathcal{H}(\sigma)) - \left(\frac{1}{Z}\sum_{\sigma\in\Gamma}\mathcal{H}(\sigma)\exp(-\beta\mathcal{H}(\sigma))\right)^2\right] \\ &= \frac{1}{k_B T^2}\left[\langle\mathcal{H}^2\rangle - \langle\mathcal{H}\rangle^2\right] \\ &= \frac{1}{k_B T^2}Var(\mathcal{H}) \end{aligned} \tag{2.15}$$

Because of the relationship between the heat capacity and the variance $Var(\mathcal{H})$, this variable is significant for simulation: the observation of $C(T)$ shows at which temperature the greatest changes occur. The system must be in a thermal equilibrium at any temperature; otherwise the Boltzmann-distribution cannot be used. The equilibrium needs time to be set; that has to be considered in simulation. In statistical physics systems in a thermal equilibrium are analysed numerically with **Monte-Carlo-methods**. This methods are algorithms which use random numbers to calculate mean values in statistical systems. But how can theoretically derived observables be calculated in practice? For an exact calculation all possible states of the systems must be considered. But in practice it is difficult to do this. Therefore the thermal expectation values are determined by a limited number of configurations. In order to get close to realistic values, two methods have been developed: *simple sampling* and *importance sampling*.

2.2.2 Simple Sampling

The basic idea of simple sampling [BH02] is to replace the exact equations for the thermal expectation values through a sum, which does not consider *all* possible configurations $\sigma_1, \ldots, \sigma_G$. Instead, a statistical selection of characteristic points $\sigma_1, \ldots, \sigma_M$, $M \leq G$ is taken from the phase space. The expectation value of an

observable is:

$$\bar{\mathcal{A}} = \frac{\sum_{i=1}^{M} \mathcal{A}(\sigma_i) P_{equ}(\sigma_i)}{\sum_{i=1}^{M} P_{equ}(\sigma_i)} \tag{2.16}$$

The points σ_i are randomly selected from the whole phase space. For the extreme case the equation:

$$\lim_{M \to G} \bar{\mathcal{A}}(\sigma) = \langle \mathcal{A}(\sigma) \rangle \tag{2.17}$$

holds. The method is called simple sampling, because every configuration is determined with uniformly distributed random numbers. In practice this method only shows good results for small systems or at very high temperatures, because each configuration is selected with the same probability.

But the distribution function of a macroscopic variable is strongly centered around its expectation value. Therefore only a small area of the phase space contributes significantly to the thermal expectation value of an observable. The distribution function $P_T(E)$ of the observable E shows for the temperature T a peak at E_T with a full width at half maximum proportional to $\frac{1}{\sqrt{N}}$. There N is the number of freedom degrees. Besides the scope of critical temperatures the distribution has the form:

$$P_T(E) \propto \exp\left(-N \frac{(E - \langle E \rangle_T)^2}{2 \cdot CT^2}\right) \tag{2.18}$$

With a decreasing temperature, E_T goes down and the distribution changes. But simple sampling chooses points of the phase space, which are common for the distribution at $P_\infty(E)$ and not for lower temperatures. The left curve in Figure 2.7 describes the energy distribution of a canonic ensemble at low temperatures. The right curve shows the distribution, which is produced by simple sampling and is the equivalent to an infinite high temperature with $\langle \mathcal{H} \rangle = 0$. The distribution $P_T(E)$ is very small for low energies because of the exponential decrease. Thus simple sampling mostly produces physically unimportant configurations at low temperatures. What follows is a wrong calculation of the physical variables. These disadvantages can be prevented by "importance sampling" of Metropolis.

2.2.3 Importance Sampling

Just like simple sampling, importance sampling takes a selection $\sigma_1, \ldots, \sigma_M$ of all possible states $\sigma_1, \ldots, \sigma_G$. The points $\sigma_1, \ldots, \sigma_M$ are not selected with the same probability, but with a special distribution $P(\sigma_i)$. It follows for the observables:

$$\bar{\mathcal{A}} = \frac{\sum_{i=1}^{M} \mathcal{A}(\sigma_i) P_{equ}(\sigma_i)/P(\sigma_i)}{\sum_{i=1}^{M} P_{equ}(\sigma_i)/P(\sigma_i)} = \frac{1}{M} \sum_{i=1}^{M} \mathcal{A}(\sigma_i) \tag{2.19}$$

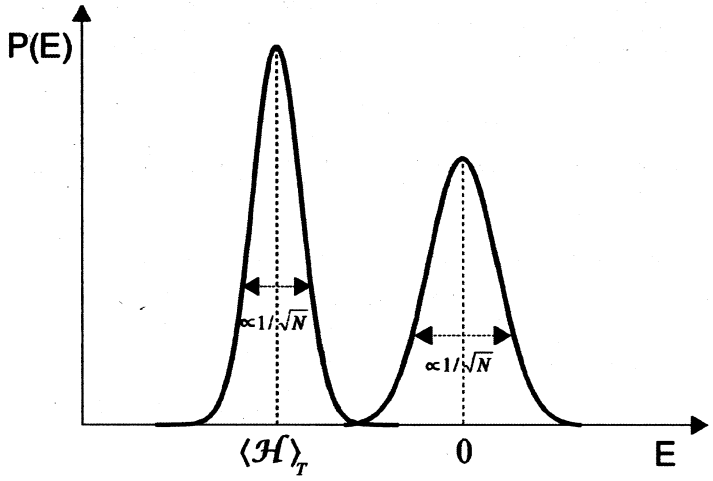

Figure 2.7: Probability distribution of the energy E

Thus the expectation value of the observable $\mathcal{A}(\sigma)$ shall be equivalent to the arithmetic mean. Metropolis et al. demanded that produced states σ_i which follow one another shouldn't be dependent. A state σ_{i+1} shall be produced with an adequate probability $W(\sigma_i \rightarrow \sigma_{i+1})$, which depends on the previous state. That is called **Markov process**. The transition probability shall be chosen in such a way that the distribution function $P(\sigma_i)$ is equal to $P_{equ}(\sigma)$ in the limiting case $M \rightarrow G$. An important but not necessary condition is the principle of **detailed balance**:

$$P_{equ}(\sigma_i)W(\sigma_i \rightarrow \sigma_{i'}) = P_{equ}(\sigma_{i'})W(\sigma_{i'} \rightarrow \sigma_i) \tag{2.20}$$

If one puts Equation 2.11 in Equ. 2.20 and shifts around, it can be seen that the transition probability only depends on the energy change $\Delta\mathcal{H} = \mathcal{H}(\sigma_{i'}) - \mathcal{H}(\sigma_i)$.

$$\frac{W(\sigma_i \rightarrow \sigma_{i'})}{W(\sigma_{i'} \rightarrow \sigma_i)} = \exp\left(-\frac{\Delta\mathcal{H}}{k_B T}\right) \tag{2.21}$$

The transistion probability $W(\sigma_i \rightarrow \sigma_{i'})$ is not fully determined by this equation. Normally it is chosen:

$$\begin{aligned} W(\sigma_i \rightarrow \sigma_{i'}) &= \frac{1}{2}\left[1 - \tanh\left(\frac{\Delta\mathcal{H}}{2k_B T}\right)\right] \\ &= \frac{\exp\left(-\frac{\Delta\mathcal{H}}{k_B T}\right)}{1 + \exp\left(-\frac{\Delta\mathcal{H}}{k_B T}\right)} \end{aligned} \tag{2.22}$$

Or alternatively:

$$W(\sigma_i \to \sigma_{i'}) \quad = \quad \begin{cases} \exp\left(-\frac{\Delta\mathcal{H}}{k_B T}\right) & : \text{for} \quad \Delta\mathcal{H} > 0 \\ 1 & : \text{else} \end{cases} \tag{2.23}$$

Equation 2.22 is the so called **Glauber function** and Equ. 2.23 the **Metropolis function**. With this transistion probabilities a sequence of states $\sigma_i \to \sigma_{i'} \to \sigma_{i''}$ is produced. What remains to do is to show that the probability distribution $\mathrm{P}(\sigma_i)$ converges to $P_{equ}(\sigma_i)$. This can be shown with the **central limit theorem** of probability theory; the complete proof can be found in the corresponding literature.

Simulation of the ±J-Model

In the following an explanation shall be given, how the ±J- model can be simulated with the single-spin-flip algorithm. For this a lattice of the size L×L×L with periodical constraints shall be given. Every lattice point is occupied with one spin s_i; the start configuration is random. The interaction J_{ij} between adjacent spins is randomly chosen with $+J$ or $-J$ and remains constant during the simulation. The next steps are shown in Table 2.3.

1.	Selection of a lattice point i with s_i.
2.	Calculation of the energy change, when the spin turns from s_i to -s_i.
3.	Calculation of the transition probability W for this spinflip.
4.	Selection of a random number Z between zero and one
5.	Spinflip for Z<W; no spinflip for Z≥W.
6.	Calculation of the interesting variables: energy, heat capacity, magnetisation, susceptibility.

Table 2.3: Simulation of the ±J-Model

Configurations which follow one another just differ in one spin; thus the physical properties are highly correlated. Moreover the calculation time of the thermal expectation values is very extensive. Therefore the expectation values should only be calculated from time to time. The physical interpretation is that at the beginning there is no thermal equilibrium in the system and many new configurations have to be produced before measuring the single variables.

2.3 Optimisation Algorithms

The *random walk* (RW) is the simplest acceptance rule. Every transition $\sigma \to \sigma'$ is accepted, no matter where σ' is:

$$p(\sigma \to \sigma') = 1 \tag{2.24}$$

That is the equivalent to $T \to \infty$. Every possible configuration can be reached by a random walk, but the way through the energy landscape is completely random. Thus the RW is only applied, when the energy landscape has a smooth structure. In addition to that the RW does not fulfil the condition of detailed balance.

The counterpiece to the random walk is the **greedy**; just those configurations are accepted, which lead to a configuration with the same or a better quality:

$$p(\sigma \to \sigma') = \Theta(-\Delta\mathcal{H}) \tag{2.25}$$

$\Theta(x)$ is the **Heaviside function** and $\Delta\mathcal{H} = \mathcal{H}(\sigma_{i'}) - \mathcal{H}(\sigma_i)$. The greedy goes straight to the next local minimum. And that is the problem: mostly the greedy is trapped in a local minimum without the possibility to reach the global minimum. Thus the greedy is mostly used for systems with energy landscapes, which either have just a few local minima or the energy differences between the local minima and the global maximum are very small. But normally the energy landscapes are not known and their shape lies somewhere between the extremes. Therefore one tries to combine the advantages of both algorithms. A good algorithm has to compare the current energy configuration with the new one; it has to accept a temporary worsening of the momentary configuration, because only in this way the global minimum can be found.

2.3.1 Simulated Annealing - SA

The method SA [KGV83] uses an intrinsic system temperature to optimise the system. There is a cooling schedule according to which the system is cooled down from a high temperature to a low temperature. Step by step the free movement through the phase space is limited. The transition probability between two states σ and σ' is determined by the Metropolis function:

$$W(\sigma_i \to \sigma_{i'}) = \begin{cases} \exp\left(-\frac{\Delta\mathcal{H}}{k_B T}\right) & : \text{for} \quad \Delta\mathcal{H} > 0 \\ 1 & : \text{else} \end{cases} \tag{2.26}$$

This optimisation algorithm is called **simulated annealing**. $\Delta\mathcal{H}$ is the change of energy which is the result of the transition from state σ to σ'. Normally, the Boltzmann constant k_B is set to one and T is calculated in units of $\mathcal{H}$. SA is the

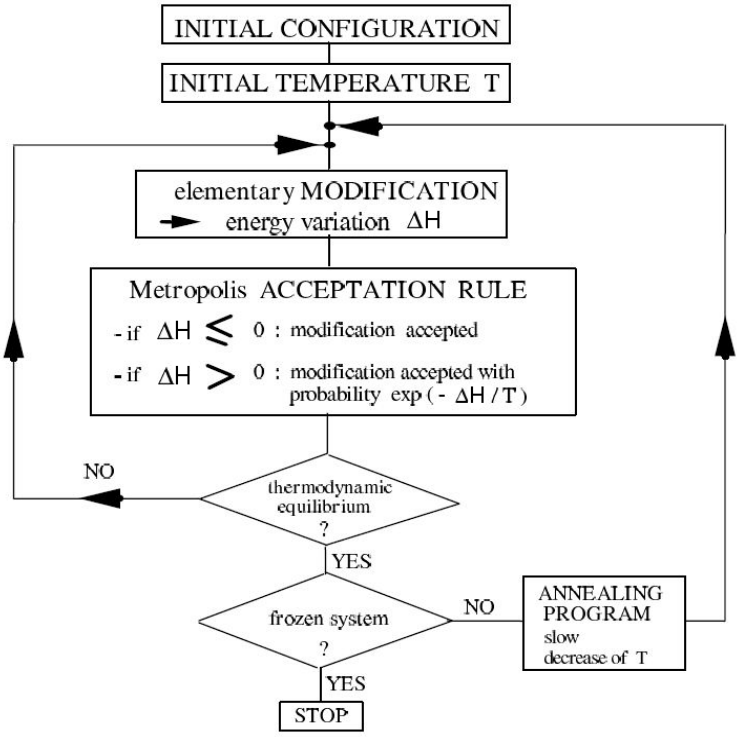

Figure 2.8: Flow chart of simulated annealing

classical optimisation algorithm in physics; it is used to find low energy states in complex systems for which no analytical solution exists.

The name of the method comes from metallurgy: in the annealing process the metal is heated for a long period of time and then cooled down slowly. At lower temperatures the atoms in the lattice can move less freely. If the system is cooled down very slowly, it remains in thermal equilibrium and the atoms can arrange themselves in the ground state, even at low temperatures. If the cooling is too fast, polycrystalline or amorphous structures with a higher energy are formed.

SA fulfils the condition of **ergodicity**: a system is ergodic, if every point of the phase space can be reached. The ergodicity is essential for the calculation of the expectation values of the observables. In ergodic systems band average and time average are equal. But that is *not* fulfilled for glassy systems. Here the measure time τ is crucial; the system has to find its equilibrium in the time τ. SA is a powerful method to treat combinatorial optimisation problems. The algorithm can also be applied to many $\mathcal{NP}$**-complete** problems. $\mathcal{NP}$-complete problems are problems for which no deterministic algorithm exists which solves the problem optimal in a time $t < N^x$. Thereby is N the system size and the exponent x is an upper bound for the computation time. *Complete* means that

all problems of this class can be transformed into one another by a polynomial map (see also section 1.3.2).

2.3.2 Threshold Accepting -TA

TA [DS90] is an optimisation algorithm with a formal similiarity to SA. But the transition probability from one configuration σ_i to another $\sigma_{i'}$ is defined in a different way:

$$W(\sigma_i \to \sigma_{i'}) = \Theta(Th - \Delta\mathcal{H}) = \begin{cases} 1 & : \text{for} \quad \Delta\mathcal{H} \leq Th \\ 0 & : \text{else} \end{cases} \tag{2.27}$$

Θ is the Heaviside function and Th is called **threshold**. Th is some kind of a temperature or a control parameter. During the optimisation process Th is lowered from a high start value to zero. This method guarantees that a new configuration $\sigma_{i'}$ is accepted, if it is much worse than the former configuration σ_i.

In contrast to TA simulated annealing can also accept solutions with a low probability. Because of this, the condition of ergodicity is not fulfilled for TA: not every point of the phase space can be reached and also no thermal equilibrium. Therefore TA is a non-equilibrium algorithm and is thus no physical method. Also the principle of "detailed balance" is not fulfilled.

A special problem of TA are the so called **golfholes**. If a configuration $\sigma*$ has only neighbours σ_i for which is valid:

$$\mathcal{H}(\sigma_i) - \mathcal{H}(\sigma*) > Th,$$

then those neighbours cannot be reached from $\sigma*$. Some energy landscapes have relatively narrow and deep local minima. If the threshold is to small, the system cannot get out of it and is trapped. In contrast to TA, SA can leave the golfhole in finite time. For this reason it is better to make several optimisation runs with TA.

TA is not a physical algorithm and thus the calculated variables have no real physical meaning. But the variables have the same relation to one another and allow essential statements about the system; therefore the variables get the known names. TA can be seen as approximation of SA, if the area under the curves for the transition probability is compared. The Heaviside function is used instead of the exponential curve of SA. Therefore T and Th have the same order when the system changes its state from low to high order.

In spite of the disadvantages TA is an established method. The great advantage is that one does not need to calculate the exponential function like in SA. TA just compares Th to $\Delta\mathcal{H}$ and is therefore faster. In practice one makes several short optimisation runs with TA; normally good solutions are reached with small effort.

2.3.3 Great Deluge Algorithm - GDA

Another simple and successful optimisation method is the GDA. A random walk through a part Γ_S of the phase space Γ is carried out [Nu93]. Every configuration $\sigma_i \in \Gamma_S$ is characterised by the fact, that the energy of σ_i lies below a special level T_S. The transition probability from $\sigma_i \in \Gamma_S$ to $\sigma_j \in \Gamma$ is given through the Heaviside function:

$$W(\sigma_i \to \sigma_j) = \begin{cases} 1 & : \text{for} \quad \mathcal{H}(\sigma_j) \leq T_S \\ 0 & : \text{else} \end{cases} \tag{2.28}$$

Every configuration σ_i with a lower energy level than T_S is accepted with the same probability. T_S is called **water level** or pseudo temperature. By a slow lowering of the water level T_S the system is forced to take energetically lower configurations. Just like TA the GDA has problems to get stuck in a a local minimum. The condition of ergodicity is violated, because not all points of the phase space can be reached and there is no thermal equilibrium. GDA is a non-equilibrium algorithm. But "detailed balance" is fulfilled: for a given T all configurations under the local level T_S have the same possibility.

The algorithm is named after the great deluge in the old testament. If one changes the problem and wants to know the maximum of the phase space, T_S can be interpreted as water level, which rises continuously. The problem here is that islands in the energy landscape are formed, if the water is rising: probably the system did not reach the island with the highest mountain, but a very low mountain island. That is not such a big problem, because the state σ_i has a lot of neighbours in the high dimensional phase space and can be left on many ways, when the water rises. This is the reason, why GDA shows very good results for many optimisation problems.

2.3.4 Cooling Scheme

For SA cooling methods have been developed, which guarantee a global minimum for an infinite long calculation time. There the temperature has to be like following:

$$T_k = \frac{a}{b + \log(k)} \tag{2.29}$$

a and b are positive and system-dependant constants; k is the number of already executed temperature steps. The decisive disadvantage of this method is that the calculation time is longer than the complete enumeration of all configurations. Another problem is that it's not clear, whether the real optimum has been found. So this cooling strategy is of no use in practice; instead empirical curves are used, which converge faster.

The first empirical method to mention is **linear cooling**. The temperature is reduced constantly with ΔT:

$$T_k = T_{start} - k \cdot \Delta T \qquad \text{with} \qquad 0.01 \leq \Delta T \leq 0.5 \tag{2.30}$$

T_{start} is the start temperature, which has to be determined for each optimisation run. Besides T_k mustn't be smaller than zero; the optimisation run has to be stopped before.

The **logarithmic** or **exponential** cooling uses a repeated multiplication of the start temperature with the factor α:

$$T_k = \alpha^k \cdot T_{start} \qquad \text{with} \qquad 0.8 \leq \alpha \leq 0.999 \tag{2.31}$$

The best cooling method depends on the optimisation problem. Therefore a test run has to be made, in order to estimate the curves of the physical variables. Especially from the heat capacity it can be derived, how the system behaves. If the system freezes very fast, the linear cooling method is chosen; in the opposite case the logarithmic method is better.

Start and End Temperatures

The correct start temperature is important for the optimisation run. If it is too high, calculation time is wasted at the beginning; if it is too low, the solutions are bad. The start temperature cannot be given directly, because it depends on the single optimisation problems. For SA a good start temperature can be found in the following way: for the temperature T_{start} the system shall be able to move more or less freely in the phase space. At the beginning, transitions shall be accepted which raise the energy of the system. The acceptance rate P_{acc} for this transitions can be set freely. Then a random walk through the phase space is made and the number n of transitions measured, which raise the energy level of the system. The number of accepted transitions for simulated annealing can be approximated as follows:

$$n_{acc} \approx n \cdot \exp\left(-\frac{\Delta\bar{\mathcal{H}}_+}{T_{start}}\right) \tag{2.32}$$

where $\Delta\bar{\mathcal{H}}_+$ is the expectation value of the transitions raising the energy level of the system. The acceptance level P_{acc} is given by

$$P_{acc} = \frac{n_{acc}}{n} \approx \exp\left(-\frac{\Delta\bar{\mathcal{H}}_+}{T_{start}}\right) \tag{2.33}$$

It follows

$$T_{start} \approx -\frac{\Delta\bar{\mathcal{H}}_+}{\ln(P_{acc})} \tag{2.34}$$

Mostly the acceptance rate is chosen between 80 and 90 %. Of course this is a very rough estimation of T_{start}, but the order of the temperature can easily be found with this method. A similiar consideration can be made for TA and GDA:

$$T_{start} \approx \Delta\bar{\mathcal{H}}_{+} \qquad \text{threshold accepting} \tag{2.35}$$

$$T_{Sstart} \approx \mathcal{H}_{max} \qquad \text{great deluge algorithm} \tag{2.36}$$

The end temperature T_{end} shall be determined in such a way that the system is mostly frozen; the acceptance rate of all transitions shall tend to zero. But in degenerated systems there can be transistions with no effect on the energy. Those don't need to be considered, when the acceptance rate is calculated. It is also good to make several steps at $T = 0$. If the acceptance rate of all non-trivial transitions is zero over a long period of time, the optimisation run can be stopped. Of course there is no security, whether the global optimum is reached, but the probability is very high to find a local optimum near the global one. Especially for systems with very broad energy valleys in the area of the global minimum the acceptance rate could be clearly above zero, in spite of a constant energy level. In this case it is better to take the Hamilton function as a criterion for the closeness to the optimum: if the energy doesn't change over several temperature steps, the system can assumed to be frozen.

Chapter 3

Different Metaheuristics

Physical optimisation is a global optimisation technique which traverses the search space by generating neighbouring solutions of the current solution. A superior neighbour is always accepted. An inferior neighbour is accepted probabilistically based on the difference in quality and a temperature parameter. The temperature parameter is modified as the algorithm progresses to alter the nature of the search. In order to have a contrast to physical optimisation, some related algorithms are presented in the following:

At first **genetic algorithms** (GA) which maintain a pool of solutions. New solutions are generated not only by "mutation" as in simulated annealing (SA), but also by "combination" of two solutions from the pool. Probabilistic criteria, similar to those used in SA, are used to select the candidates for mutation or combination, and for discarding excess solutions from the pool.

Secondly, **evolution strategies** (ES) which evolve individuals by means of mutation as well as intermediate and discrete recombination. Thus ES are very similiar to genetic algorithms; they are designed particularly to solve problems in the real-value domain; they use self-adaptation to adjust control parameters of the search.

The metaheuristic **tabu search** (TS) is similar to SA: both traverse the solution space by testing mutations of an individual solution. While SA generates only one mutated solution, tabu search generates many mutated solutions and moves to the solution with the lowest energy of those generated. In order to prevent cycling and encourage greater movement through the solution space, a tabu list is maintained of partial or complete solutions. It is forbidden to move to a solution that contains elements of the tabu list which is updated as the solution traverses the solution space.

At last **ant colony algorithms** are presented. They use many ants to traverse the solution space and find locally productive areas. While usually inferior to genetic algorithms and other forms of local search, it is able to produce results

in problems where no global or up-to-date perspective can be obtained, and thus the other methods cannot be applied.

In this dissertation tabu search and ant colony algorithms are introduced for the sake of completeness. Only genetic and physical optimisation algorithms are applied. Thereby the focus is on physical optimisation; for evaluation and comparison of the results also a genetic algorithm was implemented.

3.1 Genetic Algorithms - GA

3.1.1 Biological Background

Selection, **mutation**, **crossover** and the principle **"survival of the fittest"** are the essential building stones of genetic algorithms (GA). For a better understanding of GA thus the biological background has to be explained. In 1859 Charles Darwin published his famous work *"On the Origin of Species"*. Therein he declared that all living beings have developed from primitive species. Therefore the concept of **Darwinism** is the theory, which considers the natural selection as the main factor in the development of species. Thereby two things are presupposed:

1. Random variation of non-aligned heritable characteristics

2. Overproduction of descendants (offspring); those organisms survive, which have the best adjustment to the environment.

In the long run each species produces more offspring than food supplies are available; thus a so called **selection-pressure** evolves. This leads to a decrease of population members until there are enough food supplies. Further on it is important for Darwinism that living beings of each species have a more or less strong variation in their heritable factors. Those heritable variations which have been approved in the fight for survival will occur more often in the following generations. Over several generations the small variations can lead to a perfection and optimisation of all creatures. The fight for survival causes a natural selection; the most suitable individuals have the highest chance to survive. Therefore this concept is called "survival of the fittest". Because just the fittest survive, in the following generations especially those characteristics will be inherited which are responsible for survival.

Genetics is a section of biology which is concerned with the inheritance and variation of organisms. The central theme is to clear the function of genes and the way they are inherited. In order to understand what genes are, one has to know what a cell nucleus is: it is a kind of control center of the cell and has great

importance for inheritance processes, because it contains the **chromosomes** as carrier of the inherited material. The chromosomes have the form of a thread and carry the genes. Chromosomes consist of nuclein acids and proteins; the most important nucleic acid is the **deoxyribonucleic acid** (**DNA**). Man have some millions of nucleotides in their DNA; the DNA looks like a double-strand molecule. The nucleotides are the basic building stones of the nucleic acids. The double-strand molecule of the DNA is connected by hydrogen bonds to the famous *double helix*.

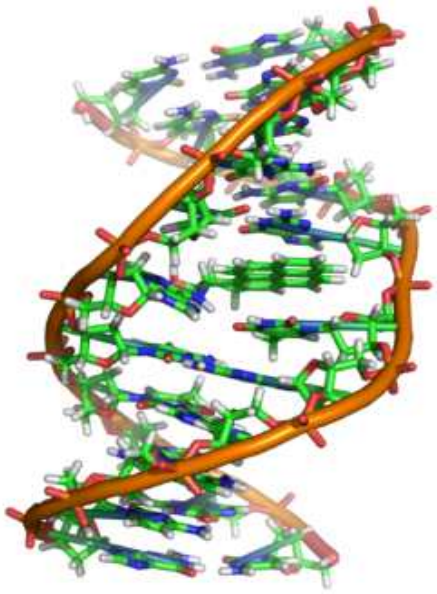

Figure 3.1: Structure of the DNA

The structure of the DNA is built on 4 alkalis: *adenine, guanine, cytosine* and *thymine*. The alkalis are a kind of alphabet for the genetic code. In the protein synthesis the alkali sequences are translated into special amino acids. This amino acids form the proteins, whereby the molecules of a protein own the amino acid sequence; those sequence is genetically determined. The proteins in turn are the most important building stones of the cells; beside others they control the whole metabolism. After this the alkalis code the whole structure of a living being.

A gene is known as carrier of the genetic information. Genes are special parts of the DNA, which serve for the production of polypeptid chains. Those chains are necessary for the formation of the protein molecule. Those genes are some kind of a unit of the DNA which contains information about the production of proteins. The chains can underly spontanous changes and are characterised as mutations. If these mutations happen in germ cells, they can be inherited;

mutations in body cells are not inheritable, but mostly lead to a damage of the living being. Gene mutations can be caused by outer and inner influences. Chemicals and rays, especially high energy rays like X-rays and UV-light are the most important reasons for mutations. The frequency of mutations for living beings is relatively different; it characterises the number of mutations per gene and generation, which is mostly very low. For higher living beings a mutation frequency of one mutation per $10^5 - 10^9$ genes is expected. For more simple organisms the frequency is even lower.

Another possibility to change the DNA is the process of cell and nucleus separation. Hereby two types have to be distinguished: the **mitosis** and **meiosis**. The mitosis is a hereditary-similiar cell and nucleus separation from unsexual reproduction; the new cells have the same genetic information as the origin cell. More essential for evolution is the meiosis. During the meiosis chromosomes are recombined by the so called crossover and then randomly distributed to the different germ cells. The meiosis is responsible for the combination of the genetic material.

3.1.2 Algorithmic Realisation

At first the concept of GA was used by J.D. Bagley. But especially J. Holland has found the basis for the development of GA with his research in the sixties. Holland wanted to know, how and why the evolutionary process works. He tried to find the necessary factors and to develop models, in order to explain the adjustment process to the environment. These models form the base of the proposed models from Holland [Ho92]. Besides he recognised their value for optimisation. But there is no exact definition for this class of algorithms. Moreover an algorithm belongs to the class of GA, when it contains the characteristic building stones. When those onsets are transferred from genetics and evolution to algorithms, the following factors must be given:

- A population of individuals; all individuals are different strings over an alphabet.
- Genetic operations which change the individuals.
- A function which characterises the fitness of an individual
- After several changes the population is newly ordered (reproduction) depending on the fitness of the individuals.

Reproduction causes the survival of the chromosomes with high fitness and the death of the others. Because of that the chromosomes improve from generation to generation in relation to their task.

For illustration an **artificial model duck** shall swim through a model pond. The pond is rectangular and divided into squares (Figure 3.2); the duck is in position A1. It can move one square forward from the left to the right side or one square down. These two forms of movement produce a swimming lane of the duck. The task for the duck is to swim through the pond and to catch as much feed (<>) as possible. Thereby a horizontal movement of the duck is coded by the number zero and a vertical move by one. Then a swimming route is clearly described through a sequence of zeros and ones. The duck swims as long as it reaches one of the contrary sides of square A1; that means the duck has either to reach the last column S or the last row 8. For the longest route through the pond a maximum of 25 characters is needed for the correspondig binary code.

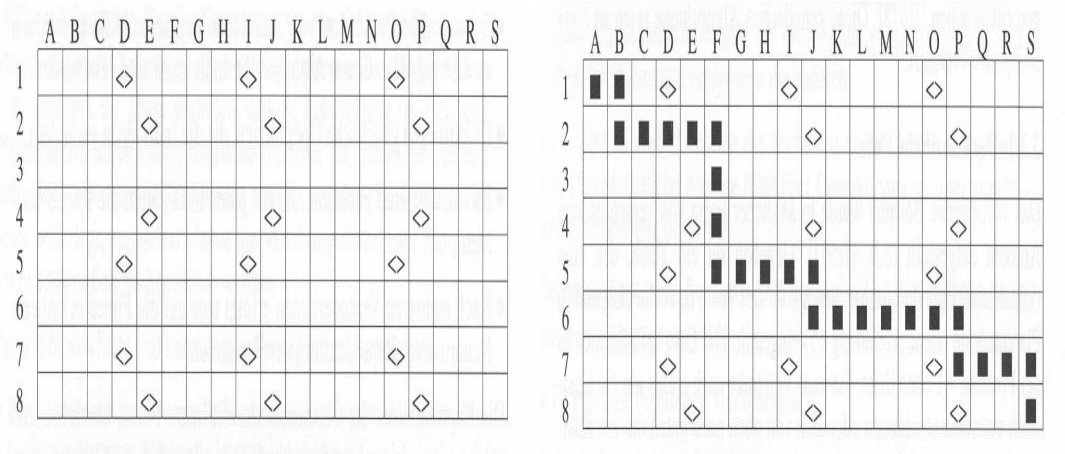

Figure 3.2: Route of the duck searching for feeding places <>

A so called **fitness function** gives each route a number. On its way through the pond the duck shall pass as many feedings places as possible; the more the duck feeds, the fitter it gets. The task is now to find routes with high fitness. In order to get an algorithm, following has to be done: by random numbers a so called **population** as set of routes is generated. Every route is adjusted to a binary sequence, which can be characterised as chromosome. The chromosomes get a certain fitness value from the fitness function; the sum of all fitness values of the population is called **total fitness**. Analogue to nature the chromosomes have to be changed in their composition, in order to get fitter ones. Two possible genetic operations are:

- **Mutation**: each element of the binary sequence is changed with a certain probability.
- **Crossover**: the binary sequences of two routes have to be cut at a certain point; then they are put together as two new routes.

After each change the chromosomes with lower fitness are deleted and replaced by those with a high fitness. This operation is called **reproduction** and causes the selection of individuals. If mutation and crossover would be the only genetic change, there would be no improvement. Thus all genetic operations have to be used to rise the average fitness of a population from generation to generation.

With this concepts the genetic algorithm itself can be described. It shall be given a set $\mathcal{D}$, which is called **search space** (see 1.3). The elements of $\mathcal{D}$ are individuals, strings or chromosomes. Each individual is a sequence of the binary values 0 or 1. All strings have the length s. A fitness function assigns each element of $\mathcal{D}$ to a real number;

$$\mathcal{H} : \mathcal{D} \to \mathbb{R} \tag{3.1}$$

The aim is to find individual x for which $\mathcal{H}(x)$ is maximal. If one tries to find the minimum, $\mathcal{H}(x)$ just has to be multiplied with -1. N individuals are produced by random numbers and compounded to a population P, which is called a **start population**. Individuals of a population can be changed by genetic operations. The concept of a genetic algorithm is based on an iteration method: for each step one or several genetic operations are executed with a certain probability. The created individuals are compounded to a new population (**generation**). The basic form of the algorithm is (1.) to select a start population. Then (2.) new individuals from the population are generated by genetic operations and compounded to a new population. (3.) If the stop criterion is not fulfilled, step (2.) is repeated.

Point 2 shall be specified: a new generation is created by one of the randomly chosen operations recombination, mutation and reproduction. The newly produced individuals are collected in the set P. This is repeated as long as P has not reached the size of a population; then P is the new population or generation. The genetic operations crossover (C), mutation (M) and reproduction (R) are selected in a probabilistic way and each operation gets a certain probability with

$$p(C) + p(M) + p(R) = 1 \tag{3.2}$$

Then a genetic algorithm is described in Table 3.1. Figure 3.3 shows the algorithm in a graphic way. The hexagonal forms refer to the so called **variation operators** (crossover and mutation), while the "rounded squares" represent the **selection operators**.

1. Choose a start population P with N individuals and define P' as empty set.

2. Calculate the fitness for all individuals of P.

3. Execute one of the operations recombination, mutation or reproduction.

4. Add the new individuals to the population P'.

5. If the number of individuals is smaller than N, continue with 3, otherwise go to 6.

6. The created individuals form a new generation P'. Test of the stop criterion. If it is not fulfilled, set $P = P'$ and continue with 2. Set $P' = 0$.

Table 3.1: Genetic algorithm

The algorithm is determined by the population size N and the frequencies $p(C)$, $p(M)$ and $p(R)$. The selection of the frequencies depends on the application, but there are some heuristic rules [Go89]:

- The population size N is mostly between 50 and some hundred.
- The recombination rate should be higher than 0,5.
- The mutation frequency should be small; it is recommended that $p(M) \leq \frac{1}{N}$.

Normally the coding of the individuals is binary; but other codings are possible. The best proceeding is to take the smallest alphabet that can represent the problem in a sufficient way. In most applications extreme values are sought which fulfil certain restrictions. When new populations are formed, following simple method fulfils the restrictions: each individual which does not fulfil the restictions gets a bad fitness value and cannot survive in the long run.

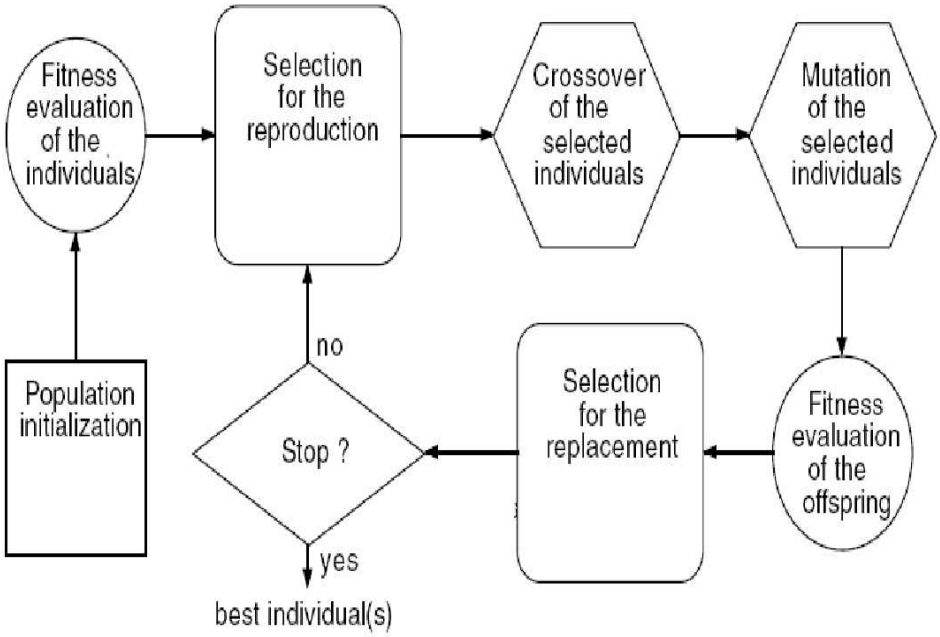

Figure 3.3: Procedure of a genetic algorithm

Decodation

Mostly the individuals are composed of binary numbers 0 and 1. But in general the optimised object is characterised by real numbers. Thus the question arises, how real numbers can be encoded by the binary sequence of chromosomes. If binary individuals shall represent real numbers from an interval $a \leq x \leq b$, then a transformation is needed. Let $z_1, z_2, \ldots, z_S$ be a binary sequence. As a dual number it stands for an integer:

$$N = \sum_{j=1}^{S} z_j \cdot 2^{j-1} \tag{3.3}$$

where S is the length of the sequence.

$$r(z_1, z_2, \ldots, z_S) = a + \frac{b-a}{2^S - 1} \cdot N \tag{3.4}$$

transforms the binary sequence $z_1, z_2, \ldots, z_n$ in a real number $r \in [a, b]$. A graphic example is given in Figure 3.4.

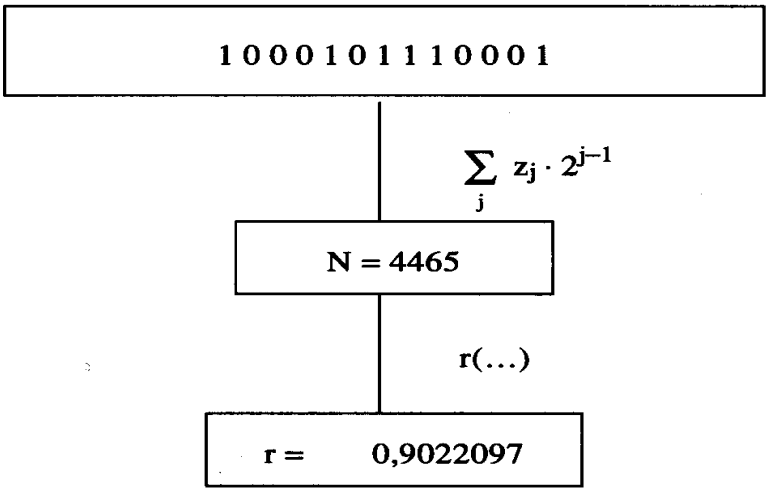

Figure 3.4: Decodation of 1000101110001 in the sequence [-10,10]

Diploid and Dominance

Most plants, nearly all animals and human beings have a double set of chromosomes in their cell. This leads to a better stability for the preservation of the populations of living beings. If all characteristics occur double, one of them has to be in the background and is called **recessive**; the active component is called **dominant**. If there are two hair colours, for example blond and black, and blond is dominant, the human being is blond. Nevertheless the black color can be inherited. Analogue to nature an algorithm can characterise one individual by two strings. Thus one string can be declared dominant and the other one recessive. From time to time there must be a crossover to exchange information. In other approaches every single bit is fixed as dominant or recessive. But most applications work with simple (haploid) chromosome sets and not with diploid ones.

Hybrid Methods

Genetic algorithms don't guarantee convergence. The populations develop in such a way that they increase their fitness; but they do not necessarily find the relevant optima. In this sense genetic algorithms can be characterised as *soft* methods.

For many applications there are conventional iteration methods that converge for good start values; thereby the convergence can be proved. But this methods mostly have the disadvantage that the start values have to be near the solution

that shall be found. Therefore it would be good to connect both methods: at first some generations are produced with GA in order to get near the optimum; then conventional methods are used to go further. An alternative would be to make one or several steps with a genetic algorithm and then succeed the calculation in a conventional way. It can be shown that such a proceeding has a good computaion time for special applications. This method is often better than the chosen conservative method and better than a pure genetic proceeding. Further on the convergence seems to be secured.

Hybrid methods correspond to an evolution which enables greater steps than simple mutations. Such a perspective can be illustrated by the evolutionary development of a spider: the net of a spider must have a minimum size; but after evolution theory only a development from a small to a less small net is possible. Normal evolution theory leads to difficulties and thus there has to be an evolutionary leap.

3.1.3 Genetic Operations

Selection Methods

Genetic algorithms change the individuals by genetic operations. In order to determine the surviving individuals, a fitness-based selection method has to be constructed. The selection method has to guarantee that principally all individuals can be selected, even those with lower fitness. Examples of different selection methods are: proportional selection, linear rank selection and (N, μ)- selection.

Proportional selection mostly chooses individuals with high fitness. The higher the fitness the higher the probability to get selected. The method is orientated at the roulette game and used very often. Let N be the number of individuals of a population and $1 \leq j \leq N$. Moreover $\mathcal{H}(j) = \mathcal{H}(x(j))$ shall be the fitness of an individual j in a population. The total fitness up to individual i is defined by:

$$F(i) := \sum_{j=1}^{i} \mathcal{H}(j) \quad \text{with} \quad 1 \leq i \leq N, \quad i, j, N \in \mathbb{N} \tag{3.5}$$

Now a random number z is generated, $1 \leq z \leq F(N)$. Then the individual i is selected, if:

$$F(i-1) < z \leq F(i) \tag{3.6}$$

The method can be illustrated graphically. In a circle each individual gets a sector whose area is proportional to the fitness. The generated random number is the equivalent of the roulette ball and decides, which sector of the circle is chosen.

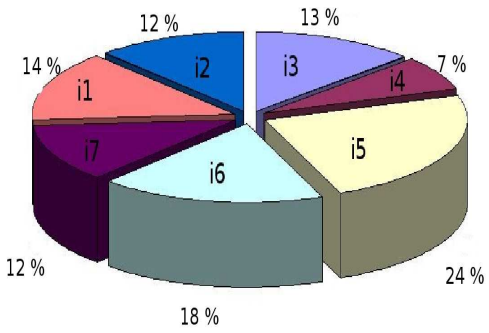

Figure 3.5: Proportional selection of an individual

Linear rank selection has another proceeding: instead of using the individual fitness directly to determine the selection frequency, the individuals are arranged due to their fitness. For a population size of N, the best individual i has rank $R(i) = 1$, the worst one j gets $R(j) = N$. The values p_{max} and p_{min} are used to determine the minimum and maximum reproduction frequencies. For $p_{max} + p_{min} = 2$ it follows:

$$p_i = \frac{1}{N}\left(p_{max} - (p_{max} - p_{min}) \cdot \frac{R(i) - 1}{N - 1}\right) \quad i = 1, \ldots, N \tag{3.7}$$

$$\sum_{i=1}^{N} p_i = 1$$

$F(i) := \sum_{j=1}^{i} p_j, 1 \leq i \leq N$ is the corresponding distribution function. Then the following selection rule is fitness orientated:

1. The individuals of a population are arranged according to descending fitness values; this means that the individuals with a high fitness are the first and those with a low fitness the last.

2. Select a random number z, $0 \leq z \leq 1$ and find the number i, for which $F(i-1) \leq F(i)$ holds.

3. The individual i is selected.

This selection method prefers individuals with high fitness. If a random number is chosen due to (2.), the corresponding number i is in average nearer to 0 than to N, what means a higher fitness.

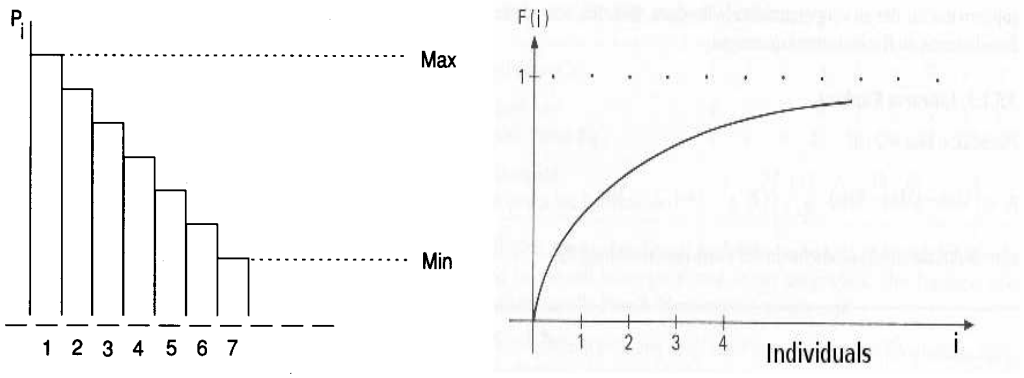

Figure 3.6: Probabilities for the linear ranking (left); distribution function (right)

In the $(\mathbf{N}, \mu)$-**selection** the best μ individuals of a population with N individuals are selected. Out of the μ best strings one is selected with the probability $p = \frac{1}{\mu}$. The method can be realised as follows:

1. Arrange the individuals of a population by fitness values in decreasing sequence; the individuals with high fitness are the first, those with low fitness the last.
2. Select a random number z with $1 \leq z \leq \mu$, $z, \mu \in \mathbb{N}$.
3. The individual z gets selected.

In contrast to the previous described selection methods, individuals with a low fitness have no chance to survive; this implies a faster convergence. But therefore the populations are forced into the nearest optima and the global optimum is often missed. Thus the survival of a bad solution is an advantage, because local optima can be left.

Recombination

The **1-point-crossover** is one of several recombination variants. Thereby two individuals are selected and called **parents**. Then a random number $z \in \mathbb{N}$ is determined which is smaller or equal to the dimension of the coding strings. Thereafter two strings are cut at position z; the parts are exchanged. The following example shows a crossover after the 7th bit:

Parent 1:	1 1 1 1 1 1 1	1 1 1 1 1
Parent 2:	0 0 0 0 0 0 0	0 0 0 0 0
Descendant 1:	1 1 1 1 1 1 1	0 0 0 0 0
Descendant 2:	0 0 0 0 0 0 0	1 1 1 1 1

The **2-point-crossover** is quite similiar: two random numbers $z_1, z_2 \in \mathbb{N}$ are determined and then a crossover between z_1 and z_2 is executed:

Parent 1:	1 1 1	1 1 1 1	1 1 1 1 1
Parent 2:	0 0 0	0 0 0 0	0 0 0 0 0
Descendant 1:	1 1 1	0 0 0 0	1 1 1 1 1
Descendant 2:	0 0 0	1 1 1 1	0 0 0 0 0

The **uniform-crossover** produces a random **template** by following rules: the template is written below the parents and the column elements above the template are exchanged, if there is a zero in the column.

Parent 1:	1 1 1 1 1 1 1 1 1 1 1 1
Parent 2:	0 0 0 0 0 0 0 0 0 0 0 0
Template:	0 1 1 0 1 0 0 1 1 1 0 1
Descendant 1:	0 1 1 0 1 0 0 1 1 1 0 1
Descendant 2:	1 0 0 1 0 1 1 0 0 0 1 0

For many applications this type of crossover is not possible, because in contrast to the 1- or 2-point-crossover characteristics of a good fitness are not preserved.

The **intermediary-crossover** is not usable for binary vectors, because the elements of chromosomes have to be real numbers. A descendant is generated by the average of the elements of the parents; odd averages are rounded up.

Parent 1:	5 7 1 9 3 6 4 6
Parent 2:	3 6 5 7 5 2 6 9
Descendant :	4 7 3 8 4 4 5 8

The **PMX crossover** (partially matched crossover) is used for applications, where the elements of an individual are unique in the corresponding string. A simple example is the TSP: a 2-point crossover sometimes would lead to new tours with one city twice and that is not allowed.

Generation n	1 4 2	3 7 6	9 5 8
	3 7 5	6 1 9	2 4 8
Generation n+1	**1** 4 2	6 **1 9**	**9** 5 8
	3 7 5	**3 7** 6	2 4 8

But a simple reorganisation compensates this problem: if there are identic numbers in one string, like number 1 in the first string of generation $n+1$, 1 is

replaced by 7 (in the crossover part of the second string). This is repeated until no number occurs more than once in a string. The result for the example is found after two steps:

1. step	**7** 4 2	6 1 9	9 5 8
	3 **1** 5	3 7 6	2 4 8
2. step	7 4 2	6 1 9	**3** 5 8
	9 1 5	3 7 6	2 4 8

Mutation & Inversion

For individuals with b bits in a population of size N, two random numbers i, k ($1 \leq i \leq N$ and $1 \leq k \leq b$) are generated and bit k of individual i is changed. For example in the following string 1101 **1** 00 the fifth bit is changed from 1 to 0: 1101 **0** 00.

The mutation frequency should be small, because mutations can destroy important informations. On the other side mutations are important for leaving local optima. Mutations can produce completely new aspects in the evolution of generations; they guarantee the irreversibility in the development of generations.

The **inversion** is another mutation operator, which reverses the sequence of bits. Randomly two numbers $k, n \leq b$ with $k < n$ are chosen; after that in any chromosome all elements between k and n are reversed:

1 1 0 1	**1 0 0 1 0**	0 1 1 0
1 1 0 1	**0 1 0 0 1**	0 1 1 0

3.1.4 Miscellaneous

Convergence

J.H. Holland as father of the genetic algorithms formulated some convergence theorems. A short introduction is given in the following. Let P be a population with N elements and P' a subset of P. Further on $\mathcal{H}(P')$ is the average fitness of all elements from P'; $\mathcal{H}(P)$ is the average fitness of the whole population P. In case of the roulette selection

$$p = \frac{\mathcal{H}(P')}{\mathcal{H}(P)} \tag{3.8}$$

is the probability that one element from P' survives the next generation.

That proves the selection of the better individuals when the generation changes. But this theorem considers neither recombination nor mutation; it just refers to the roulette selection. As conclusion there are two statements:

1. The most likely number of elements of N in the next generation is

$$N' = \frac{\mathcal{H}(P')}{\mathcal{H}(P)} \cdot N \tag{3.9}$$

2. The number of elements with a high fitness rises during the time.

In principle every chromosome has the chance to be taken into the population. That is due to mutation which can change every single bit. Above that, strings with a good fitness have a high probability to get into the population, because of the reproduction method which is orientated at high fitness values. For a high number of generation changes, the probability to reach a string with good fitness is near to one. If the number of generations G strives to infinity, for the probability p of the maximum x^* holds:

$$\lim_{G \to \infty} p(x^*) = 1 \tag{3.10}$$

Those individuals having the best fitness are reproduced more often than others; therefore they replace the worse. If the variation operators are blocked, the best individual should reproduce more quickly until its copies take over the complete population. This leads to the so called **selection pressure**. If the selection pressure is high, there is a great risk of a premature convergence; the copy of a suboptimal indivual could reproduce more quickly than others and the algorithm gets stuck in a local optimum.

Evolution Strategies - ES

Evolution strategies are orientated at principles of evolution, too: there are populations and genetic operations like mutation, crossover and selection. The creation of new generations is executed more or less in the same way as for genetic algorithms; the only difference is the algorithmic implementation. In the early seventies Rechenberg [Re73] had the idea of evolution strategies. First applications were experimental optimisations with discrete mutations: for example the optimisation of plate forms of a wind channel. Some time later similiar computer simulations were started. Meanwhile this method has been applied successfully in different cases. Some of them are:

- optimisation of optical lenses
- optimisation of socio-economic systems
- regression analysis

In spite of the identical principles, the development of evolution strategies was independant from those of genetic algorithms. The first contact between GA researchers from the US and ES researchers from Germany was in 1990.

In contrast to GA, mutation is the main operator for evolution strategies and not just a background operation. Apart from that, both methods work with the same concepts; evolution strategies just have a different concretion. A population consists of N individuals, whereby each individual is characterised by a real vector. The start population is generated by real-valued random vectors which fulfil the restrictions; strategy parameters are added to the object variables.

Although mutation is the main operator, a crossover operator is necessary for the self adaption of strategy parameters. There are two cases: **intermediary** and **discrete** crossovers. The intermediary crossover produces offspring mostly by means of components from two randomly selected parents; the discrete crossover sustains diversity. It makes sense to use the discrete crossover for object variables and the intermediary one for strategy parameters.

For genetic algorithms selection is probabilistic, because individuals have probabilities for their survival; the probabilities are derived from the individual share of total fitness. Evolution strategies have a deterministic selection concept: the μ best descendants survive. There are two selection approaches:

1. (μ, Γ)-concept: μ parents produce Γ descendants from which the μ best survive.

2. $(\mu+\Gamma)$-concept: μ parents produce Γ descendants; from the $\mu+\Gamma$ individuals the μ best survive.

Evolutionary Computing

Evolutionary computing (**EC**) characterises a group of approaches, which are in touch with evolution and genetics. There are several main directions in the area of EC that were mostly developed independently from one another. Of course there was an adjustment of ideas behind those approaches; different ideas were integrated into the particular concepts. Thus genetic algorithms, evolution strategies, evolutionary and genetic programming are summarised under the concepts of evolutionary computing.

Evolutionary programming (EP) is comparable to a genetic algorithm, but there are two important differences: firstly the data structure that shall be optimised is directly given by chromosomes and not by a binary code. Secondly there are just mutation and selection as optimisation operators; the crossover is not used (asexual reproduction).

Genetic programming is the latest approach in the research area of evolutionary computing and it is based on genetic algorithms. In general, the system to optimise is defined by real parameters. Genetic algorithms search for a set of parameters with the optimal fitness. But this concept can be generalised easily: if a parameter defined system is replaced by a theoretical construct like a computer program, one can try to find out which program solves a given task in the optimal way. In case of binary input and output values, a boolean expression has to be found. For this type of optimisation one can also try to find a solution by methods similiar to evolution: syntactically correct calculation rules are changed until an optimal solution for a given task has been found. The research area, which deals with such questions is called **genetic programming**. This concept was successfully applied to the determination of calculation rules. In principle, computer programs can be developed genetically, too: the computer learns to solve a problem without being programmed; but this was applied only to very small programms.

3.2 Ant Colony Algorithms

Ant colony algorithms are derived from the foraging of ants. In 1991 the Italian mathematician Marco Dorigo assigned the functioning of ant colonies to combinatorial optimisation problems. In their search for forage, ants orientate by means of a chemical secretion called pheromone. During their movement, ants eject this secretion for other passing ants. The higher the quantity of pheromone on a special way the higher the probability that the ant will choose it. Thus the pheromone is a kind of collective memory of the ant colony that saves the previous path decisions. Observations show that ants have roads from their nest to the feeding place. But how is it possible that pheromone enables the entire colony to find the shortest way ? Shorter ways can be traversed faster and thus more pheromone is deposited than on longer ways, because per time unit more ants pass the short way than the longer one. Therefore shorter ways get more and more attractive till the best one between nest and feeding place is found (Figure 3.7). It is the idea of an ant colony algorithm to mimic behavior of real ants with **virtual ants**. The TSP was one of the first problems for which an ant colony algorithm was implemented. Therefore the TSP shall be used to illustrate the basic ant colony algorithm.

In each iteration $t \in \mathbb{N}$, $1 \leq t \leq T$, each ant $k \in \mathbb{N}$ $(k = 1, \ldots, K)$ builds a complete path of $n = |N|$ cities. For each ant the path between city i and j depends on:

1. The list of non-visited cities W_i^k, when ant k is in city i.

2. The so called **visibility** $\nu_{ij} = \frac{1}{d_{ij}}$ with d_{ij} as the distance between two cities i and j; this information is used to influence ants to choose close cities and to avoid remote cities.

3. The **intensity** τ_{ij} of the trail is the quantity of pheromone deposited on the connection of two cities i and j.

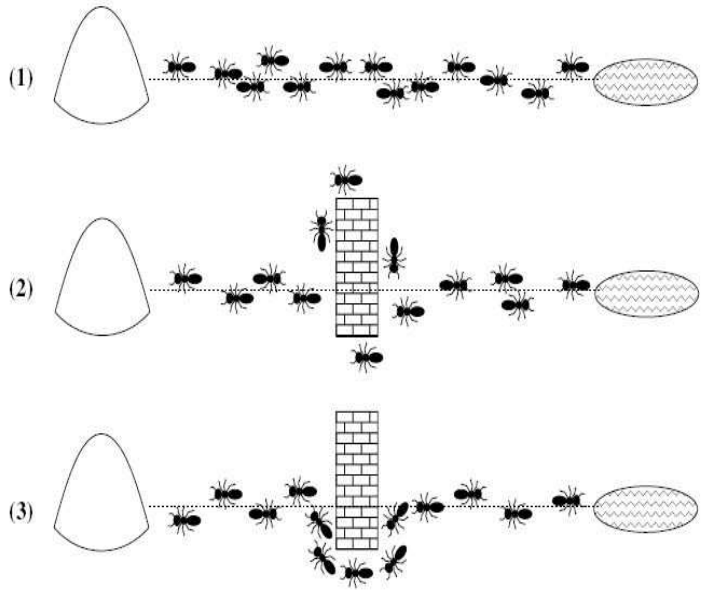

Figure 3.7: Ants finding the shortest way after blocking

For ant k the probability $P(x_{ij})$ of going from city i to j is given by following expression:

$$P^t(x_{ij}) = \frac{\tau_{ij}^{\alpha}(t) \cdot \nu_{ij}^{\beta}(t)}{\sum_{l \in W_i^k} \tau_{il}^{\alpha}(t) \cdot \nu_{il}^{\beta}(t)} \qquad \forall \qquad i = 1, \ldots, n \qquad \text{and} \qquad j \in W_i^k \quad (3.11)$$

Thereby t is the iteration of the algorithm, τ_{ij} the intensity and ν_{ij} the visibility; α and β are the parameters which determine the influence of pheromone and visibility. With $\alpha = 0$ only the visibility is taken into consideration; thus in each step the nearest city is chosen; for $\beta = 0$ only pheromone is decisive. In order to get a good solution, a compromise between these two parameters representing **diversification** and **intensification** is essential. Due to this probability distribution the variables are selected by Monte-Carlo numbers. This basic form of the algorithm has many extensions; the most important of them is called **ant colony**

optimisation (ACO). Here beside the Monte-Carlo selection another rule is applied: j is selected as the next city, if the product of pheromone and visibility is maximal. One of both rules is chosen by a random number $0 \leq z \leq 1$. If z is beneath a threshold Q, $0 \leq Q \leq 1$, Equation 3.11 holds; otherwise $P(x_{ij})$ is defined by following:

$$P(x_{ij}) = \begin{cases} 1 & \forall \quad i = 1, \ldots, n \qquad j = \max_{l \in W_i^k} \{\tau_{il}^{\alpha}(t) \cdot \nu_{il}^{\beta}(t)\} \\ 0 & \text{else} \end{cases} \tag{3.12}$$

Because of this modified selection rule the neighbourhood search of the best solution is intensified.

Beside route searching the labelling with pheromones is the most important element of the algorithm. But the implementation is slightly different to nature: at first all artificial ants of one iteration have to find their way for themselves. After that for each ant the quality of the found solution is determined and characterised with pheromone. Each ant leaves a certain quantity of pheromone $\Delta\tau_{ij}^k(t)$ on its entire course; the quantity depends on the quality of the found solution:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{\mathcal{H}^k(t)} & \text{for} (i,j) \in T^k(t) \\ 0 & \text{else} \end{cases} \tag{3.13}$$

$T^k(t)$ is the path of ant k during iteration t, $\mathcal{H}^k(t)$ the total travel length and Q a fixed parameter. Good solutions can additionally be emphasized by not allowing all artificial ants to deposit pheromones. For example just the ant with the best objective value $\mathcal{H}_{max}^k(t)$ can be chosen.

However the algorithm would not be complete without the process of **evaporation**. ρ is the so called **evaporating factor** with $0 < \rho < 1$. This factor weakens the virtual spur of old iterations in support for new ones. New solutions have more operating experience and thus the search for good solutions is enforced, contrary to randomly found solutions.

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t) \tag{3.14}$$

with $\tau_{ij}(t) = \sum_{k=1}^{K} \tau_{ij}^k(t)$ and K as the number of ants. The initial quantity $\tau \geq 0$ of pheromone on the edges is uniformly small.

Recapitulatory, ant colony algorithms primarily consist of two basic programs. One of them is responsible for all ants of the colony searching their way through the solution space. The second program simulates the way of *each* ant k through the search space. The complete algorithm is presented in Table 3.2.

For $t = 1, \ldots, T$
    1. For each ant $k = 1, \ldots, K$
        1.1 Select a city randomly
        1.2 For each non-visited city
                Select a city j from the list W_i^k
                of remaining cities
            End For
        1.3 Deposit pheromone $\Delta\tau_{ij}^k(t)$
        End For
    2. Evaporate trails
End For

Table 3.2: Basic ant colony algorithm

It is clear that an ant colony algorithm is just another heuristic: real ants do not always find the shortest way from nest to feeding ground; and so do virtual ants. The biggest danger in simulation is that the pheromone level of a single, suboptimal way is too strong. Then all virtual ants go this way and the system is trapped in a local minimum.

Beside the TSP (Dorigo et al., 1991) several other problems have been solved by ant colony algorithms: Vehicle Routing Problem (Bullnheimer et al., 1999), Quadratic Assignment Problem (Stützle und Hoss, 1999), Portfolio Selection (Maringer, 2002), JIT Sequencing Problem (McMullen, 2001) and others.

3.3 Tabu Search - TS

Tabu search is a **local search** method to solve combinatorial optimisation problems. For the solution of an optimisation problem described in section (1.3.1) a local search method presupposes a start configuration x^0 and a neighbouhood structure $\mathcal{N}$ over the solution space Z. A subsequent solution is chosen from the neighbourhood of x^0, which is the start solution for the next step. This iteration is continued until the method stops; the result is the best solution that has been found so far. The advantages of those local search methods are the simple determination of all adjacent solutions and the fast calculation of the objective function. For a suitable neighbourhood structure adjacent solutions have similiar characteristics which can be considered for the calculation of the objective function. A pseudocode of a local search method can be described for the solution

of a problem with a coherent neighbourhood structure $\mathcal{N}$ is given in Table 3.3. Possible break conditions are the reaching ...

- ... of a maximum number k_{max} of iterations
- ... of a maximum number $\bar{k}_{max}$ of iterations without an improvement of the objective function
- ... of a lower bound $\mathcal{H}^*_{min}$ for the objective function.

1. Begin with a start solution $x^0 \in Z$

   $x^* := x^0$ best solution found so far
   $\mathcal{H}^* := \mathcal{H}(x^*)$ best objective value up to now
   $k := 0$ iteration counter
   $\bar{k} := 0$ iteration counter since last objective improvement

   The counters k and $\bar{k}$ are necessary for the break conditions

2. Select $x^{k+1} \in N(x^k)$.
   If $\mathcal{H}(x^{k+1}) < \mathcal{H}^*$, then set $x^* := x^{k+1}$, $\mathcal{H}^* := \mathcal{H}(x^{k+1})$, $\bar{k} := 0$.
   Set $k := k + 1$ und $\bar{k} := \bar{k} + 1$.

3. If there is no break, go to 2.

Table 3.3: Pseudocode of local search methods

A variant of this proceeding is following method: in every iteration step the solution x^{k+1} is chosen that has the lowest objective value beneath all solutions of the neighbourhood $N(x^k)$. For $\mathcal{H}(x^{k+1}) \geq \mathcal{H}(x^k)$ this method stops; the obvious problem is that just local optima can be found. Another variant is the **local search method** with Monte-Carlo numbers. At first in every iteration step a $x^{k+1} \in N(x^k)$ is selected randomly and accepted as solution for $\mathcal{H}(x^{k+1}) < \mathcal{H}(x^k)$. In another case the new solution is accepted with a probability which gets smaller with the worsening of the objective. In difference to the Monte-Carlo method, tabu search can be formulated as deterministic method; but there is also a **probabilistic tabu search**. Deterministic tabu search was developed in 1986 from GLOVER and HANSEN. In the recent years it was successfully applied to many standard problems of OR.

The decisive idea of tabu search is to look for the best adjacent solution in every iteration and to accept it, even if it is worse than the momentary one. Because of this simple criterion a convergence to a local optimum is prevented. At first tabu search seems to accept only improvements, because in every step the neighbour with the best objective value is selected. It produces a sequence of solutions that converges against a local optimum; but after that, further solutions are selected until one of the stop criteria is fulfilled. Because in every iteration the adjacent solution does not have to be better than the momentary one, a solution can be accepted for several times and thus **cycles** of solutions can develop which prevent the finding of the global optimum. The prevention of cycles is a non-trivial problem. By restricting the neighbourhood $N(x^k)$ of a solution x^k to a subset $N'(x^k) \subseteq N(x^k)$, cycles are mostly prevented. Then the sequent solution $x^{k+1} \in N'(x^k)$ with the best objective value is chosen. The principle proceeding of tabu search is described in Table 3.4. The best solution found is x^* with the

1. Begin with a start configuration $x^0 \in Z$

   $x^* := x^0$ best solution found so far
   $\mathcal{H}^* := \mathcal{H}(x^*)$ best objective value found so far
   $k := 0$ counter iterations
   $\bar{k} := 0$ counter iterations since last objective improvement

   The counters k and $\bar{k}$ are necessary for the break conditions.

2. Determine the neighbourhood $N(x^k)$ and $N'(x^k) \subseteq N(x^k)$
   If $N'(x^k) = \emptyset$, then STOP.
   Select a sequent solution $x^{k+1} \in N'(x^k)$.

3. If $\mathcal{H}(x^{k+1}) < \mathcal{H}^*$,then set $x^* := x^{k+1}$, $\mathcal{H}^* := \mathcal{H}(x^{k+1})$, $\bar{k} := 0$.

4. If the break criterion is not fulfilled, go to 2 .
   else STOP.

Table 3.4: Pseudocode of tabu search

objective value $\mathcal{H}^* := \mathcal{H}(x^*)$. This pseucocode describes tabu search as modified local search method. The limitation of neighbours is a strategy to prevent cycles and to find fastly the optimum. The prevention of any cycle during the search is practically not possible, because for this at least n solutions have to be saved

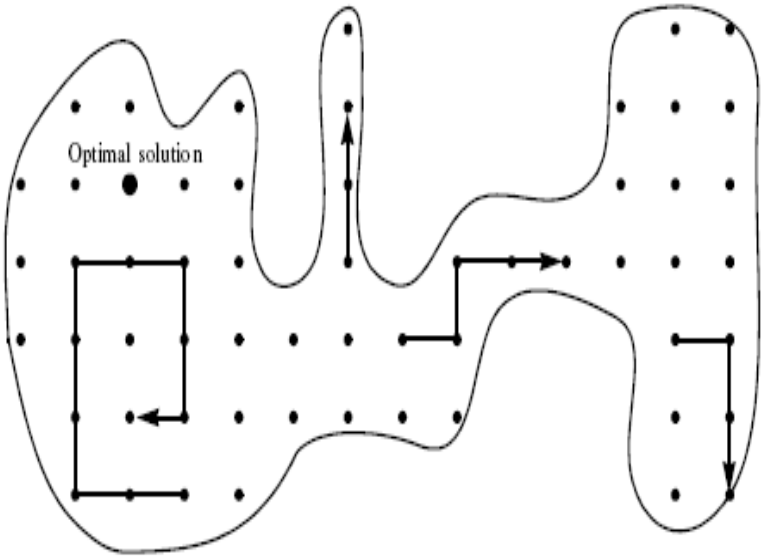

Figure 3.8: Different trajectories blocked or disconnected from the optimum

to detect a cycle with length n. If $x^i = x^{i+n} \quad \forall \quad i > n'$ holds for a sequence of solutions starting from n', there is a cycle of length n. The identification of longer cycles than n are not detected; but also shorter cycles can only be found with many additional tests in the saving list. Because this proceeding would cause to much effort, just a few cycle-restrictions are tested. For the prevention of cycles it is sufficient (but not necessary) to forbid examined solutions. For this purpose they can laid down in a so called **tabu list** T. The tabu solutions are not accepted any more in the search process. If the neighbourhood of every solution is restricted to non-tabu configurations, always new solutions are found and cycles can be prevented. Thus the sequent solution is selected from $N'(x) = N(x) - T$. This proceeding is just sufficient and not necessary, because there are trajectories in the search space, which contain a solution several times, but don't describe a cycle.

In practical applications with a solution space of several thousand solutions the tabu list can grow very large. Apart from an immense saving effort the complete list would have to be searched in every iteration, in order to ascertain the tabu-status of a solution. A reduction of saving and computation effort could be reached by a tabu list with a restricted length $|T| = k$. T is organised as queue: in every step the oldest tabu solution is deleted and the newest is added. The new solution is tabu for the next k iterations. But this proceeding has its disadvantages: a TSP for example has solutions with n cities to visit. If $k = 1000$, all tabu solutions with 100 cities each have to be compared with the momentary

configuration. Therefore even a limited tabu list leads to an enormous effort, if the problem is huge enough. Obviously the saving of already found solutions is not the best way to prevent cycles during the search process. Instead it is better to characterise the solutions by attributes.

The principle characteristic of tabu search is based on the use of mechanisms inspired by human memory. Thus it is different to simulated annealing, which is unable to learn from the past. But modelling the memory introduces multiple degrees of freedom and thus a mathematical analysis of tabu search is very difficult. Two main aspects have to be considered in the development of tabu search: firstly it is necessary to have an effective mechanism of evaluating neighboured solutions; secondly the system has to be prevented from getting trapped in a local minimum.

Chapter 4

Theory of Inventory Control

An **enterprise resource planning** system (ERP) integrates all data and processes of an organisation into a unified system. One of the central problems of enterprise resource planning is the optimisation of lot sizes in order to minimise costs of ordering and storage. Therefore inventories are an important investment for all types of firms. Sometimes huge quantities of materials are kept on stock to deal with constraints of production or to fulfil dynamic demand patterns. In this sense it is vital to have enough information to aid the management for the decision making process, in order to maximise the customer service, minimise total investment and maintain the operating efficiency. The situation turns complex because these objectives are in conflict with each other and trade-offs occur when trying to improve one of them. For maximising customer service, a relatively high investment in inventories is required, and due to capital constraints these funds could have the opportunity of better profit in some other investment. The conflict finds its solution by an efficient inventory control, levelling these trade-offs between investment and costs to find an adequate policy for the operation of the business. This principle is well known and simple in concept, but the complexity of real situations makes its application difficult. Mostly, real situations not only face a single item problem, but multiple items with several periods of replenishment.

In this chapter the basics of inventory control theory are explained. Section 4.1 deals with different kinds of storages, the main cost parameters and the most important order policies. Section 4.2 and 4.3 introduce a few established single- and multi-item models. In the last section some methods and aspects of forecasting are described, because before any inventory control system can be planned and established, it is necessary to have an estimate of the future demand.

4.1 Introduction

An **inventory** is the volitional break of the material flow. This leads to the formation of stocks and therefore inventory control needs a storage (room, building or area) to store the items. The incoming items are called **storage input**, the outgoing items **storage output**.

Figure 4.1: The elementary storage transaction

Therefore inventory control contains all activities and considers all consequences that are connected with the storage of items. There are technical / logistical aspects of inventory control, for example the storage layout, and general aspects which are related to the total stock of a company.

An important issue of inventory control is the size of the inventory. Therefore many mathematical models have been developed that are summarised under the concept of inventory control within the scope of operations research. For a supermarket the outflow is induced by customer demand and the replenishment by orders. Therefore material planning deals with the right order *quantity* (**lot size**) at the right *time*. Less order costs follow less orders; but for a higher order quantity the **storage costs** rise. The advantage of a great inventory is that there is a high level of service and most customer requirements are fulfilled. Short term inventory problems are those, which deal with order / storage costs and service level. Problems of long term inventory control do not belong to this issue, because in the long run system parameters can be changed, e.g. the storage size.
The situation is similiar with **intermediate storages**. Nevertheles those are strongly bound to production and thus there is no standard inventory problem. But the results of inventory control theory can be used for material planning of intermediate storages.

The models of inventory control are applied to retail and industry inventories. Subsequent to industrial inventories there is a system of distribution. The ma-

terial planning of such hierarchical systems is in the domain of **multi-echelon inventory control**; that is an extension of inventory control theory. The problems of inventory control are characterised by the following:

1. *Several items* are managed in a single stock; this means that order handling and storage occur together.

2. *Demand* and *delivery time* are often stochastic or not known.

3. Not only the costs of material planning have to be considered, but also *non-monetary* and *non-quantitative* aspects.

For problems of this type inventory control has developed a mass of models. The characteristic feature is either stochastic-deterministic or stationary-dynamic.

The classification of stochastic and deterministic models marks two totally different directions of research. Especially in the sixties of the twentieth century the main focus was on stochastic models. The underlying theory was named **AHM-theory**, labelled after the authors (Arrow, Harris, Marschak).

For the determination of cost parameters only those costs have to be considered which affect the order date and quantity. As mentioned there are three different cost parameters:

1. **Order costs** are caused by an order transaction. This transaction comprehends all activities from the triggering of an order (storage determination, supplier selection, etc.) to the storage and the payment of the bill. Some of the order costs depend on the order quantity, e.g. discounts on acquisition prices. Others depend on the order transaction, for example mass independent costs of transport or quality control; one speaks of fixed costs. These costs can be indirect or direct; contrary to indirect costs, direct costs can easily be assigned to their causation. Fixed order costs are mostly indirect costs, because the order handling is carried out collectively for all items.

2. **Storage costs**: Analogue to order costs there is a classification in direct and indirect costs. Direct costs are interest charges of bound assets in storage, taxes, insurance or costs by damage, loss and ageing. Indirect costs are personnel and leasing costs, amortisations, etc.

3. **Shortage costs** arise, when the inventory is not ready for delivery. Direct shortage costs for example are additional costs for an express delivery or penalties, if the items are not delivered in time.

But how to determine those cost parameters ? Apparently there is no problem with direct costs. But indirect costs can not be allocated to their origin; thus there

are restrictions in a model. The standard models do not include this restrictions, but consider them by opportunity costs. The cost parameters will be fixed in such a way that the optimum order policies for single items do not disturb the restrictions.

In spite of great efforts, these models have not been used in practice. Mainly there are two reasons for this:

1. In order to use stochastic models, the stochastic processes have to be identified. This is a very elaborate task, because often there are thousands of items in a single stock.

2. In reality the demand processes are frequently instationary and high correlated. Thus the optimum cannot be determined with justifiable costs.

Real storage problems are multi-item problems and can not be described by a deterministic demand. Aside this, the cost parameters have to be fixed and several criteria determined to find the correct order policy. In practice it is assumed that the considered item is managed by a special rule: if the inventory $y \geq 0$ falls below order point s, the stock is filled up to S; otherwise nothing is ordered:

$$q = \begin{cases} S - y & \text{for} \quad y < s \\ 0 & \text{for} \quad y \geq s \end{cases} \tag{4.1}$$

The order point s is calculated by the forecasted demand during delivery time d plus safety stock SB. Hereby the temporal variability of delivery time can easily be considered. Also a dependency on quantity can be regarded approximately. For a stochastic delivery time the current estimate has to be used. This proceeding is called **re-order level** or $(\mathbf{s}, \mathbf{S})$-policy; another re-order level policy is received, when the stock balances are maintained with a fixed replenishment order size:

$$q = \begin{cases} Q & \text{for} \quad y < s \\ 0 & \text{for} \quad y \geq s \end{cases} \tag{4.2}$$

Technically the re-order level policies require a continuous review of stocks; therefore it tends to have much lower stocks than the **re-order cycle** policies, where replenishments are made on regular times. A re-order cycle policy can be easier planned and allows the stockholder to order many items from a single supplier; thus he can reduce replenishment costs considerably compared with the re-order level policies.

Multi-item-models are rather complex and many practical stock problems are determined by organisational conditions. In practice therefore mostly **single-item-models** are used [Fr07]. However, in theory there are a few models dealing with multi-item inventories. Thereby a basic classification is drawn between so

called **single-echelon** inventories with several items on *one* level and **multi-echelon** inventories with several items on different levels. This dissertation does not refer to multi-echelon inventories, but it applies to the holding of multi-item and single-echelon inventories where the demand is at the item or service level. That does exclude the sector of the manufacturing industry based on the production of complex units or assemblies (e.g. automotive production), where the demand at the component level is directly related to production of units or assemblies at the top level; but it does not exclude the sector of manufacturing where the end product is relatively simple. Examples of inventory control situations where the ordering of replenishments to re-stock inventories is controlled by parameters determined by forecasts of demand and costs of inventory operations are:

- Supermarkets
- Gas and water utilities
- Stockists, wholesalers and distribution

4.2 Single-Item-Models

Multi-item-models are rather complex and many practical stock problems are determined by organisational conditions. In practice therefore only single-item-models are used. An overview is given in Figure 4.2

4.2.1 Deterministic Models

Normally the demand is not known. In case of standard models therefore it is necessary to substitute the real existing process of demand by a row of forecasts $d(t+\tau), \tau = 0, 1, 2 \ldots$. Here $d(t+\tau)$ means the forecast of demand in period $t+\tau$; t is the current period. Those forecasts will be repeated as often as necessary, in order to minimise errors. Thus standard models are used in a continuous way: after every forecast the order policies are calculated anew. But not every forecast error can be averted; therefore the safety stock SB_t is necessary.

For static time series the methods of moving average and exponential smoothing of 1. order are available. If there is a trend, then exponential smoothing 2. order or linear regression analysis can be used. For seasonal fluctuation, one can avail the forecast method of Winters.

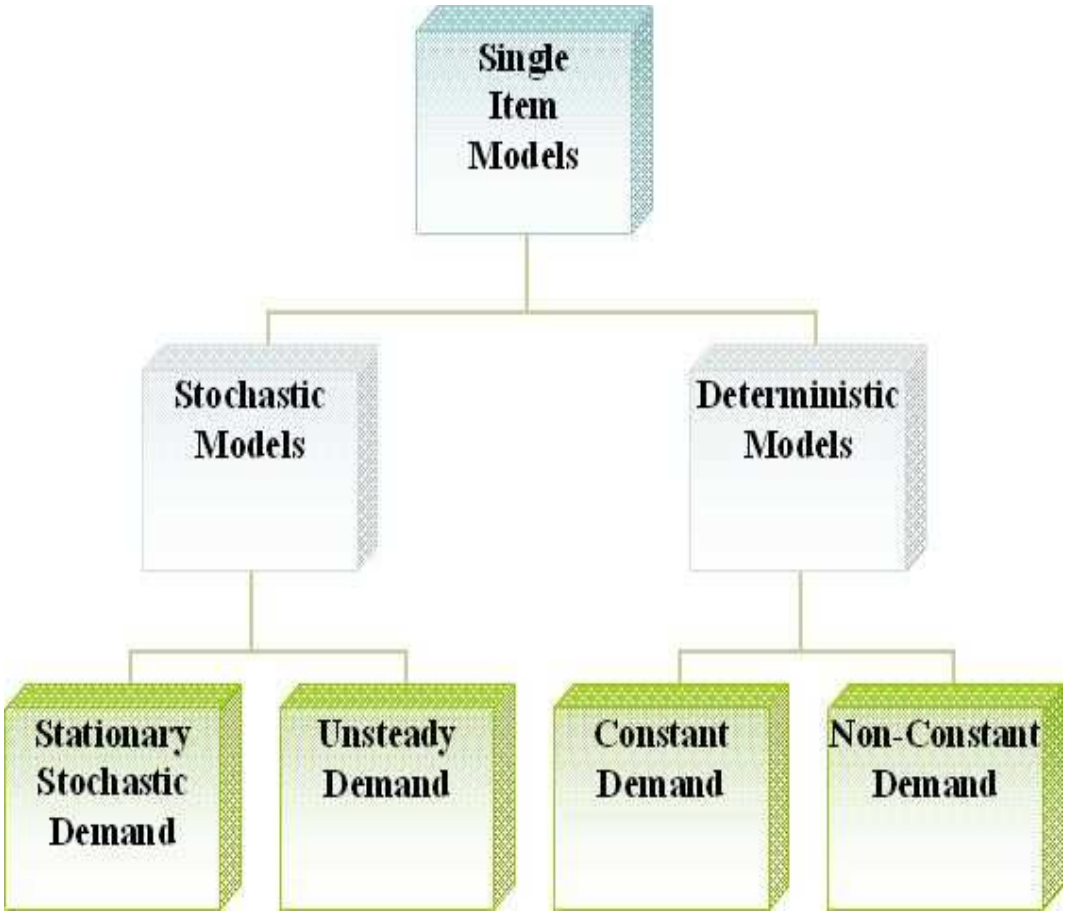

Figure 4.2: Scheme of single-item-models

Classical Lot Size Model

Instead of the classical lot size model one speaks of the Andler-, Harris- or Wilson-model. Precondition is the assumption that demand is constant and continuous. Moreover there is no delivery time and no shortage. The problem is to determine the best order size q and the best length of the order interval T in such a way that the sum of order and storage costs is minimal. The relevant order costs $B(q)$ are fix costs that arise by ordering.

$$B(q) \quad = \quad \begin{cases} c_B & \forall \quad q > 0 \\ 0 & \forall \quad q = 0 \end{cases} \tag{4.3}$$

Storage costs $L(q)$ of the order cycle T are

$$L(q) = T\frac{q}{2}c_L \tag{4.4}$$

with $\frac{q}{2}$ as *average inventory* and c_L as costs per unit and time intervall. In order to determine the optimum order policy, following procedure is applied: the average total costs C per time unit are

$$\begin{aligned} C &= \frac{1}{T}(c_B + T\frac{q}{2}c_L) \\ &= \frac{c_B}{T} + \frac{q}{2}c_L \\ &= \frac{d}{q}c_B + q\frac{c_L}{2} \end{aligned} \tag{4.5}$$

with $T = \frac{q}{d}$ and d as demand rate. The average order costs $\frac{c_B}{T} = \frac{d}{q}c_B$ and the average storage costs $\frac{L(q)}{T} = \frac{q}{2}c_L$ are illustrated in Figure 4.3:

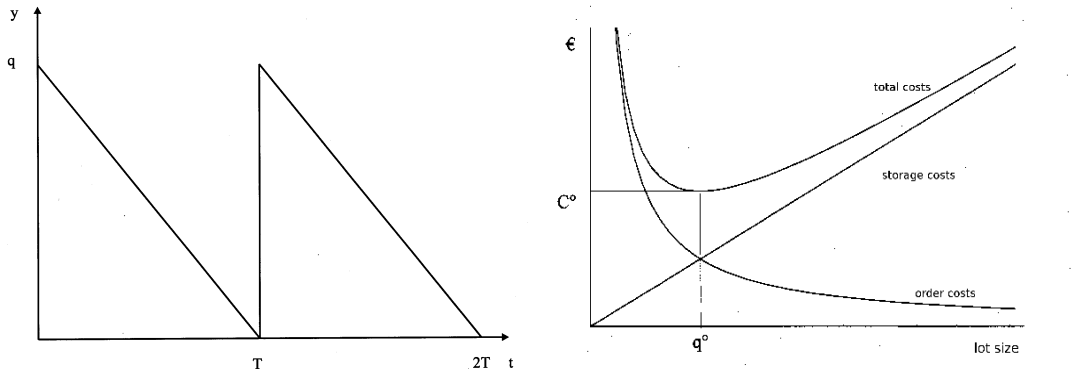

Figure 4.3: Classical lot size model; Left: inventory process. Right: different costs.

While the average storage costs rise linearly with the order size, the order costs diminish in a hyperbolic way. There is a minimum of total costs for a certain order quantity. The minimum can be calculated by differentiating 4.5 after q:

$$\begin{aligned} \frac{\partial C}{\partial q} &= -\frac{d}{q^2}c_B + \frac{c_L}{2} = 0 \\ &\Longrightarrow q^* = \sqrt{\frac{2dc_B}{c_L}} \end{aligned} \tag{4.6}$$

q^* is the classical lot size. For the optimal length of cycle T^* and the optimal average costs following holds:

$$T^* = \frac{q^*}{r} = \sqrt{\frac{2c_B}{dc_L}} \tag{4.7}$$

$$C^* = \sqrt{2dc_Bc_L} \tag{4.8}$$

The classical lot size model can easily consider many restrictions, for example delivery dates, continuous inflow, shortages and similiar things. In this way the lot size model is a special case of many complex models.

Wagner-Whitin-Model

The Wagner-Whitin-model is marked by a row of specialisations of the general deterministic system. It is defined by the deterministic dynamic decision problem of Table 4.1

(1)	y_t:	inventory at the beginning of period t; $t = 0, 1, \ldots, N$
	$Y_t = y_t : y_t \geq 0$	state domain
	y_0:	initial inventory
		inventories can take every positive value; that means there is neither a restriction on storage capacity nor shortfalls.
(2)	q_t:	order in period t; $t = 0, 1, \ldots, N-1$
		orders can take every positive value. size restrictions and quantisations do not exist; but the orders have to avert shortfalls.
(3)	d_t:	Demand in the intervall of inspection $t, t+1$; $t = 0, 1, \ldots, N-1$
(4)	$y_{t+1} = y_t + q_t - d_t$:	equation of stock balance
(5)	Cost Criterion:	min $C = \sum_{t=0}^{N-1}(B(q_t) + L(y_{t+1}))$
	Cost of Ordering:	$B(q_t) = c_B(q_t) = c_B$ for $q_t \neq 0$ $B(q_t) = c_B(q_t) = 0$ else
	Cost of Storage:	$L(y_t) = c_L(y_t - \frac{d_t}{2})$

Table 4.1: Deterministic decision problem

In this model delivery time was omitted. But that is no constraint; it only serves the simplification of notation. There are only fixed costs for orders. Quantity based costs are not considered, because they don't influence the moment of ordering or the order size. Quantity dependent, non-proportional costs are not included in the cost criterion above. Storage costs evaluate an average inventory with a storage cost rate of c_L. The special model structure implies two essential simplifications for the determination of the optimal order policy:

1. If there is an empty stock or the inventory is fallen to a minimum level, an order will be placed; otherwise there would be unnecessary storage costs.

2. The consolidated demand of future periods will be ordered; otherwise there would be unnecessary storage costs as well.

These evident conditions to an optimal policy lead to a basic restriction of different policies. The saturation of conditions 1 and 2 leads to the so called Wagner-Whitin-algorithm.

At first the periods 0 and 1, then $0, 1, 2$, then $0, 1, 2, 3$, and so on, are optimised; thereby the results of the last optimisation (with one period less) are used. Because of that just a fraction of different policies has to be considered.

Heuristic Methods

Although the Wagner-Whitin-model is a very efficient algorithm, several methods have been developed as approximation to this model. Contrary to Wagner-Whitin those heuristic algorithms don't consider the whole planning period. Therefore the computation time and the solution quality are lower. The following methods check, whether the demand of a period can be satisfied by the last order or whether a new order has to be dismissed. The first order is determined by the first period with a demand. The next order quantities are satisfied by one order as long as a special criterion is fulfilled; otherwise a new order is dismissed.

Least-Unit-Cost: The order quantity in period t is increased with future material requirements as long as the average costs per quantity unit can be reduced. If there is an order in period τ and the demand is covered up to period j with $j > \tau$, the average costs are defined by:

$$c_{\tau j}^{unit} = \frac{c_B + c_L \sum_{t=\tau+1}^{j} (t-\tau) d_t}{\sum_{t=\tau}^{j} d_t} \tag{4.9}$$

Thus those order quantity of period τ has to be determined that leads to a minimum of equation 4.9. The decision problem of period τ then can be formulated

as:

$$\max\{j|c_{\tau j}^{unit} < c_{\tau j-1}^{unit}\} \tag{4.10}$$

Thus the highest j has to be found, which fulfils condition $c_{\tau j}^{unit} < c_{\tau j-1}^{unit}$. In other words: that period has to be found, whose demand can be satisfied by the order in period τ without an increase in average costs.

Part-Period-Balancing: The key-note of this heuristic is that an order reaches for as many periods as the storage costs are equal to the order fix costs:

$$c_L \sum_{t=\tau+1}^{j^*} (t-\tau)d_t \leq c_B \tag{4.11}$$

where j^* is the period up to which the order reaches.

Silver-Meal: In the style of the classical lot size model the Silver-Meal method tries to minimise costs per time unit or period. If there is an order in period τ which covers the material requirement up to period $j, j > \tau$, following costs have to be considered:

$$c_{\tau j}^{period} = \frac{c_B + c_L \sum_{t=\tau+1}^{j}(t-\tau)d_t}{j-\tau+1} \tag{4.12}$$

In period τ those j is sought, which fulfils following condition:

$$\max\{j|c_{\tau j}^{period} < c_{\tau,j-1}^{period}\} \tag{4.13}$$

This means that the costs per period shall be minimised.

Other Algorithms: Further on **Groff** and **Savings** are two methods for determining order time and quantity. They work similiar to the previous algorithms; therefore they shall not be presented here. For further reading [Te03] is recommended.

4.2.2 Stochastic Models

The optimal policies of the classical lot size model and the Wagner-Whitin-model depend on exact information about the demand. But those informations are based on insecure forecasting. The decisive tasks of stochastic lot size models are to clarify the problem structure and the foundation of material planning methods.

The Newsboy-Problem

This problem is a single period model with uncertain demand. The objective function is defined as follows:

$$\min \quad h\sum_{u=0}^{x}(x-u)p_u + g\sum_{u=x+1}^{\infty}(u-x)p_u \tag{4.14}$$

with

x	stock of newspapers
u	number of sold newspapers
p_u	probability of u sold newspapers
$\phi(u)$	density function of the number of sold newspapers
μ	expectation value of u
h	loss per non-sold newspaper
g	loss per missing newspaper ($g > h$).

Table 4.2: Parameters of the Newsboy problem

The optimal value of x is those with the minimal loss. The solution can be found by a changeover to a continuous loss function:

$$\begin{aligned}
\min \quad & h\int_0^x (x-u)\phi(u)du + g\int_x^\infty (u-x)\phi(u)du \\
= \; & h\int_0^x (x-u)\phi(u)du + g\int_0^\infty (u-x)\phi(u)du - g\int_0^x (u-x)\phi(u)du \\
= \; & (h+g)\int_0^x (x-u)\phi(u)du + g(\mu - x)
\end{aligned} \tag{4.15}$$

With partial integration and $\Phi(u) = \int_0^u \phi(z)dz$ as distribution function of u one receives:

$$\min \quad (h+g)\int_0^x \Phi(u)du + g(\mu - x) \tag{4.16}$$

After derivation of x the restriction for the optimum is received:

$$(h+g)\Phi(x) - g = 0 \quad \Leftrightarrow \quad x = \Phi^{-1}(\frac{g}{h+g}) \tag{4.17}$$

Hadley-Whitin-Model

Stochastic models are characterised by the fact that shortages can not be avoided; thus shortages have to be evaluated with costs. Analogue to the storage cost rate c_L, there shall be a shortage cost rate π . Like the classical lot size model it shall be continuous. The probability density $\Phi(r)$ is the demand at a certain moment. The time of delivery λ shall be deterministic and constant; and there shall be just one order.

Generally it can be shown in the context of AHM-theory [Ho69] that the optimisation of expectation costs of such a model leads to a (s,S)-policy:

$$q = \begin{cases} 0 & \text{if} \quad y > s \\ S - y & \text{if} \quad y \leq s \end{cases} \tag{4.18}$$

whereas y is the disposable stock. Furthermore it is assumed: 1. Because of the computer support of inventory management every demand immediately leads to an increase of stock (thus the so called inspections interval is zero); an order can be placed any time. Then following holds:

$$q = \begin{cases} 0 & \text{if} \quad y > s \\ Q & \text{if} \quad y \leq s \end{cases} \tag{4.19}$$

This is a so called (s, Q)-model: as soon as the disposable stock level falls under the order point s, an order with size Q is placed. It is the task of following considerations to determine values $s = s^*$ and $Q = Q^*$, which minimise the average of the expectation costs over a period of several order cycles.

The *average annual order costs* are $B = \frac{\mu}{Q} c_B$. Here μ is the average annual demand and therefore $\frac{\mu}{Q}$ the average annual orders. The *average annual storage costs* L are determined as follows: the minimum inventory level SB is defined by the stock that has to be available, when the new order arrives. The average annual inventory is therefore

$$\frac{Q}{2} + SB = \frac{Q}{2} + s - \mu^{\lambda}$$

with μ^{λ} as the average outflow during time of delivery. So the average annual storage costs are:

$$L = (\frac{Q}{2} + s - \mu^{\lambda}) c_L \tag{4.20}$$

The average annual shortage costs can be received by calculation of the average shortage costs per cycle, which then have to be multiplied with the average annual number of cycles $\frac{\mu}{Q}$. If x is the accumulated (stochastic) demand during time of

delivery, there is a shortage of $x - s$ for $x > s$. The average shortage per cycle is therefore:

$$\eta := \int_s^\infty (x-s)\phi^\lambda(x)dx = \int_s^\infty x\phi^\lambda(x)dx - sH(s) \tag{4.21}$$

with $\phi^\lambda(x)$ as probability density function for the accumulated demand during time of delivery and $H(s)$ as related distribution function. Therewith the average shortage costs are per year:

$$F = \frac{\mu}{Q}\eta(s)\pi$$

Finally for the total annual average costs holds:

$$C = \frac{\mu}{Q}c_B + (\frac{Q}{2} + s - \mu^\lambda)c_L + \frac{\mu}{Q}\eta(s)\pi \tag{4.22}$$

The optimum values of Q and s are determined as follows:

$$\frac{dC}{dQ} = -\frac{\mu}{Q^2}c_B + (\frac{1}{2}c_L - \frac{\mu}{Q^2})\eta(s)\pi = 0 \tag{4.23}$$

$$\frac{dC}{ds} = c_L + \frac{\mu}{q}\pi(-s\phi^\lambda(s) + s\phi^\lambda(s) - H(s)) = 0 \tag{4.24}$$

$$\tag{4.25}$$

And with numerical methods Q^* and s^* are calculated.

4.3 Multi-Item-Inventories

4.3.1 Flaccidities of Single-Item-Models

Up to the middle of the sixties theory of inventory control just dealt with single-item models and not with several items in one stock. A relationship between items is considered by restrictions effecting all items; e.g. all items compete for a fixed budget, limited stockroom or restricted order quantity. Thereby the reciprocal dependencies because of complementary and substitutional properties were neglected. Those relations influence the demand of a single item and the total demand of all items.

A further aspect to consider in the multi-item case are the cost savings that result from a collective order of several items. Single-item models deal with this in two ways: either the order is postponed until a certain, cost-saving lot size is reached or there is a fixed order point for all items. The results do not satisfy, because such methods don't consider the specialties of individual items.

All those objections against an isolated approach are amplified, if the stock is replenished by own production: the relations between items have a greater

influence on production costs than on the costs of external procurement. The independent treatment of connected items is used because of the complex and dynamic reality of inventory control. Thus single item solutions are used as first approximation and have to be improved further.

In some situations this treatment may lead to a useful solution, for example in a homogeneous assortment of goods; and also in special complex systems such a treatment may be necessary. But two grave exceptions cannot be ignored: at first the policy of single-item models constitute the frame of an optimal global policy, which shall be determined empirically; but the one dimensional consideration rarely provides an overview about all possible policies of the total problem. Secondly the single solutions can be far away from the global optimum.

4.3.2 Multi-Item-Models

The models of multi-item inventories deal with several items at one time. They can be described by following objective function:

$$\mathcal{H} = \min \quad \sum_{i=1}^{M} \sum_{t=1}^{T} c_B^i b_{it} + c_L^i x_{it} \tag{4.26}$$

under the restrictions:

$$\begin{array}{cll} y_{i,t-1} + q_{it} - x_{it} & = d_{it} & \text{(a)} \\ q_{it} - K b_{it} & \leq 0 & \text{(b)} \\ q_{it} & \geq 0 & \text{(c)} \\ x_{it} & \geq 0 & \text{(d)} \\ b_{it} & = \{0, 1\} & \text{(e)} \\ \sum_{i=1}^{M} x_{it} & \leq C & \text{(f)} \end{array}$$

$$i = 1, \ldots, M; \quad t = 1, \ldots, T$$

with the variables:

T	Length of the planning period
M	Number of items
d_{it}	Net material requirement of item i in period t
c_L	Storage fee
K	Large number
q_{it}	Lot size of item i in period t
b_{it}	Binary variable
x_{it}	Inventory at the end of period t
C	Maximum storage capacity

Table 4.3: Variables of an Multi-Item-Inventory

The objective function 4.26 consists of order and storage costs for each item. The binary variables b_{it} have the value 1, if the lot size q_{it} is higher than one. This is realised by restriction (b) in connection with the objective function: the binary variable x_{it} must be 1, if the lot size q_{it} is larger than zero. Thereby K is a large number which has to be higher than the maximum lot size. Equation (a) states a connection between the demand of a period, the stock at the beginning and the end of a period and the inward stock movement. Equation (f) is responsible for preventing an overrun of inventory capacity. A way to solve this problem are methods of mathematical optimisation like the classical lot size model.

Classical Lot Size Model

A stock of several items $i = 1, \ldots, M$ with *deterministic* demand d_i can be optimised with the classical lot size model. Further assumptions are: (1) delivery without delay and (2) restricted stockroom. Item i needs a space of b_i; C is the upper limit of the average inventory. The objective function is:

$$\begin{aligned} \min \qquad \mathcal{H} &= \sum_{i=1}^{M} \Big(\frac{1}{2} q_i c_L + \frac{c_B^i d_i}{q_i} \Big) \\ \text{under the restriction} \qquad & \sum_{i=1}^{M} \frac{1}{2} q_i b_i \leq C \end{aligned} \tag{4.27}$$

with the Lagrange-function

$$\Lambda = \sum_{i=1}^{M} \Big(\frac{1}{2} q_i c_L^i + \frac{c_B^i d_i}{q_i} \Big) + \lambda \Big(\sum_{i=1}^{M} \frac{1}{2} q_i b_i - C \Big) \tag{4.28}$$

and following restriction for the transfer prices λ

$$\lambda \quad \left\{ \begin{array}{ll} = 0 & \text{for} \quad C > \\ > 0 & \text{for} \quad C = \end{array} \right\} \sum_{i=1}^{M} \frac{1}{2} q_i b_i \tag{4.29}$$

The optimum lot size of every item is received by setting the first derivation to zero

$$\frac{\partial \Lambda}{\partial q_i} = \frac{c_L^i}{2} - \frac{c_B^i}{{q_i}^2} + \lambda b_i = 0 \tag{4.30}$$

$$\Longrightarrow {q_i}^* = \sqrt{\frac{2 c_B^i d_i}{c_L^i + \lambda b_i}} \tag{4.31}$$

The multi-item inventory is a difficult problem in **combinatorial optimisation**. Because there are no relevant solution methods in practice, many heuristics have been developed. One of them is the so called Dixon-model.

Dixon - Model

In this model a capacity restriction is integrated. It is based on the Silver-Meal heuristic (section 4.2.1), which tries to summarise the orders as long as the average costs per period are minimal. Because many items compete for the restricted storage capacity, there is no guarantee that each lot is ordered in the period with the minimal average costs. Thus it can be necessary that several items are ordered earlier, in order to get a valid plan without overcharging the inventory.

The solution quality depends on the sequence of considered alternatives. Therefore a rule is necessary to determine the sequence in which the items are dealt with. Dixon calculates **priority numbers** from the known average costs of the Silver-Meal-heuristic:

$$p_{\tau i} = \frac{c_{\tau j}^{period} - c_{\tau, j+1}^{period}}{d_{i,j+1}} \tag{4.32}$$

The numerator expression describes the increase of average costs per period, if the order quantity for item i in period τ is enlarged by the demand $d_{i,j+1}$. The denominator expresses the raised usage of the inventory. The connection of both variables ($p_{\tau i}$) describes the marginal increase in costs per additional used capacity unit. For $p_{\tau i} \leq 0$ the costs rise because of the additional order size.

The basic proceeding of this method is following: at first the order quantities of all items in period $\tau = 1$ are fixed; then the order quantities in period $\tau = 2$, etc. The determination of order quantities is carried out similiar to the Silver-Meal method. The sequence of single items is stated by the priority numbers

of Equation 4.32. As long as the average costs are decreasing and the storage capacity is not depleted, the order quantities will rise. If the space of ordered items exceeds the storage capacity, some orders have to be delayed.

The initial situation is that the material requirement of period 1 is ordered in period 1; then the remaining capacity of this period is calculated. Next it has to be checked, whether there are capacity shortages in the planning period. After that the ordered quantity in period 1 is enlarged. For this the item with highest priority number is chosen; then the Silver-Meal criterion decides, whether it is profitable to rise the lot size. This is repeated until the storage capacity is depleted or all priority numbers are $p_{\tau i} \leq 0$. After that another test is made, whether there is a storage overload in the following periods.

If the sum of the later material requirements (from period t_c on) is higher than the available storage capacity, the required capacity C^* is calculated. Afterwards those items are considered, whose order quantities in period τ don't cover the material requirements till period t_c. In a test the order size is enlarged for one period or more and the associated costs are calculated. The order size is enlarged for item i with the smallest increase of costs. This procedure is repeated as long as $C^* \geq 0$.

4.4 Forecasting

Forecasting is a necessary pre-requisite to all inventory control situations. Without an estimate of the future customer demand, it is impossible to plan the levels of inventories that will be required to offer customers a reasonable level of service.

In general terms, forecasting at all levels from long term to short term can be interpreted as being a deterministic process of estimating a future event by casting forward past data. In all these forecasting processes, past data are initially analysed to establish the basic level of demand (the stationary element) and any underlying trends (such as growth and seasonality) which characterise the data. This information is then used in a predetermined way to obtain an estimate of the future. Thus forecasting processes are usually largely computer-automated. In contrast to forecasting, prediction is generally interpreted as a process of estimating a future event based primarily on subjective considerations; therefore it is not automated but based on manual methods. The forecast is calculated on assumptions that characteristic trends (identified in past demand data) will continue into the future. Therefore an automatically produced forecast should always be open to alteration, if predictions (e.g. changes in market conditions) appear to suggest that such assumptions could be invalid. Because predictions are predominantly subjective and involve manual interruption, they are generally far more expensive to implement on a routine basis than forecasts. For many

items involved it is thus normally more effective to operate on the assumption that scientifically produced forecasts are assumed to be satisfactory unless and until a monitoring procedure indicates that the forecast for a particular item is no longer in control. In order to control forecasts, several effective monitoring systems are available.

4.4.1 Different Types of Forecasting Methods

A useful way of classifying demand forecasting methods is to define the type of forecast on the basis of the time period associated with the demand data which are being analysed, as illustrated in Table 4.4.

Category	**Time Period**	**Example of Application**	**Forecasting Techniques**
Immediate Term	1/4 day to 1 day	Electricity demand forecasting	various
Short term	1 week to 1 month	Demand forecasting in industry and commerce	Simple exponentially weighted averages and derivatives for growth and seasonal trends
Medium Term	1 month to 1 year	Sales and financial forecasting	Regression, time series analysis
		Econometric forecasting	Multi parameter models
Long term	1 year to 1 decade	Technological forecasting	DELPHI think tanks

Table 4.4: Types of demand forecast based on underlying time unit

Although there is no strict demarcation between the various types of forecasting categorised within Table 4.4, it is generally assumed that short time forecasting methods are most suitable in situations where there are many components or item lines as typically does occur in an inventory control environment. Whithin such an environment it is also often true that the demand patterns being analysed are relatively fast moving. The forecasting models used when operating in such an environment are therefore necessarily required to be simple and relatively cheap to operate while still being robust.

Inventory control systems are required to cope with a variety of different customer demand patterns for which forecasts are necessary, if an effective overall

policy for controlling inventory is to be achieved. In practice it is assumed that the following demand patterns can exist.

Stationary Demand

This assumes that although customer demand per time unit fluctuates, there is no underlying growth or seasonal trend. The left part of Figure 4.4 illustrates the basic stationary character of such data but also identifies the fact that variability in demand exists.

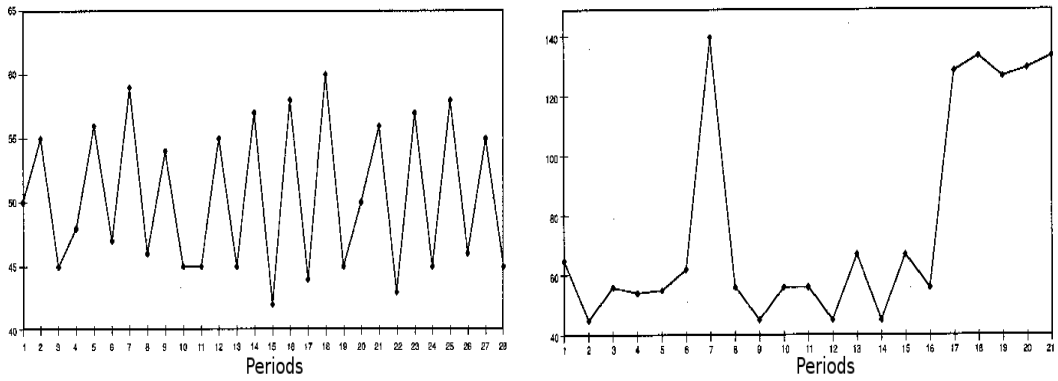

Figure 4.4: Stationary demand patterns

Because no growth or seasonality are assumed in stationary demand patterns, forecasts ahead are fixed in value and the forecast for one period ahead is the forecast for any number of periods ahead. But it should be accepted that occasionally fundamental changes in the demand pattern may occur, but these are assumed to be short-term in nature, such as:

- **Impulses** - individual demands which are significantly higher or lower than normal. Such impulses are best ignored by a forecasting system linked to an inventory control policy, since such policies are basically designed to cope with a reasonably level of demand with a known, measurable degree of variation.

- **Step changes** - a series of successive demands which are significantly higher or lower than normal which in effect produces two stationary demand situations: one before the step change followed by another stationary situation at a different level subsequent to the step change.

The ideal response of a forecast to a step change in demand is that it should react as quickly as possible in adapting to the post step change level of demand. Should this not be feasible, a competent forecasting system should at least identify

that such a step change has occurred and should also instigate remedial action to ensure that the forecast, which will naturally lag behind such a sudden change of level, is corrected. Unlike an impulse, a step change is sustained beyond the period of the initial increase/decrease in demand.

The right part of Figure 4.4 illustrates a demand pattern where a single period impulse (a significant, high demand occurring for one period only) is followed by a positive step in demand (a succession of significantly high values). The stationary demand pattern is the simplest type of demand characteristic to analyse. However, more complex demand patterns do occur as can be evidenced by plotting demand values against time to demonstrate trends in either growth/decline or seasonality.

Demand with Growth and Seasonal Characteristics

Where a demand pattern exhibits a growth characteristic over a longer time, the forecasting models are required to be more complex than those used in the stationary demand pattern. In growth situations, stationary forecasting models not only produce forecasts which in retrospect lag behind known data, but also produce forecasts ahead which are fixed in value and therefore do not respond to the underlying growth situation. There are many examples of demand patterns exhibiting growth, at least in the medium term. Thus the forecasting models are required to ...

- ... identify the rate of growth of the demand data.
- ... incorporate the rate of growth in the forecasts.

Many demand series are influenced by the seasons of the year and by other events which occur anually (Figure 4.5). In such situations it is possible to establish the degree to which demand in any particular period of the year is higher or lower than for a typical average period. Hence the aim of forecasting models taking seasonality into account is to establish this relationship for each and every period within the year and to use the de-seasonalizing factors that are identified by this process to produce forecasts. For technically reasons it is generally assumed that growth may also exist in demand patterns characterised by seasonality (right side in Figure 4.5. If there is no growth, the analysis simply registers actual growth as negligible.

The simplest demand environment within which to produce forecasts occurs when it can be assumed that the underlying demand process is stationary. The basic assumption within a stationary demand process is that there is variation about a relatively stationary average value and that any change in the average value is due to a special, one-off cause rather than to overall growth or seasonality.

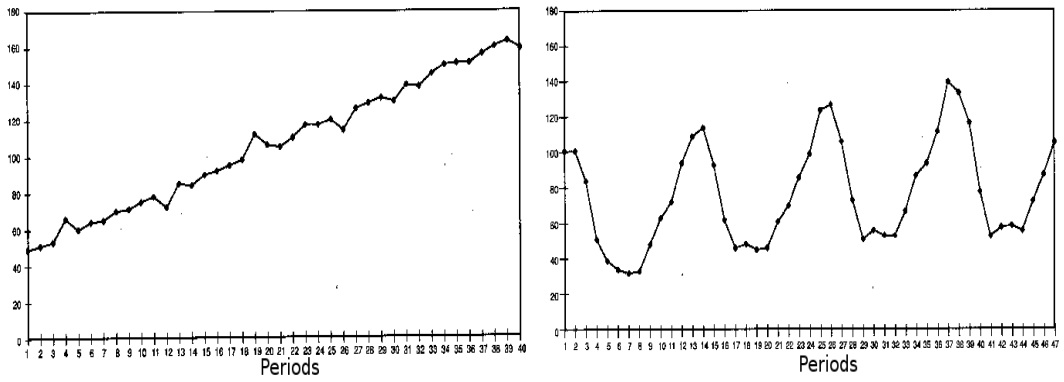

Figure 4.5: Demand patterns with growth (right) and seasonal influence (left)

Before developing specific forecasting models to be linked with inventory control policies, it is clear that in all forecasting situations it is necessary to define the timing of both forecasts and demand data to the particular time period to which they belong or relate. The convention is normally to regard the current period as present time t and refer all other timings to present time. Therefore d_t defines the demand that occurred in the most recent period under consideration. Past time is considered as negative within respect to the current period t, hence d_{t-1} defines the demand that occurred in the period immediately previous to the period in which d_t occurred. Although demand data can only occur in the past, forecasts are clearly targeted to the future. Hence, future time is defined as positive with respect to the current period and f_{t+1} would define the timing of the forecast for the next period follwing the current period. In a stationary demand situation the forecast for one period ahead is the forecast for any number of T periods, where T is any specified forecast horizon projecting into the future. Hence, in the stationary demand situation only the forecast for T periods f_{t+T} is given by:

$$f_{t+T} = f_{t+1} \tag{4.33}$$

The Moving Average

The general form of the **moving average** m_t as a forecasting model is:

$$f_{t+1} = m_t = \frac{1}{n}d_t + \frac{1}{n}d_{t-1} + \cdots + \frac{1}{n}d_{t-n+1} \tag{4.34}$$

where $n = 2, 3, 4 \ldots$ and so on, and where the sum of the n weights will always sum to one, this being the definition of a true average.

However, in practice, the use of a moving average as a forecasting model has the following significant, practical problems:

- It is difficult to start from a situation where no data exist.
- The sensitivity of the number of periods included can not be varied.
- It imposes a sudden cut off in weighting for data not included.
- All data are weighted equally irrespective of their age; but simple logic would suggest that more recent data should be weighted more heavily than older data.

The final problem of equal weighting could be overcome by developing an one-period ahead forecast on an unequally weighted moving average, such as:

$$f_{t+1} = m_t = 0.5d_t + 0.3d_{t-1} + 0.2d_{t-2} \tag{4.35}$$

which is a valid, average based forecasting model since the sum of the weights do indeed add up to one. It is the extension of this concept of an unequally weighted moving average which leads to the development of an average with an infinite number of weights which decrease exponentially with time.

Exponentially Weighted Average

The definition of an average u_t with weights declining exponentially with time would be of the general form of an infinite series defined as:

$$u_t = \alpha d_t + \alpha(1-\alpha)d_{t-1} + \alpha(1-\alpha)^2 d_{t-2} + \alpha(1-\alpha)^3 d_{t-3} \ldots \tag{4.36}$$

where α is a constant whose value must be between zero and one, since to produce a true average the sum of weights must sum to one. A value of $\alpha = 0.2$ is a good compromise. On first examination, a forecast based on Equation 4.36 would appear to be relatively complicated to implement; besides there is an infinite number of demand values. However, it is possible to show that Equation 4.36 can be modified to a much simpler statement such that a one-period ahead forecast f_{t+1} is of the form:

$$\begin{aligned} f_{t+1} = u_t &= \alpha d_t + \alpha(1-\alpha)d_{t-1} + \alpha(1-\alpha)^2 d_{t-2} \ldots & (4.37)\\ &= \alpha d_t + (1-\alpha)\Big[\alpha d_{t-1} + \alpha(1-\alpha)d_{t-2} \ldots & (4.38)\\ &= \alpha d_t + \alpha(1-\alpha)u_{t-1} & (4.39) \end{aligned}$$

which is the equivalent of

$$f_{t+1} = u_t = u_{t-1} + \alpha(d_t - u_{t-1}) \tag{4.40}$$

and since the current forecasting error $e_t = d_t - u_{t-1}$ can be defined as the current demand value d_t minus the one-period ahead forecast evaluated last period u_{t-1}, then

$$f_{t+1} = u_t = u_{t-1} + \alpha e_t \tag{4.41}$$

follows.

In contrast to the moving average, the simple exponentially weighted average offers the following advantages:

- It is easy to initialise, since once an estimate for u_{t-1} is made, forecasting can proceed since all the unknowns on the right hand side of Equation 4.37 are then defined.
- The data storage is economical since u_{t-1} embodies all previous data and hence only the value of u_{t-1} needs to be retained from one period to the next.
- The sensitivity can be changed at any time by altering the value of α just as long as the value of α is set between zero and one.
- It does not produce a sudden cut off in weighting of demand data irrespective of age.

For the simple exponentially weighted average, when the value of α is high, a good response to an upward change can be anticipated. However, with such a high value of α a single high demand value can cause an over-reaction one period late. Conversely, when the value of α is low, the response to an upward change will be poor. For the extreme case of $\alpha = 0$ the forecast is totally insensitive to changes in the demand pattern; and for $\alpha = 1$ the forecast is extremely sensitive to changes and can over-react to relatively small changes. Ideally the best value of α will be that, which minimises the sum of squared forecasting errors, but in the majority of practical situations values of 0.1 or 0.2 are useful compromise numbers.

The simple exponentially weighted average represents an ideal model for producing relatively short-term forecasts for inventory control systems when demand is stationary. When more complex demand patterns exist, such as those influenced by growth or seasonality, adaptions of the simple exponentially weighted average are required.

4.4.2 Monitoring Forecast Systems

Because of the adaptability and flexibility of the family of forecasting models based on the exponentially weighted average principle, these tend to predominate

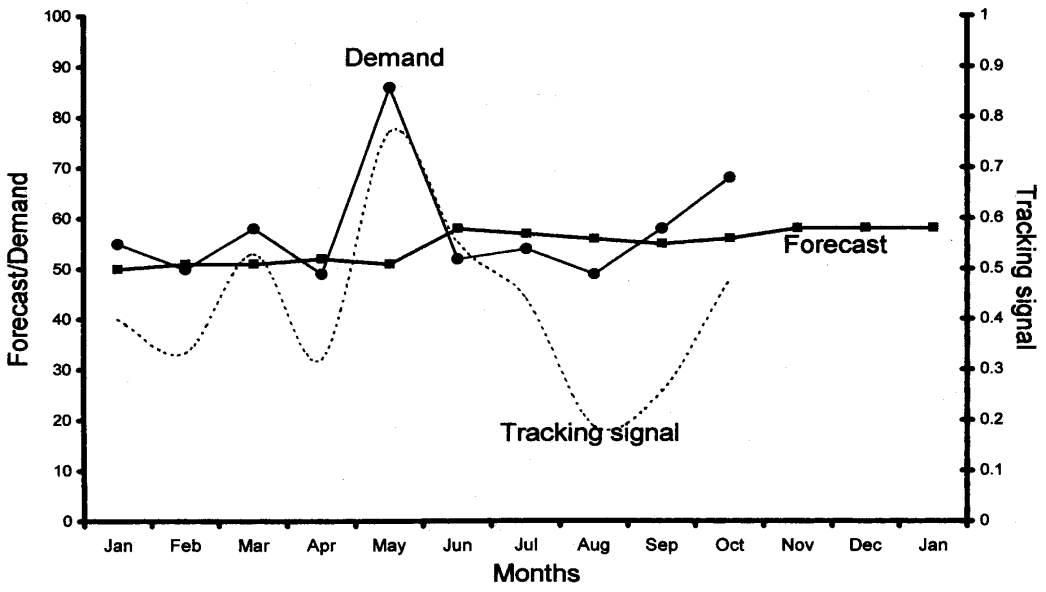

Figure 4.6: Response of a simple exponentially weighted average forecast with $\alpha = 0.2$ [BR01]

in inventory control systems. Hence, when choosing which particular forecasting model to use the choice of model in this case simplifies to the choice of the value of the exponentially smoothing constant α. Although typical values of α are 0.1 or 0.2 for exponentially weighted average forecasting models, it is necessary to have statistical information available, when trying to establish which forecast is best in any particular situation. The two most used statistics for selecting the suitability of forecasting models are now described in detail.

The mean squared error (MSE) is the average of the squared forecasting errors. As such it is often the statistic used to ascertain the best forecasting model, it being assumed that the model with the minimum MSE will be best where:

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \tag{4.42}$$

The mean absolute percentage error (MAPE) is one of the most commonly used monitoring systems in all types of forecasting. It gives an indication of the average size of forecasting error expressed as a percentage of the relevant demand value, irrespective of whether that forecasting error is positive or negative. In computational terms, if the forecasting error e_t is defined as the demand d_t minus

the forecast f_t, it then follows that the MAPE is defined as:

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \frac{|e_t|}{d_t} \tag{4.43}$$

where $|e_t|$ represents the absolute values and n is the number of observations involved. Because the MAPE measures the average relative size of the absolute forecasting error as a percentage of the corresponding demand value, in practice a value of less than 10 per cent would be regarded a very good fit and providing potentially very good forecasts. If the MAPE is < 20 %, the forecast is potentially good; it is reasonable for < 30 and it is inaccurate for > 50 %.

Because the MSE is not a relative measure and cannot be used to compare the forecasting effectiveness between different data series, its main application is to determine the ideal forecasting parameters for a particular data series. The MAPE in contrast is a relative measure and can be used for comparing different data series.

Whithin any forecasting system it is necessary to monitor the accuracy of the forecasts being produced and to manually correct those forecasts which go out of control due to significant changes in the demand pattern. In the following, the monitoring of short-term forecasts is discussed with particular emphasis placed on those situations where many stocked item forecasts are being produced to establish inventory control parameters. Most practical forecasting systems which involve many items operate on the basis that if there is no evidence to the contrary then it is assumed that the forecast is in control; that means there have been no significant changes in the demand pattern to make current forecasts invalid. For such a policy of management by exception to operate successfully, clearly an effective monitoring system is essential. Although several different approaches have been taken with regard to monitoring forecasts, Trigg's (1964) proposal for a tracking signal has become an essential part of the majority of comprehensive short-term forecasting systems.

The **Trigg** or **smoothed error** tracking signal is based on the fact that if forecasting errors e_t are defined as demand minus forecast then the current smoothed error $\bar{e}_t$ is defined as the exponentially weighted average of the forecasting errors e_t and is produced by:

$$\bar{e}_t = \alpha' e_t + (1 - \alpha')\bar{e}_{t-1} \tag{4.44}$$

where $\bar{e}_{t-1}$ is the value of the smoothed error for the previous time period. The current value of mean absolute deviation (MAD) is then defined as the exponentially weighed average of the absolute forecasting errors $\bar{e}_t$ using the formula:

$$\text{MAD}_t = \alpha' \bar{e}_t + (1 - \alpha')\text{MAD}_{t-1} \tag{4.45}$$

where the absolute value signs | | indicate that all errors e_t are treated as positive irrespective of their actual polarity, and where MAD_{t-1} is the value of the mean absolute deviation for the previous time period. In both Equations 4.44 and 4.45 the parameter α' is an exponential weighting constant whose value must be between zero and one. By convention, for monitoring applications α' is set universally at a fixed value of 0.2. Having defined the smoothed error $\bar{e_t}$ and the mean absolute deviation MAD_t, the tracking signal T_t is then defined as the ratio of the smoothed error to the mean absolute deviation, hence:

$$T_t = \frac{\bar{e_t}}{\mathrm{MAD}_t} \tag{4.46}$$

Given that the value of α' used to produce both $\bar{e_t}$ and MAD_t are the same and set at 0.2, then in practice, irrespective of the data involved, the value of the tracking signal can only vary between $+1$ and -1. In the extreme case where a significant increase in demand has occurred, all forecasting errors are positive and effectively Equations 4.44 and 4.45 become the same and $\bar{e_t} \rightarrow \mathrm{MAD}_t$ and hence $T_t \rightarrow +1$. Contrariwise, in the extreme case where a significant decrease in demand has occurred, all forecasting errors are negative and it follows that $\bar{e_t} \rightarrow -\mathrm{MAD}_t$ and hence $T_t \rightarrow -1$. If the value of the tracking signal exceeds 0.7, the user can be 95 % confident in the hypothesis that the accompanying forecast is out of control due to an untypically high set of demand values for which there should be an identifiable, external cause. If the signal is lower than -0.7 the forecast is also out of control, but this time because of an untypically low demand. In an inventory situation a comprehensive method of implementing the smoothed error tracking signal would be: at first calculate the value of the tracking signal for all items. Then those items should be listed, for which the absolute value of the tracking signal exceeds a value of 0.7; after that for this listed items the reasons of the forecast errors have to be investigated. In addition it is necessary to check that the forecast is producing reasonable results and is typically achieving a mean absolute error of less than 20 %; besides the tracking signal shall have a value of less than 0.7.

4.4.3 (Auto-)Correlation

Correlation

In probability theory and statistics, covariance is the measure of how much *two* random variables vary together. The simple variance measures how much a *single* variable varies. If two variables tend to vary together in the same direction, then the covariance between the two variables will be positive. On the other hand, if one variable goes down during the other is rising, the covariance between two variables will be negative.

The covariance between two real-valued random variables X and Y, with expected values $E(X) = \mu$ and $E(Y) = \nu$ is defined as

$$\begin{aligned} Cov(X,Y) &= E((X-\mu)\cdot(Y-\nu)) \\ &= E(X\cdot Y) - \mu\nu \end{aligned} \tag{4.47}$$

The second equation is valid because of the Steiner theorem. If X, Y are random variables and a, b are constant, the following facts are a consequence of the covariance definition:

$$\begin{aligned} Cov(X,Y) &= Cov(Y,X) \\ Cov(X,X) &= Var(X) \\ Cov(aX,bY) &= ab\cdot Cov(X,Y) \end{aligned} \tag{4.48}$$

If X and Y are independent, then their covariance is zero. This follows under the assumption of independence:

$$\begin{aligned} E(X\cdot Y) &= E(X)\cdot E(Y) \\ \Rightarrow Cov(X,Y) &= \mu\nu - \mu\nu = 0 \end{aligned}$$

The converse is not true: if X and Y have covariance zero, they don't have to be independent. The measurement units of covariance Cov(X, Y) are those of XY. The correlation (which depends on the covariance) is a dimensionless measure of linear dependence:

$$\sigma = \frac{Cov(X,Y)}{\sqrt{Var(X)}\cdot\sqrt{Var(Y)}} \tag{4.49}$$

Random variables whose covariance is zero are called uncorrelated.

Autocorrelation

Empirical time series normally have a partly repeating pattern. Such a pattern can be described by a measurement of the correlation between the values of a time series, which is called *autocorrelation* or *autocovariance*. This is a measure for the relation between data, which have a fixed time lag to each other. For example the autocorrelation with time lag $\Delta t = 1$ is a measure for the relation between y_t and y_{t+1} for $t = 1, \ldots, n-1$; with $\Delta t = 2$ the autocorrelation is a measure between y_t and y_{t+2} for $t = 1, \ldots, n-2$; and so on. From n values of a time series, $n-1$ successive pairs $(x_1, x_2), (x_2, x_3) \ldots (x_n - 1, x_n)$ can be formed. Their autocovariance is:

$$\gamma = \frac{1}{n-1}\sum_{t=1}^{n-1}(x_t - \mu)(x_{t+1} - \nu) \tag{4.50}$$

Thereby μ is the arithmetic average from the values $x_1, \ldots, x_{n-1}$ and ν is the average from $x_2, \ldots, x_n$. This means that the autocovariance measures the linear relation of $n-1$ values of two time series. According to this, the autocovariance of values, which are further apart than one time unit, can be determined. Therefore the autocovariance depends on the time lag τ:

$$\gamma(\tau) = \frac{1}{n-\tau} \sum_{t=1}^{n-\tau} (x_t - \mu)(x_{t+\tau} - \nu) \qquad \tau = 0, 1, 2, \ldots, n-1 \tag{4.51}$$

With $\gamma(\tau = 0) = Var(X)$. The autocorrelation is the standardised form of the autocovariance:

$$\rho = \frac{\gamma(\tau)}{Var_\mu Var_\nu} \tag{4.52}$$

Thereby Var_μ, Var_ν are the variances of the time series belonging to the expectation values of μ, ν. The autocorrelation can be seen as a hint, whether there is a regular component in time series. If there is one, the autocorrelation is near +1 when the periodicity is met and near -1 at half of periodicity. If all values are near zero, presumably no regular components occur. The calculation of the autocorrelation makes only sense, if enough data are available.

Chapter 5

Physical Optimisation and Forecasting

In this chapter a new forecasting model is going to be developed. Besides, this model will be compared to other methods using different data types. Thereby a special period of time is chosen (for example a year) depending on the data at hand. In order to verify the quality of the forecast, several dates (for which a forecast is made) are taken within the chosen period. Thus the forecast can be compared directly to real values of the data series and the variation can be measured.

5.1 Short Term Forecast

5.1.1 Model with Simple Deviation

A simple way to make a forecast is to look at historic data and to extrapolate it into the future. Actually most methods work like this with a more or less complicated theory behind it. If there is no historic information, of course there can be no extrapolation and other ways of forecasting have to be found. But here at least three historic values shall be available. If somebody wants to make an economic forecast for the sales figures of a business without the help of statistical methods, he really would have much to do, because in many cases there are hundreds or thousands of items with historical data over several periods. In practice the most used forecasting models in inventory control are the **moving average** and the **exponentially weighted average** (see section 4.4.1) and therefore they shall be used for reasons of comparison.

The new model works like this: it starts with a fixed budget to be invested for production or orders of the coming period; alternatively the budget restriction

can also be transformed into a capacity restriction, depending on the intention of the management. If there are historic figures for a range of items, the planned investment has to be distributed among the different items such that it is near the real sale as far as possible. And if the sales figures are transformed into the equivalent of the required stockroom space, the available stock capacity has to be distributed according to the same principle as for the budget. The above mentioned methods of the 'moving average' and the 'exponentially weighted average' cannot integrate this restriction in a proper way: they just calculate the average of the historic data and the budget is not considered. But how is the new model able to cope with such a restriction?

At first the budget (or the stock capacity) is distributed randomly among the considered items; this could be called the first heuristic forecast, which is of course very poor. In order to determine how bad a forecast *could* be, it is compared to the figures of the last periods. A measure for this is given by following function:

$$\mathcal{H} = \sum_{i=1}^{N} \sum_{t=1}^{T} |f_i - d_{it}| \tag{5.1}$$

Here N is the number of items, T the number of historic periods, f_i the current forecast and d_{it} the sales figures (or the needed stockroom) of item i in period t. $\mathcal{H}$ is the so called Hamiltonian of physical optimisation (see chapter 2), which is the equivalent of energy in physics and of costs in economics. $\mathcal{H}$ is going to be optimised by one of the physical algorithms (for example simulated annealing), which try to find the solution with the minimum deviation of the forecast to the historic data. Naturally there can be no perfect forecast, because nobody knows the future; but the past is at least a good point of reference and thus it makes sense to determine a forecast, which is the best fit to the historic data. In order to start the physical algorithm, the so called **move** has to be substantiated. Here two different items are randomly chosen and from one of them a random budget fraction is taken and shifted to the other one. As part of the algorithm this move is repeated as often as necessary to find a good forecast. Another way to measure the quality of a forecast with respect to the past can be the following energy or cost function:

$$\mathcal{H} = \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha(1-\alpha)^t |f_i - d_{it}| \tag{5.2}$$

with α as weighting factor, which considers the newest data more than the older ones, depending on the value of α. Both energy functions 5.1 and 5.2 shall be used and applied to different data series.

5.1.2 Model with Value at Risk

This model was developed from the author [Zi05] in his diploma thesis and applied to different types of problems. It has been shown that this model can be used to forecast the sales figures of a supermarket, the performance of a team or the championship of the German Soccer League. The results were good from the standpoint of physical optimisation: in all cases a physically related model could be developed and the optimisation has shown the typical characteristics. That was the main part of the diploma thesis and no comparison to other forecasting methods was made; therefore this shall be done in the following chapter 5.1.3.

The model with Value at Risk (VaR) works similiar to those with simple deviation: Here we have also a forecasting vector, which contains the forecast for different items. Then this vector is compared to the historic data and changed corresponding to the budget and the quality of the solution. So far the proceeding is the same. The difference is that the VaR-Model doesn't sum up the whole deviation of the comparison to all historic periods, but only for each period separate. This produces T deviations (the number of periods), which can be illustrated and summarised in a frequency distribution.

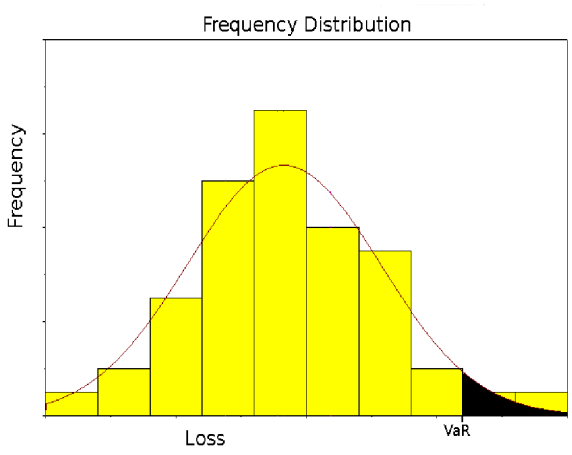

Figure 5.1: Frequency distribution of the deviation between the forecast and the historic periods

A possible characterisation of the distribution is the average and the so called 'Value at Risk'. The VaR_x is the loss-value, which is not exceeded with a special probability $x\%$. For example with $x = 95\%$ the deviation (of the forecast vector to the single periods) is lower than the VaR according to the underlying frequency distribution. These two variables define the Hamiltonian according to which the forecast vector is optimised by a minimisation of the deviation:

$$\mathcal{H} = VaR_x + \gamma \bar{R}_P \tag{5.3}$$

where $\bar{R}_P$ is the average of the frequency distribution and γ the weight of the average in contrast to the VaR. The usefulness of this model is theoretically good, but practically its use is problematic. The first problem is that normally there are just a few periods of historic data and thus the VaR is hard to calculate and not a good measure for the risk of the frequency distribution. In the diploma thesis of Zizler this problem was solved by the generation of large data series for each item based on historic values. After that the existing correlations between items had to be re-adjusted to the original ones by another optimisation algorithm with the Hamiltonian:

$$\mathcal{H} = - \sum_{i,j=1,i\neq j}^{N} |\sigma_{ij}^{new} - \sigma_{ij}^{old}| \tag{5.4}$$

σ_{ij}^{new} are the new correlations and σ_{ij}^{old} the old ones. The results of the correlation optimisation are quite good and also the forecast is respectable. But this method needs a lot of effort in programming and computation time and therefore a practical application is doubtful; especially because the results are not better than the 'moving average'. The advantage of this method is that a fixed budget or stockroom can be integrated into the optimisation; above that the optimised frequency distribution could be manually changed because of external incidents like intensive advertising. Naturally this doesn't result in a perfect forecast, but it can be used as tool to analyse the impact of special occurences. In order to reduce computation time, a variation of the VaR-Model shall be presented: thereby the generation of new data is skipped and only historic data are used. Because of that the Value at Risk has to be calculated in a different way and is now just the highest value of the above stated deviations of all periods. For ten historical periods this would be the VaR_{90}. If there are not more then 100 periods of historic data, it should be a good and efficient approximation. Thus the Hamiltonian is the same as in Equation 5.2; just the VaR is calculated differently.

5.1.3 Application to Grades of Soccer Players

At first grades of soccer players as a very special kind of data shall be used. Those grades are given by the "kicker", a german sports magazine. The magazine evaluates each player of the German Soccer League for every game of a season with a grade between 1 and 6; also half-grades are possible. At most there are 34 different grades in one season. On the first view the grades don't have to do anything with sales figures, but they can easily be interpreted as such. Actually they are better than pure random numbers, because there are correlations between soccer players of one team, and that is just like in a real company with different items to sell or store. Of course it is a disadvantage not to work with real figures, but for a first test of the described model above the interpreted grades are quite useful.

Before the simulation starts and the random forecast is optimised, the different parameters of the simulation have to be determined. The first non-physical variable to determine is the number of items for which a forecast is going to be made. Here the figures of 439 items are forecasted, because this was the number of soccer players in the German Soccer League of the season 2003/04. The next variable to fix is the budget for production or ordering. Analogue, the available stockroom has to be fixed when the grades are not interpreted as sales figures, but as space units of each item. One might object that it is a great difference to have square meters instead of sales figures; but normally each space unit of an item can be related to a monetary unit and thus there should be no problem. Because of the equivalence between budget and stockroom, in the follwing only the term 'budget' is used. Anyway the budget has to be proportional to the number of items. If the budget is not fixed by the management, it can be optimised by several simulation runs. Here a value between 660 and 700 seems to be a good one for the budget of the 439 items. Beyond this range the results of the optimisation are rapidly getting worse. For other analysed aspects of the model thus the budget was set to 680.

The physical variables in this model have a more or less important meaning. From a physical point of view they are the essential part; but practically they can be used as it is comfortable. At first the start and end temperatures have to be chosen according to subsection 2.3.4. It was calculated that the start temperature for the problem with 439 items and the budget of 680 is in the order of one thousand. Thus the start temperature was set to 1000. The determination of the end temperature is a problem, because the system is never totally frozen. The reason for this is that there are always small improvements of the energy, even for low temperatures. Therefore the optimisation run has to be cut off, when the improvements stop to be significant. This is normally easy to see and in this case the run was stopped at 0.001; the cooling scheme itself was logarithmic. In Figure 5.2 on the top the energy and the heat capacity is shown for a simulation run, where a forecast is made based on the historic data of three periods:

Both physical variables show the typical course: the energy is falling down from a high level (equivalent to a disordered state of the system) to a low level (ordered state). In between the state of the system is rapidly changing, what can be seen in the heat capacity as the variance of the energy, too. The heat capacity does not go down to zero because the system is never completely frozen and therefore the resulting small changes in the energy produce a more or less high value for the heat capacity at low temperature values. At each temperature step 10000 lattice sweeps were rejected in order to have an equilibrium; after that 10000 values were measured. The number of sweeps was multplied with 10 for $0.1 < T < 100$ because of the strong changes in this temperature range. Another simulation forecast with the same parameters, but on the basis of 13 historic

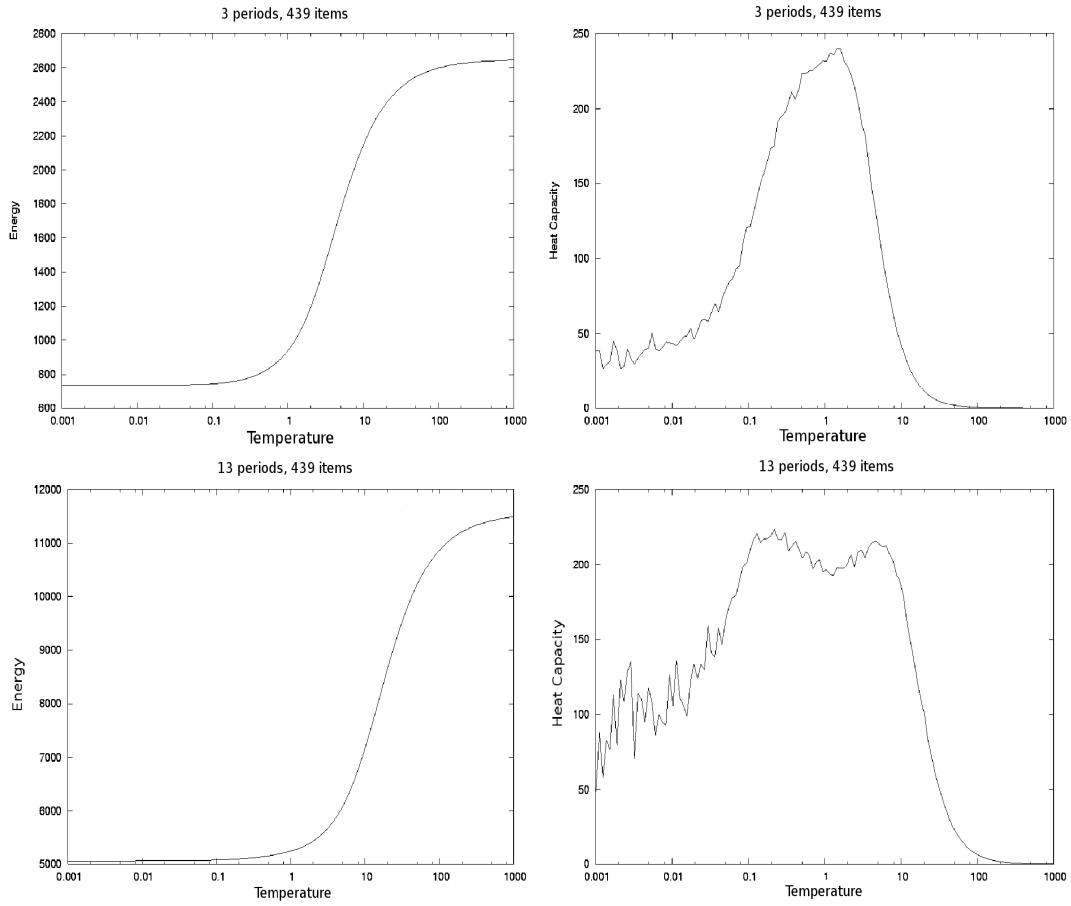

Figure 5.2: Energy and heat capacity of a simulated forecast based on three (above) and 13 (below) historic periods with 439 items and a budget of 680

periods, shows Figure 5.2 at the bottom.

The course of the energy function is quite similiar to the simulation with three periods; only the enery level is higher, because more periods are considered. In contrast to this, the heat capacity is more different. The reason is that more historic periods are considered and it is harder for the algorithm to find **one** optimal solution, when there are many similiar solutions with slightly different energies. Thus the heat capacity has a stronger fluctuation at lower temperatures and a second peak near the first maximum. This phenomenon doesn't change for a smaller number of items. For a simulation with 50 items, a budget of 80 and the same parameters as in the last example, the results are pretty similiar.

The decisive point of this analysis is the practical value of this forecast with a physical algorithm and model. Therefore the results are compared to standard methods in practice. Equation 5.1 is compared to the 'moving average' and

Equation 5.2 to the 'exponentially weighted average'. The best way to show the differences is the graphical illustration in Figure 5.3. In this Figure all variances

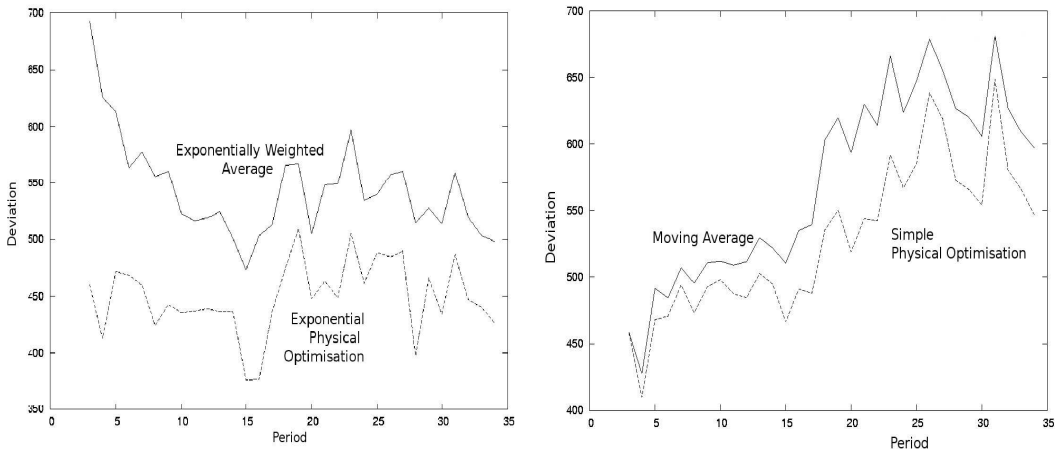

Figure 5.3: Comparison of the different forecast methods

of the forecasts from period three to period 34 are shown. Each forecast of the individual methods is compared with the real value in the following period and the deviation of every item is summed up. Here the physical model with the α-factor shows the best performance; the deviation is on the same level for all forecasts. In contrast to the 'exponentially weighted average' the results of this method are approximately 10 % better; the difference is obvious. And if there are just a few periods available, the 'exponentially weighted average' seems to be even worse. For this kind of data thus the 'moving average' is better qualified, if there are just a few periods of historic data. But the 'moving average' is still less good than both physical models. On the right side in Figure 5.3 it can be seen that the 'moving average' and the physical model without α produce worse forecasts for an increasing number of historic data.

The interpretation for this is clear: Originally the data are soccer grades and they are measures for the performance of a soccer player. During the season the formation of a team changes more and more; some players are getting better and others are getting worse. Therefore the most recent grades are more important and have to be weighted stronger than the older ones.

Recapitulating, it has to be said that the method of forecasting with Equation 5.2 shows better results than the 'moving average' and the 'exponentially weighted average', at least for this kind of data. Besides, it is not just a theoretical model, but can also be applied in practice. The computation effort is bigger than for the compared methods, but small enough to be used in daily business. Of course this results can not automatically be used for each kind of data. At first an analysis

about the type of data is necessary; after that a decision about the forecasting model is possible.

The VaR-models in subsection 5.1.2 are less useful in practice. The computation effort is relatively high and the results are not as good as they should be in order to justify a further examination. In contrast to the first model, the second VaR-model shows a better performance in computation time. The performance of the forecast with this model is similiar to those with the 'exponentially weighted average' (Figure 5.4)

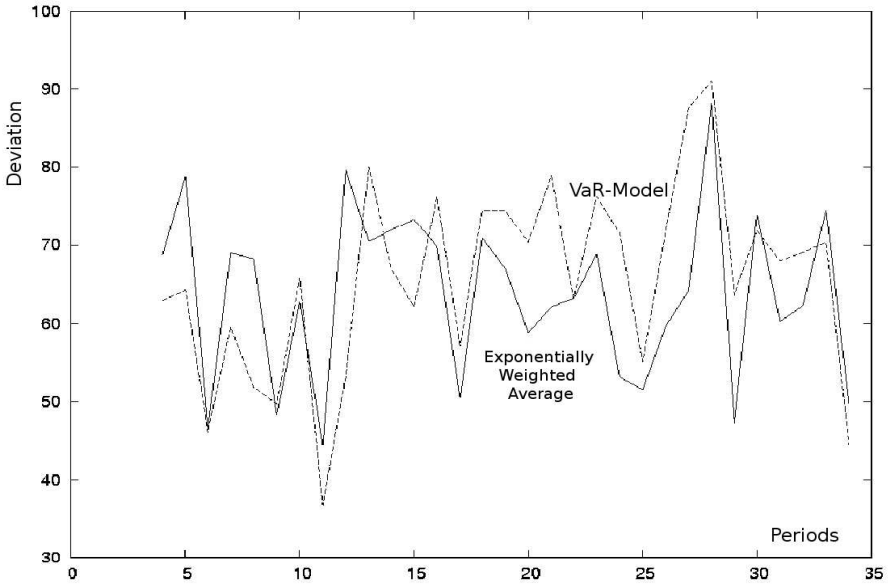

Figure 5.4: Comparison of the different forecast methods

Obviously the performance is not better. So why should somebody use the more complicated model ? The first reason is that it can integrate a budget or stockroom restriction. The next one is that no statistical distribution has to be assumed and just the real historic data are used; besides a value can be given which makes a statement about the probability of the highest deviation of the forecast in the past. If there are ten historic periods, the VaR is the value which is exceeded with a probability of 10%.

5.2 Medium Term Forecast

One way to make a medium term forecast is to produce random forecast values with a probability distribution. Therefore a Gaussian distribution can be assumed with an expectation value and a variance derived from historic values.

In this work another option shall be tested: the frequency distribution of the historic values shall be smoothed in order to get a quasi-continuous distribution. The first step is to choose an interval for the historic frequency distribution. The decision depends on the available historic data: if the range of values is very high and / or the number of values is low, then the intervals have to be broad; otherwise the intervals should be smaller. The next step is to smoothen the frequency distribution. For this the stacks of the frequency distribution are divided into two parts; then each half is adjusted to its neighbour. This proceeding is repeated several times until the distribution is quasi-continuous. In Figure 5.5 the pro-

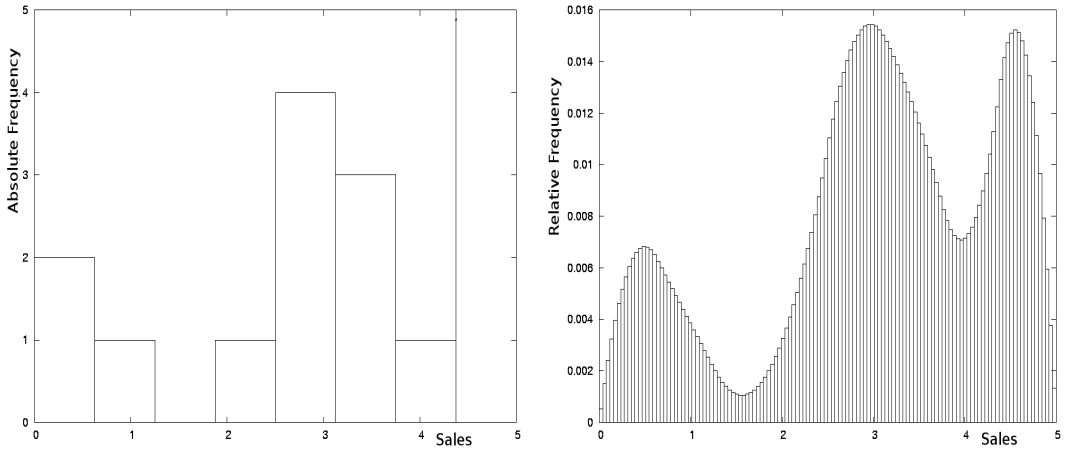

Figure 5.5: Frequency distribution before (left) and after (right) smoothing

ceeding is repeated four times. The resulting frequency distribution can be seen on the right side; the original one is on the left side. There are two benefits of the smoothing. The first advantage is that there are more detailed values; with the unsmoothed frequency distribution just a few discrete values can be produced for a forecast. Secondly the quasi-continuous distribution can be changed due to reasons of external information about the future sales figures. For example there can be a sales promotion in order to increase the sales. Then the distribution can be transformed depending on the expected change of the sales promotion.

A historic test of the medium forecast with the smoothed frequency distribution has shown plausible values. Of course this forecast cannot be better than a simple update of the historic values, because the distribution is just built on the historic data. But it makes sense to have such a distribution for the above mentioned reasons and thus it will be used for the optimisation of the inventory policies in the following section.

Another historic test about the significance of correlations couldn't prove the assumption that optimised correlations between the forecasted sales figures of

different items improve the total forecast. In principle it is clear that correlations between different items exist: people who buy ice cream will need a cone, too. Thus it should be necessary to integrate correlations in a forecast. In this case the used soccer grades should also have strong correlations between players of one team. But the advantage of the physically optimised correlations was mostly very small and not systematic.

Chapter 6

Optimisation of an Inventory System

There are many mathematical techniques like the simplex algorithm, but their preconditions are very idealised (e.g. continuity or differentiability of the objective function) and therefore mostly inapplicable for practical situations. Because of their adaptability metaheuristics like physical or genetic algorithms are able to cope with complex structures in practice. This will be demonstrated in this chapter by optimising a widely realistic inventory system; thereby different order policies are optimised with simulated annealing as the most popular physical optimisation algorithm. In chapter 7 the physical optimisation of the modelled inventory system is compared to the results of a genetic algorithm using the same model; in addition to that a few research results of other autors in the same area are presented.

6.1 Implementation of an Inventory Problem

6.1.1 Variables of the Inventory System

The system described in the following is a multi-item-inventory. The demand is fulfilled immediately in the simulations and for a few others a probability distribution is assumed. If there is a stock-out, the items are provided by a competitor. The sales price is a variable percentage higher than the cost price; the difference is the inventory return.

There are several different inventory policies. Most widely used is the (s, Q)-policy, which orders the quantity Q, if the stock is below the order point s. Often used is also the (t, S)-policy: after a fixed period t the stock is replenished up to quantity S. Because of their widespread application, these two models are used

in the following and optimised with physical algorithms. Besides, the (s,S)-policy was constructed as a combination of those two policies.

The demand depends on the type of item and cannot be determined exactly. Normally there has to be a forecast which is oriented at historic data. Here no forecast is made; the possible policies are tested only with historic data. Thus it could be said that the historic data *is* the forecast.

The monitoring period is the time between two inspections. This means: at the beginning of each period the inventory is reviewed. If there is no monitoring period, every access and outflow is included instantly. For the simulation there shall be a monitoring period; the outflow is assumed continuous. The order period is not identical with the monitoring period, but a multiple of it. In the case of a constant order period ((t, S)-policy), the order quantity should be variable, because the demand is normally stochastic. But if the order period is variable ((s, Q)-policy), the order quantity can be fixed and an order can be stated, if the stock is below a fixed safety limit s. This variable determines how much quantity units are available for security reasons in order to have an adequate customer service; besides, the safety stock has a great impact on the opportunity costs. If the lead time is zero and the monitoring period is very small, the safety stock is also zero, because each deficiency is immediately realised and the ordered items are promptly available. Beside the safety stock s and order period t, the order quantity is the decisive variable.

The time between the release of an order and the delivery of an item is called lead time. Sometimes this variable is considered zero, sometimes it is constant and in other cases it is stochastic. In the following simulation the lead time is mostly zero, but it can easily be integrated as constant and in some cases a stochastic lead time is implemented.

Basically there are two different *order* costs in an inventory system: costs which depend on the ordered quantity and fixed costs per order. Furthermore the inventory costs consist of costs for the inventory space, the employees, insurances, spoilage and capital commitment. The costs for a stock-out are difficult to calculate. One possibility is to take the costs for the purchase at another producer; this idea shall be used in the following.

6.1.2 Hamiltonian

In this section a model of an inventory system shall be operated with two re-order level policies and a re-order cycle policy; the policies itself are determined by a physical optimisation algorithm. The (s, Q)-policy is characterised by the safety stock s and the order quantity Q. Step by step those variables are changed and improved by the algorithm. For example, Q or s can be increased by a few units and then the new (s, Q)-policy is tested for its quality; the same proceeding is

applied to the re-order cycle policy. The test is done by an application of the new policy to historic data. If the return is higher, the step (move) is accepted; if it is lower, it is accepted with a special probability due to the criterion of the algorithm. During the simulation run, the probability to accept worsening moves is getting lower; thus the tested policies are going to be better in a systematic way. At the end we have a very good solution for the problem of determining the best policies. The quintessence of the optimisation algorithm is the Hamiltonian or the return/cost function:

$$\begin{aligned} \mathcal{H} &= \mathcal{H}_{Return} + \mathcal{H}_{Storage} + \\ &\quad \mathcal{H}_{Capital} + \mathcal{H}_{Order} + \mathcal{H}_{Penalty} \end{aligned} \tag{6.1}$$

The Hamiltonian consists of 5 sub-terms. The first term represents the return of a sold unit. The next four terms stand for the different costs of an inventory which are considered in the simulation. Thereby the last term has an exceptional position, because the penalty costs are mostly virtual and have to be determined according to a subjective estimation expressed by a factor λ; those costs can be real, if the exaggerated demand can be provided by a competitor. Storage and capital commitment costs are calculated per monetary units; order costs have a fixed value per order plus a variable element proportional to the order quantity. Or in greater detail:

$$\mathcal{H}_{Return} = -\sum_{i=1}^{M}\sum_{t=1}^{T} R^i \cdot \left[\Delta_t^i \cdot \Theta(x_t^i) + y_t^i \cdot \Theta(-x_t^i)\right] \tag{6.2}$$

$$\mathcal{H}_{Storage} = +\sum_{i=1}^{M}\sum_{t=1}^{T} c_L^i \cdot \left[(\frac{\Delta_t^i}{2} + x_t^i) \cdot \Theta(x_t^i) + \frac{1}{2}\frac{y_t^i}{y_t^i - x_t^i} \cdot \Theta(-x_t^i)\right] \tag{6.3}$$

$$\mathcal{H}_{Capital} = +\sum_{i=1}^{M}\sum_{t=1}^{T} c_C^i \cdot \left[(\frac{\Delta_t^i}{2} + x_t^i) \cdot \Theta(x_t^i) + \frac{1}{2}\frac{y_t^i}{y_t^i - x_t^i} \cdot \Theta(-x_t^i)\right] \tag{6.4}$$

$$\mathcal{H}_{Order} = +\sum_{i=1}^{M}\sum_{t=1}^{T} [c_{fix}^i + c_{var}^i \cdot q_t^i] \tag{6.5}$$

$$\mathcal{H}_{Penalty} = +\sum_{i=1}^{M}\sum_{t=1}^{T} \lambda \cdot |x_t^i| \cdot \Theta(-x_t^i) \tag{6.6}$$

where

M	Number of items
T	Number of periods
Δ_t^i	Demand of item i in period t;
	$\Delta_t^i = y_t^i - x_t^i$ for $x_t^i \geq 0$ and $\Delta_t^i = y_t^i$ else
$\Theta(x_t^i)$	Heaviside function;
	$\Theta(x_t^i) = 1$ for $x_t^i \geq 0$ and $\Theta(x_t^i) = 0$ else
x_t^i	Stock of item i at the end of period t; $x_t^i \in \mathbb{R}$
	$x_t^i < 0$: magnitude of a stockout
y_t^i	Stock of item i at the beginning of period t
q_t^i	Order quantity of item i at the end of period t
R^i	Return factor of item i
c_L^i	Storage costs of item i
c_C	Capital commitment costs
c_{fix}^i	Fixed order costs of item i
c_{var}^i	Variable order costs of item i
λ	Penalty factor

Table 6.1: Variables of the Inventory-Hamiltonian

In 6.1.3, 6.1.4 and 6.1.5 different model configurations are presented.

6.1.3 Standard Parameter Configuration

This multi-inventory system is a dynamic and continuous-discrete model. The lead time is zero and there is no budget restriction. Naturally there is a capital stock which has to be invested and the possibility to take out a credit. Both alternatives have different interest rates, but for a first optimisation test, the described model shall be as easy as possible and thus the interest rate is collectively set to 2% per week.

A time unit is one week. In case of the sales of a steel company the whole planning period consists of three months or 17 weeks; for the soccer grades the planning horizon is one season. Whitin the planning period the combination of order quantity and safety stock ((s, Q)-policy), order point ((t, S)-policy) or the order limit ((s, S)-policy) is constant. The policy is executed on historic data (the same as in section 5.1.3) and not on a probability distribution. The monitoring period is one week and the outward stock movement shall be continuous during this time. A new order is released, if the criterion of the particular policy is fulfilled. The fixed costs of an order are 50 monetary units; variable order costs are not considered. Furthermore the storage costs are set to 1.5 % of the cost price and the return of the sold items shall be 40 %. The stock is calculated in

monetary units: items which are produced or bought for 100 monetary units are therefore sold for 140. Analogue, a stockout shall cause costs of 40%, when the items have to be provided by another competitor.

The purpose of stockkeeping is to maximise the return at the end of the period. Thus the different costs have to be optimised in such a way that the return is equal or higher. It will be shown that physical algorithms can make an important contribution to this optimisation.

6.1.4 Standard Configuration + Stochastic Lead Time

The parameter constellation shall be the same as in the standard configuration. Only the lead time is stochastic and not zero. The case of a constant lead time unequal to zero is not tested, because it is principally the same as zero lead time; the optimisation algorithm will produce other results, but the optimisation procedure is the same. The stochastic element is introduced by a simple probability distribution for the lead time: With a probability of 50 % the ordered items shall arrive immediately; in case of the other 50 % the items arrive one period later. That is a more or less realistic assumption. In reality there can be a lot of different lead times for the single items of a company. But primarily this is a scientific analysis and does not need to reproduce all facets of real problems in the first place. Nonetheless this distribution for the lead time can be modified and adjusted to real values.

6.1.5 Standard Configuration + Capacity Restriction

Another essential parameter in an inventory model is the capacity. Normally there is a restricted space for the items to sell or distribute and thus the capacity has to be considered. Although the established supply chain management can help to reduce unnecessary stock, still there has to be some storage at one place or another. And of course it can be improved by a more efficient and ("physically") optimised inventory policy. It is always good to reduce the capacity of a stock, because each square meter being controlled costs money. But one has to know what consequences follow from a reduction of the stockroom. An important question would be, whether there is a high decrease of customer service if the inventory level is lowered. A possible evolution can be shown by simulations like those in the following sections.

6.2 Inventory Optimisation - Part I

6.2.1 (s,Q) - Level Inventory Policy

This is the most used policy in inventory control [BR01]. In the following the results of an optimisation with different algorithms shall be presented. The model has been developed and the parameters and configurations have been defined in the previous sections. At first simulated annealing (SA) as a real physical algorithm is tested; SA takes the main part of the following. Then threshold accepting (TA) as a related, but less physical algorithm is used; mostly TA is applied for reasons of a shorter computation time. The physical deficiency of TA is acceptable, because the results have nearly the same quality with a lower computing time.

Simulated Annealing The type of the first used data are the soccer grades just like in subsection 5.1.3. The reasons for the selection of this kind of data are the same as above and don't need to be stated again. In this context of optimising an inventory policy it is interesting to have a closer look at the development of sales figures over a special period of time in Figure 6.1.

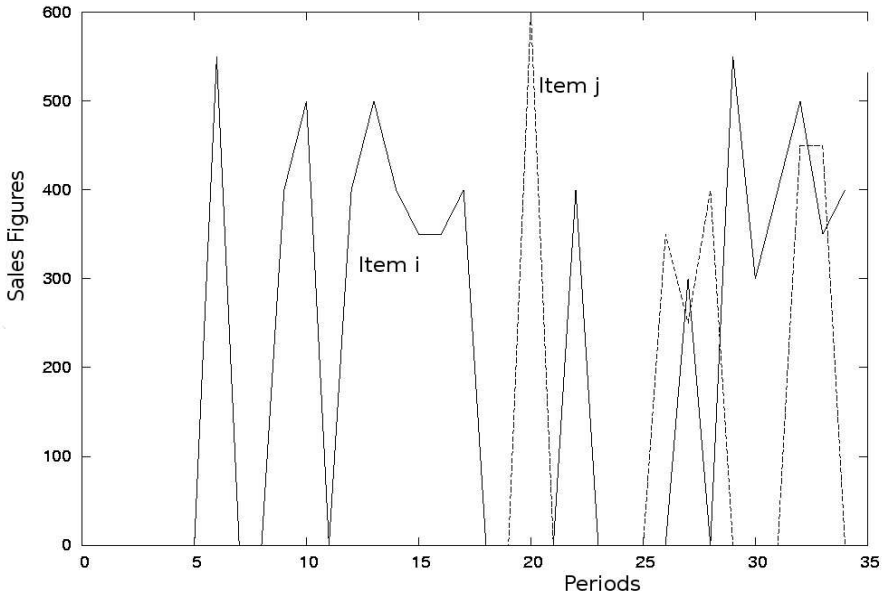

Figure 6.1: Sales Figures of two different items i and j over a time of 34 periods

The values of two different types of items are shown over 34 periods. Item i has a relatively continuous demand, whereas item j is just sold in the first 14 periods. Those two examples show the variety of the possible data series. And

because of the great number of items in one company, it is impossible to look at each item seperately. Thus in practice it would be very helpful to have a simple and well working tool to determine the right order policy for a greater number of items. In the following the developed model of section 6.1 is tested for different configurations and the results are going to be presented.

The physical variables of the simulation are determined as follows. The temperature range goes mostly from $T_{Start} = 100$ to $T_{End} = 0.1$; the cooling scheme is again logarithmic. Sometimes it can be necessary to vary one of those variables and they have been set one order higher or lower, depending on the number of included items. At each temperature step 1000 lattice sweeps have been rejected in order to have an equilibrium; after that 10000 values have been measured; the number of sweeps was set to 100. The error of the simulation results due to this parameter values lies at about 1%. Energy and the heat capacity of a simulation for 1 and 50 items and 34 periods look like presented in Figure 6.2.

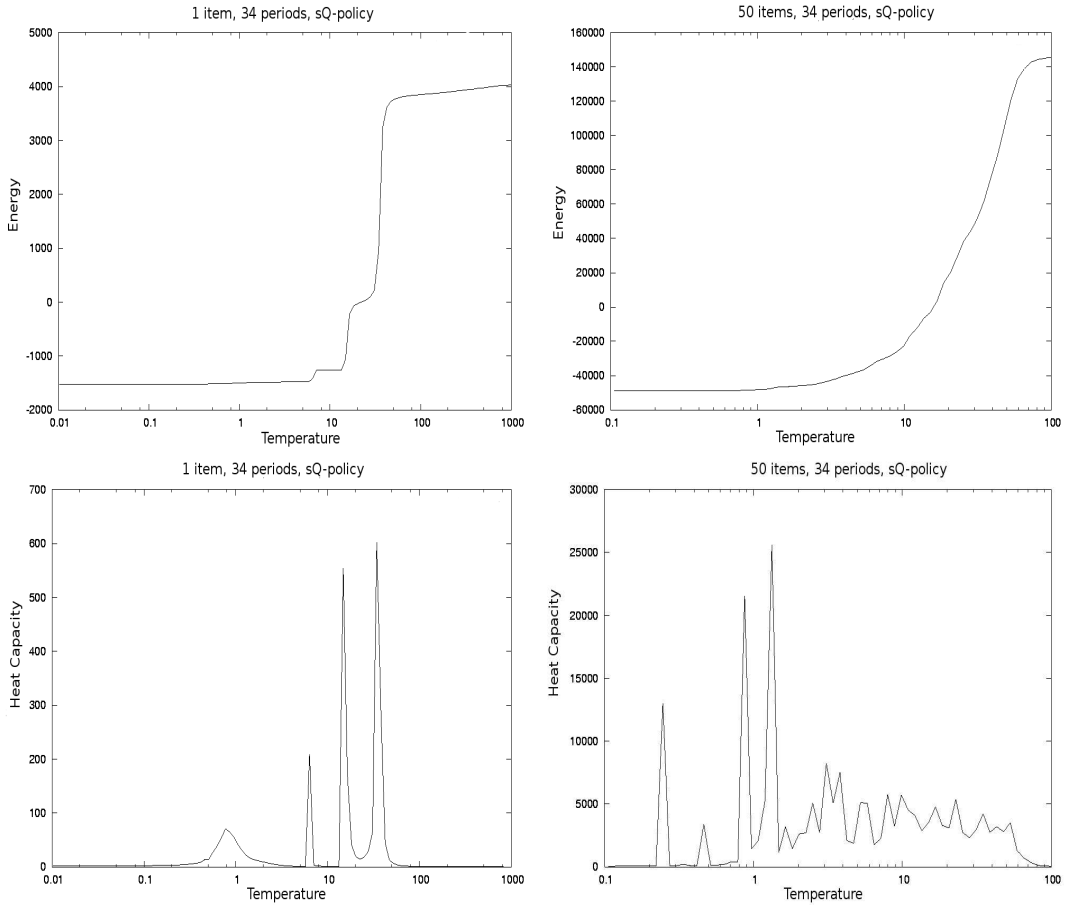

Figure 6.2: Energy and heat capacity for 1 and 50 items over 34 periods

The simulation for one item shows several maxima in the course of the heat capacity and corresponding to that the same number of plateaus for the energy. The reason for this is that the used data have discrete values like 350 or 400 and nothing in between. In contrast to this, the simulation for 50 items has a relatively smooth energy course, because those plateau effects disappear when the energy is built up by a larger number of items. A look at the heat capacity for 50 items illustrates that the energy is not perfectly smooth; the discrete values still play a role and generate a heat capacity with several maxima. The high peaks at lower temperatures are caused by the restricted computation time. From a physical point of view this is not perfect and perhaps more computation time should be invested to eliminate those high peaks. But from a practical point of view this is not necessary: the solution quality is similiar, even if quite smaller values for the lattice sweeps and the readings are used.

Another interesting picture is the energy course of the different terms of the Hamiltonian in Figure 6.3. The values of the penalty term run parallel to those of the return. That is clear, because here both variables are evaluated with the same factor. Besides they are correlated in general: each time when a stock-out is prevented and more items are sold, the return rises (the energy in the picture is negative, because of the minus sign in the Hamiltonian due to physical reasons). Remarkable is also that the "capital"-costs don't go down like the other terms. This means: the capital term cannot be optimised in the same way, because the decrease of the penalty term and the order term causes the increase of the capital term.

Stochastic Lead Time If we leave the standard configuration and introduce a stochastic lead time, the calculated policies naturally have to be different, because the restrictions are not the same and the phase space changes: the energy landscape is getting much more complex and thus it is more difficult for the algorithm to find the optimum. In contrast to the configuration with a constant lead time, the stochastic one produces more fluctuations in the simulation. That would be even worse, if in addition to the stochastic lead time a stochastic demand would have been used: then the algorithm has to work with changing conditions from one move to another. Here no stochastic demand was investigated, because normally there is no probability distribution available; if there is one, it is usually based on the historic data and therefore the simple data series is an acceptable approximation for the first step.

Capacity Restriction The third simulated configuration was the standard configuration with a capacity restriction. The sales figures are in monetary units; that is not a problem, because in reality they can be connected to the space of the

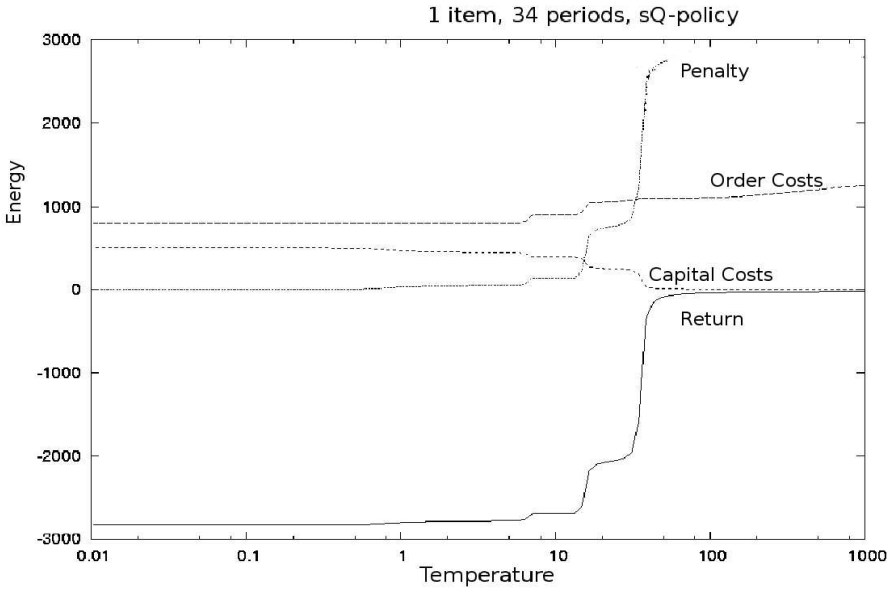

Figure 6.3: Energy of the different Sub-Hamiltonians

related items. Here no such data is available and thus the capacity restriction was formulated as budget restriction. That is another condition, but the simulation effects are the same and therefore this simplification is appropriate. Depending on the invested budget, a corresponding policy is determined. In contrast to the case without a budget restriction, some valleys of the energy landscape can not be reached, because there is not enough money available. Therefore during simulation the way through the phase space is also different, because not every configuration is allowed. Sometimes it makes sense for the simulation to set the budget higher than it is in reality: then more ways through the energy landscape are allowed and a better solution can be reached. The problem is that the generated solution at the end of the simulation is possibly not valid. But in some cases the end configuration is valid and thus it is worth trying it. Of course a lot of invalid solutions during the simulation are accepted; but from the practical point of view, just the final solution is essential.

Weighting-Factor Up to now each historic value was evaluated with the same weight. In section 5.1 it could be shown that it makes sense to weight the historic values due to their age: the older the data, the less they are considered. The forecast could be remarkably improved by weighting the historic data. Therefore the question arises, whether there could be an improvement in the determination of the order policy by weighting the data. The Hamiltonian stays principally the

same, but each term is weighted with a factor like in the exponentially weighted average of section 4.4.1. This means that the algorithm tries to find the optimal policy not by integrating all historic data in the same way, but by highlighting the values of the nearest past.

In the section about physical forecasting with a weighting factor, the result was a clear improvement of the forecast quality; the reason for this was the time correlation of the historic data. But for the determination of the best order policy the use of a weighting factor is of no positive use. Nevertheless it is interesting to show the course of the physical variables of the weighted simulation:

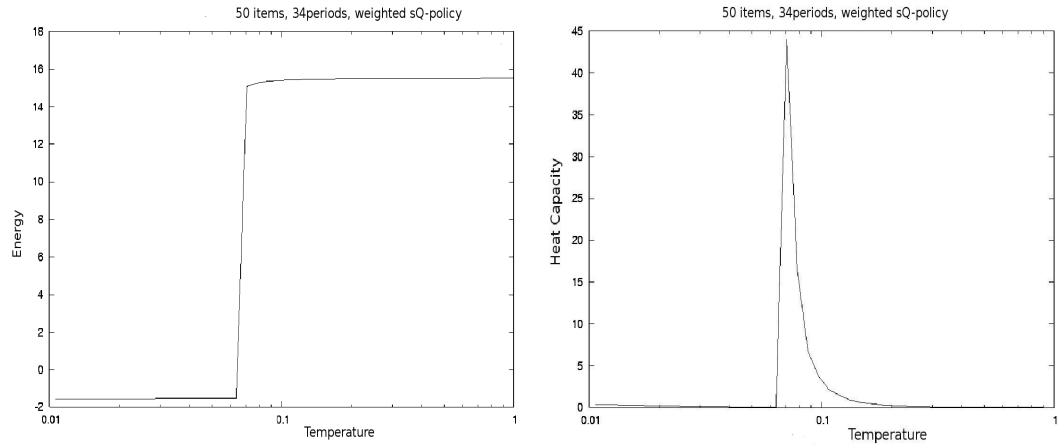

Figure 6.4: Energy and heat capacity of the weighted simulation for 50 items and 34 periods

The phase space is less complex, because mostly the last periods decide which configuration is optimal. Therefore also the energy and heat capacity have a smooth course without plateaus or peaks like in the normal simulation.

Threshold Accepting For threshold accepting (TA) as another optimisation algorithm the results are quite similiar to the simulation with simulated annealing (6.5). The results of the simulation with TA are a little bit worse than those with SA. Sometimes the same optimum is found, but often the energy at the end of the simulation is between 2 % and 5% higher; this means that in reality the return of the calculated strategy with TA is 2% to 5% lower than with SA. In this case the model has not such a complexity that it is necessary to use TA for reasons of computation time. If the model is extended and computation time is going to be a scarce factor, it can be good to work with TA and not with the laborious exponential function of SA.

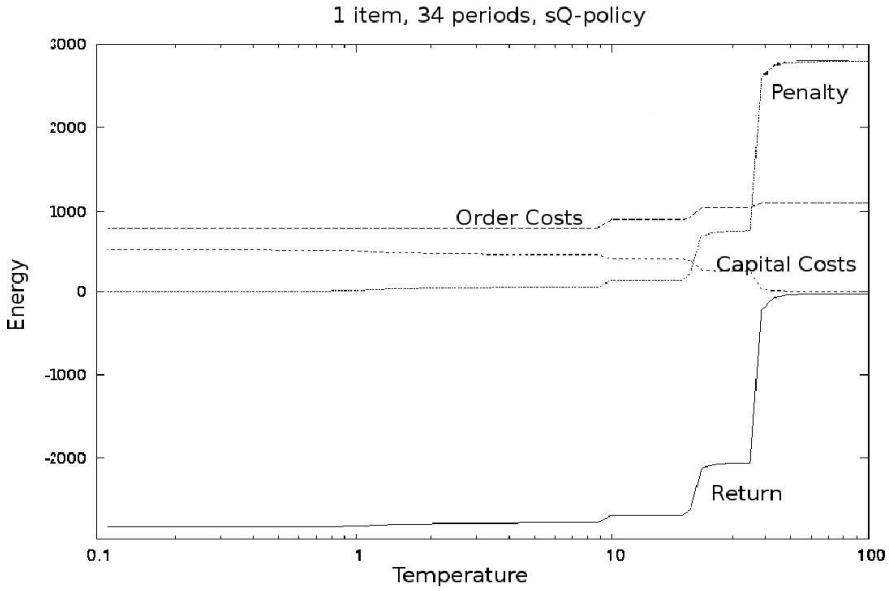

Figure 6.5: Energy of the different Sub-Hamiltonians with TA

6.2.2 (t,S) - Cycle Inventory Policy

This policy is easier to handle than the previous one, because there are fixed dates for an order. It is easier not only for the ordering company, but also for the supplier when he knows the date of an order. For the current simulation this policy has a lower performance than the (s, Q)-policy. But this doesn't mean that it is a totally bad policy. At first it has to be said that there still is a positive return due to historic data. Besides, some suppliers grant a discount for a periodical ordering. If the discount is high enough, this policy could be better than the re-order level inventory policies. Another discount could be granted, if different items from the same supplier are ordered at one time. This situation is going to be simulated in section 6.3.

From a physical point of view the following can be said about the simulation of a (t, S)-policy. The phase space is rather cliffy, because there are just discrete values for the re-order time: an order can only be stated every 1, 2, 3, ... periods and thus the simulation shows some special features. For the simulation of one item the best valley is easily found, because there are just a few possible re-order times and the fitting order quantity is fastly found. Therefore energy and heat capacity have the characteristics shown in Figure 6.6:

For the optimisation of many items at one time, the heat capacity has several singular peaks. The reason is similiar to those of the (s, Q)-policy: the values of the single items are different and thus they are not optimised at the same

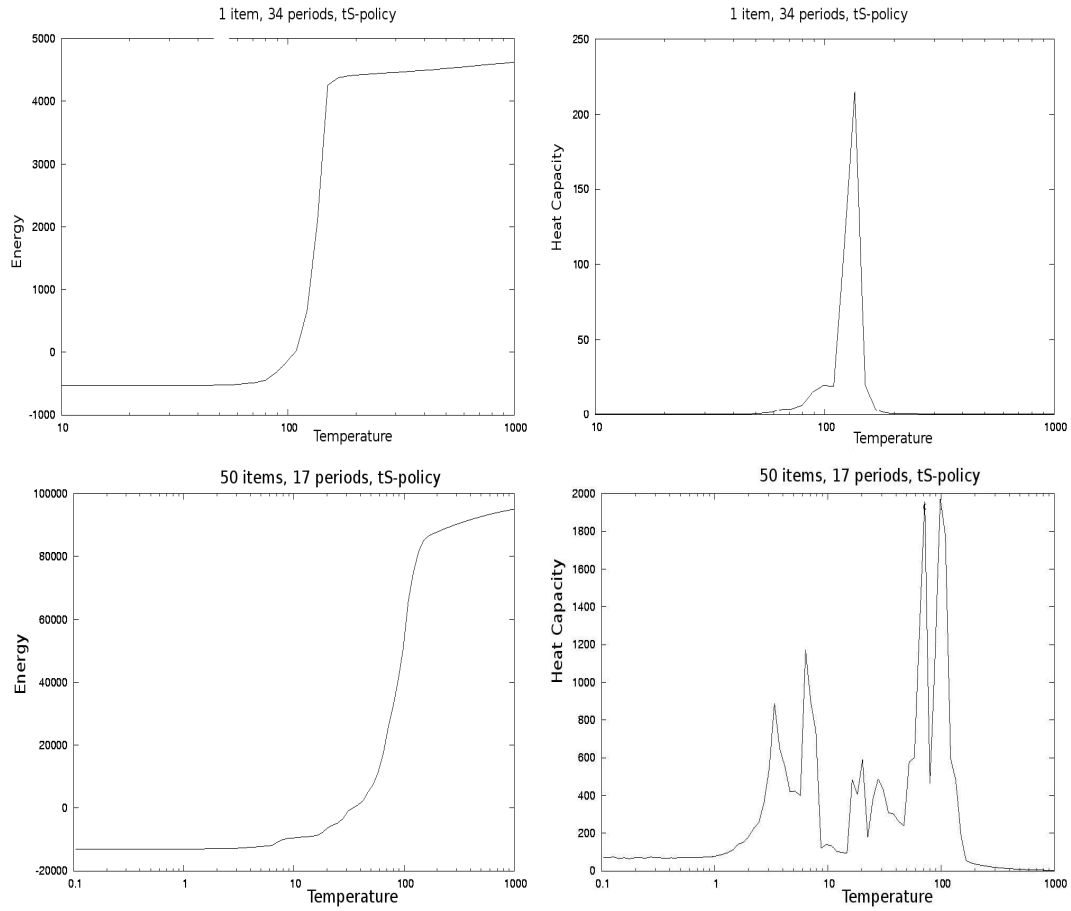

Figure 6.6: Energy and heat capacity of a (t,S)-policy for 1 item and 34 periods (above); Energy and heat capacity of a (t,S)-policy for 50 items and 17 periods (below)

temperature. But contrary to the (s, Q)-policy the (t, S)-policy has just a few single peaks, because of the less complicated phase space.

6.2.3 (s,S) - Level Inventory Policy

This policy is a combination of the (s, Q)-policy and the (t, S)-policy. Here an order is placed, when the stock falls under the safety line s. Besides there is no fixed order quantity Q, but a maximum stock of S, which is refilled when the safety line is reached. Surprisingly this proceeding shows the same performance of the introduced policies in relation to the future data (see subsection 6.2.4). The characteristics of the physical variables are similiar to those of the (s, Q)-policy and therefore shall not be presented once again at this point.

6.2.4 Application of the Different Policies to Future Periods

Each simulation tries to find the optimal policy in relation to historic data. The optimisation with historic data is no guarantee for a good solution in the future. It is always just an extrapolation and has to be used with caution. This can be illustrated by the application of the calculated policies to future data. Naturally the used future data are also historic ones, but if one half is taken for the simulation, the second one can be used as a test of the calculated policies. Here the policies are calculated with the historic data of 17 periods for 15 items and applied to the following 17 periods starting with the first three. The introduced policies have the "future" performance stated in Table 6.2.

Periods	Weighted sQ	sQ-Policy	tS-Policy	sS-Policy
3	2952.82	-1296.16	5385.72	-2326.07
4	4169.38	-946.57	6072.57	-2564.88
5	4916.81	-798.88	7436.49	-2654.65
6	5877.13	-139.97	7844.37	-2484.51
7	6675.31	198.69	9546.29	-2522.89
8	8545.46	-23.49	11708.56	-2717.80
9	9943.30	315.84	12872.39	-2646.12
10	10979.46	1921.54	16457.84	-1109.20
11	12646.03	2121.75	18172.33	-1544.80
12	13876.20	2305.20	19614.50	-1375.55
13	15649.49	3035.50	21524.80	-899.72
14	16210.29	3555.96	23694.23	-793.83
15	15681.70	4442.60	25823.99	-286.17
16	13545.93	4920.86	28257.58	-159.08

Table 6.2: Comparison of different policies

Of course these values cannot be taken as absolut values for another case, because simulation is not a black box: you are not allowed to put in some data and take out the results as good solutions. The results strongly depend on the used data and the parameters of the simulation (for example the order costs). Therefore always a previous test has to be made, when the policies are applied to a new problem.

Here two of the researched policies show positive results: the (s, Q)-policy and especially the (s, S)-policy. Both policies show a good performance for the near future of three to five periods. If the planning horizon is greater, the return is getting lower. Thus it is consequent to revise the policy after a few periods

and to calculate new values. The soccer grades are highly correlated in time and therefore it is difficult to determine a long term policy. For example if a soccer player makes a bad game and his selfconfidence is down, he probably plays bad in the following games, too; but perhaps a few periods later he has some luck and plays well again. This means in the language of the inventory problem that the sales figures are down and going up for unknown and seemingly random reasons.

The *weighted* (s, Q)-policy doesn't have good results. Actually it might be better to use the simple (s, Q)-policy with just a few historic values instead of introducing a weighting factor. The worst performance belongs to the (t, S)-policy. If a discount is considered and the data are different, the performance is better (s.section 6.3).

6.2.5 Sales Figures of a Steel Company

A more realistic data basis are the sales figures of a steel company. The algorithm is the same, just the data input is different. Here the figures are in kilogram over a time of 13 periods; one period is equivalent to two weeks. The fact that weight and not monetary units are used is no problem, because the cost parameters can easily be calculated in costs per weight unit. The sales figures of the single items have quite different values: some items are sold very often and in high numbers, others are less demanded. But the algorithm can easily cope with those circumstances.

In contrast to the previously used soccer grades (which are relatively homogeneous but quite stochastic), the steel data generate a wider energy space; therefore the moves (to search the energy space) have to be more sophisticated. And also the temperature range has to be rescaled because of the higher values. The rest of the physical simulation parameters is the same as before. A simulation run for one item shows similiar results, only the measurement scale is different (see 6.7).

The energy scale is too high, but due to the lack of real data for the economic variables (like capital and order costs per kilogram) the values of the previous simulation with soccer grades have been used. Thus at least a good ratio between the variables is guaranteed; the scale itself could be adjusted if real data are available. The application of the calculated policies to the future of the past for 50 items shows the results of Table 6.3.

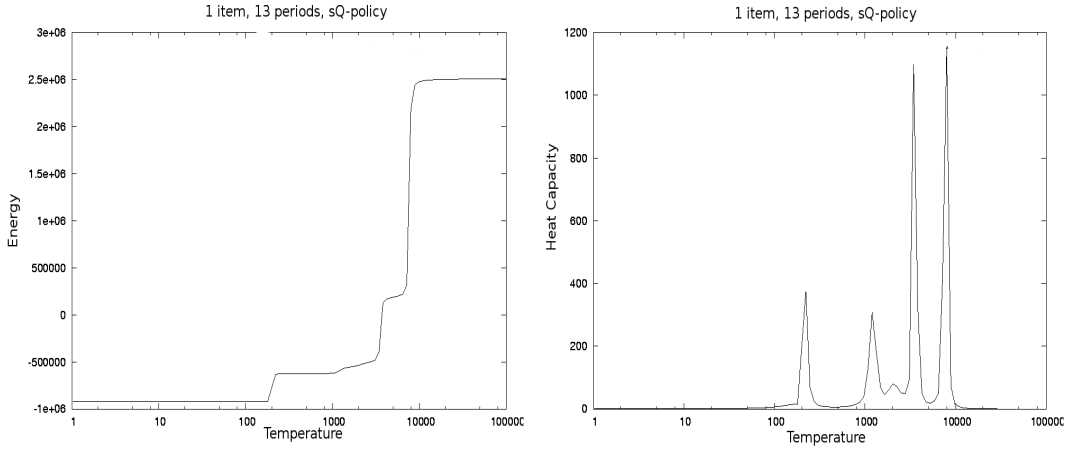

Figure 6.7: Energy and heat capacity of a (s,Q)-policy for 50 items and 17 periods

Periods	sQ-Policy	sS-Policy
3	-350006.78	-566024.16
4	-403731.26	-793079.83
5	-596814.99	-1002098.05
6	-700677.61	-1103598.31
7	-899527.25	-1304087.22

Table 6.3: Application of sQ- and sS-policy to future periods

In Table 6.3 only the (s, Q)-policy and the (s, S)-policy were used, because the basis of historic data is too short to calculate a proper (t, S)-policy: only seven periods are used to calculate a policy for the following six periods. In spite of the fact that just 6 periods have been used to calculate a policy, the results are very positive and not decreasing during the seven periods. Seemingly the steel data are more continuous in time and thus a calculated policy doesn't need to be revised as fast as for the soccer grades. Another interesting result is that here also the (s, S)-policy shows a better performance than the (s, Q)-policy. This could be a possible indication that the (s, S)-policy is better than the other ones, or at least more stable in relation to the most data types.

6.3 Inventory Optimisation - Part II

6.3.1 Implementation of further Parameters

So far the basic concepts of optimising an inventory model using a physical algorithm have been presented. Besides, a comparison between different policies was drawn and the characteristics of physical optimisation in the area of inventory control have been shown. In the following further parameters are integrated in the inventory model, in order to make it as realistic as possible.

Costs

A simple but necessary element of the simulation are costs. Up to now the storage and order costs for all items were assumed to have a certain value. Now there shall be different costs per item. In reality it would be a tough task to determine exactly the storage costs for each item, because those costs are indirect ones; normally just a good estimation can be made. For the simulation, random numbers are used to varify the costs per item. The proportion of storage and commitment costs (relative to the value of the stored items) is varied between $[0.03, 0.04]$, the penalty factor λ is between $[0.1, 0.7]$. The variation of the penalty factor can be explained by the fact that the stock-out of some items is less (more) problematic than for others. Beside this the order costs also have to be varied. They depend on the order quantity: the higher the order quantity the less the transportation and other costs. Therefore the order costs consist of a fixed part and a variable one, which decreases for rising order quantities: $c_L = a + x^b$ with the parameters $a \in [100, 200]$, $b \in [0.1, 0.3]$ and the variable order quantity x which shall be calculated in monetary units. This leads to another feature of a real inventory problem in the next subsection.

Discounts

The function of the order costs above implies discounts: if you order more at once you pay less. In a practical case, probably there will be staged order costs and not a continuous decrease of costs; for the model it is enough to have a function, which reproduces the basic principle. Another type of discounts are those, which include the order of several items from one supplier. The principle is the same as for single-item-discounts. Here the discount is modelled by the following function: $D = x^{0.6} \quad \forall \quad x \geq x_0$. x is the order quantity for a product group (in monetary units) and D the discount which is granted when x is above x_0. Of course the factor 0.6 is randomly chosen and will be more or less different in practice; x_0 is also a variable that has to be defined in practice and depends on the used data as well as the number of items. Beside the capacity restriction

and stochastic demand or lead times, discounts for groups of items represent the definite need for physical optimisation, because the optimal order policy of one item depends on those of the others. This shall be illustrated with an example: let there be 10 000 possible configurations for the optimal order policy of each item; the computation time for the total enumeration of all solutions of a **single** item is 1 minute. In order to determine the best policies of both items, each policy of one item has to be compared to 10 000 configurations of the other item. This means an enumeration of $(10000)^2 = 10^8$ configurations and a computation time of 10^4 minutes. For only 5 items the enumeration number would be 10^{20} and the computation time approximately 20 billion years. This clarifies the necessity of an alternative way of optimisation, for example with a physical algorithm like simulated annealing.

Discounts can have a great impact on the optimal order policy. [Has00] could show that it might be better to increase storage costs in order to realise discounts; therefore an integration of discounts is necessary.

Minimum Durabilities

Another often encountered matter are minimum durabilities. Some items have to be thrown away if they are stored too long; other items are sold long before the minimum durability runs out. Of course minimum durability is not an unusal trait in reality and so this aspect is considered: it shall be assumed that each item has a minimum durability between two and four periods.

Coincidence

Because lead times and sales figures are stochastic, most order policies have a safety stock. Nevertheless it is important to integrate coincidence in the determination of this variable. Small changes in the lead times can be processed by the optimisation algorithm, but bigger variances cause problems. Thus the following proceeding is possible:

1. Several data sets are produced by the quasi-continuous distribution of subsection 5.2. Besides, special dates for each item are randomly chosen at which the supplier cannot deliver.

2. For each data set and lead time configuration the optimal order policy is determined and afterwards applied to the other ones.

3. The returns of the application of the optimal policies to the different data sets and lead time configurations produce as many frequency distributions as data sets and lead time configurations are available.

4. Depending on the optimality criterion, the policy with the best return distribution is chosen.

The number of data sets for demand and lead times is for each in the order of 10^1; so in total the order is at most in the order of 10^2. The optimality criterion depends on the preferences of the decision makers: risk neutral ones for example could take the expectation value of the distribution; risk averse deciders will try to avoid losses or low returns and take the variance as decision criterion.

Hamiltonian

In total the Hamiltonian of Equation 6.1 has to be extended by three terms:

$$\begin{aligned}\mathcal{H} &= \mathcal{H}_{Return} + \mathcal{H}_{Storage} + \mathcal{H}_{Capital} + \mathcal{H}_{Order} + \mathcal{H}_{Penalty} \\ &\quad + \mathcal{H}_{Capacity} + \mathcal{H}_{Durability} + \mathcal{H}_{Discount}\end{aligned} \tag{6.7}$$

Here the terms for minimum capacity, durability and discount have to be included. In this case capacity is not a hard restriction as in section 6.2. This means that policies are possible which exceed the available stockroom to a small degree. In detail the sub-Hamiltonian for the capacity looks like:

$$\mathcal{H}_{Capacity} = \sum_{t=1}^{T} \Theta(S_t - S_0) \cdot (S_t - S_0) \tag{6.8}$$

$$\text{with} \qquad S_t = \sum_{i=1}^{M} (s^i \cdot {x_t}^i)$$

${x_t}^i$ is the available stock, s^i is the space, which is needed for one (monetary) unit of item i. For $S_t \leq S_0$ the summand of period t is zero, because the capacity restriction is fulfilled. In the other case the total Hamiltonian is increased with the difference between the available capacity and actual space of the items. The surplus has to be thrown away and a stockout is possible. The same will happen, if the minimum durability is exceeded:

$$\mathcal{H}_{Durability} = \sum_{i=1}^{M} \sum_{t=1}^{T} \sum_{j=1}^{N} \Theta(t - t_j - m^i) \cdot x^i_{t-t_j} \tag{6.9}$$

where $x^i_{t-t_j}$ is the stored quantity of item i in period t, which was ordered in period t_j ($t \geq t_j$); m^i is the minimum durability of item i, M the number of items and N is the number of orders. The third new sub-Hamiltonian describes the discount which is granted under the condition that a certain amount of items is ordered at once. For this the items have to be delivered from one supplier,

because otherwise there would be no discount. This kind of discount results from the fact that costs of transportation can be saved, if several items are ordered at one time.

$$\begin{aligned} \mathcal{H}_{Discount} &= -\sum_{t=1}^{T} \Theta(D_t - D_0) \cdot {D_t}^{0.6} \\ \text{with} \qquad D_t &= \sum_{i=1}^{M} {q_t}^i \end{aligned} \tag{6.10}$$

Above the limit D_0 a discount of ${D_t}^{0.6}$ for the total order is granted; the factor 0.6 is arbitrarily chosen and will be different from case to case. Another form of discount can be granted for the order quantity of a single item: in addition to a fixed amount of costs for an order, the variable part does not rise linear with the order quantity. The principle and reason is the same as above, realised in a slightly different way. In contrast to the Order-Hamiltonian of section 6.1 it now looks like:

$$\mathcal{H}_{Order} = +\sum_{i=1}^{M}\sum_{t=1}^{T} \left(c^i_{fix} + (q^i_t)^a\right) \tag{6.11}$$

q^i_t is the order quantity of item i at the end of period t, c^i_{fix} are the fixed costs of item i for each order; for a holds $a \in [0.1, 0.3]$.

6.3.2 Simulation Results

In order to show the basic characteristics of optimising a real inventory problem, it is enough to use 10 items with sales figures over 10 periods. A simulation could also be run with 20 items or 20 periods and the double computation time, but a compromise has to be made between the number of items/periods and the available computation time. If the number of items is too high, the course of the variables is pretty rough because of the limited computation time and thus it is difficult to give a physical interpretation of the results.

From a practical point of view there is no problem with this as long as the return is positive. But this study also deals with physical aspects of the simulation and therefore the number of items and periods was set to 10. According to the number of items a reasonable capacity restriction and discount-limit was chosen. It doesn't make sense to give a number for the reader, because the sales figures and the space of each item are random; but the numbers are chosen in such a way that they are restrictive and have influence on the simulation. Beside the number of items and periods the basic simulation variables have been set to the following

values: In a first simulation run the temperature range goes from $T_{Start} = 10000$ to $T_{End} = 0.1$. At each temperature step 1000 lattice sweeps were rejected in order to have an equilibrium; after that 10000 values were measured; the number of sweeps was set to 10. The error of the simulation results, due to this parameter values, is smaller than 10%. With this set of parameters the computation time was 2 hours on a *DELL OptiPlex GX745 DT*; this computer has a *Dual Core 3* processor with 3.4 GHz and a main memory of 1 GB. Energy and heat capacity of the three researched policies show the characteristics of Figure 6.8.

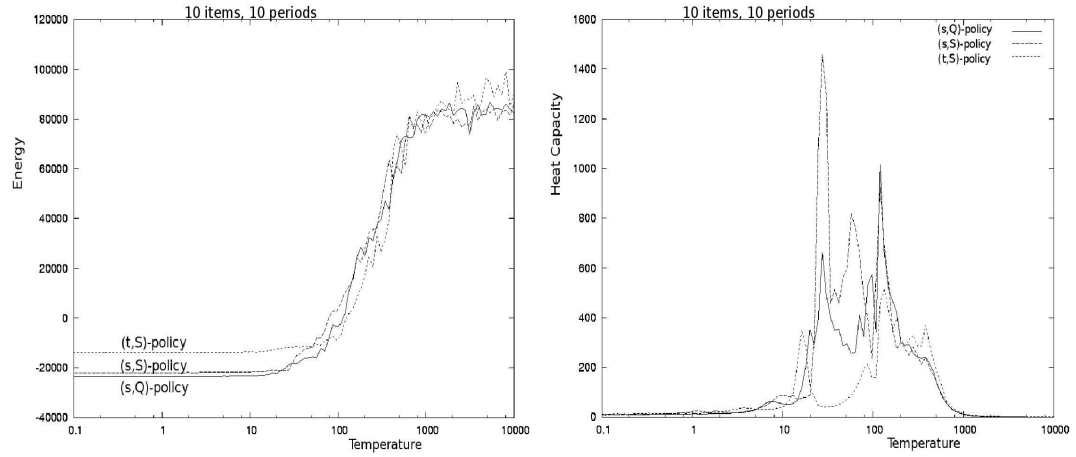

Figure 6.8: Comparison of different policies concerning energy and heat capacity

The developing of the different policies are relatively simliar, only the (t,S)-policy has a significantly higher energy at the end of the simulation; the (s,Q)-policy is the best one and slightly better than the (s,S)-policy. Of course there can be no general conclusion about the efficiency of the different policies: for other parameters the best policy can also be another one. But often the (s,S)-results are similiar to the (s,Q) ones and the (t,S)-policy is less good than the other ones. The (t,S)-policy is different insofar as the phase space is less complex than for other policies, because t is an element of $\mathbb{N}$, while s, S and Q are elements of $\mathbb{R}$. Thus there are fewer possibilities for finding the best solution and the energy is often higher (or the economic return is lower). Nevertheless the (t,S)-policy can be better for special sets of parameters.

The graphs of Figure 6.8 are not as smooth as usual, due to the reduced computation time and the big number of sub-Hamiltonians. Some peaks of the heat capacity disappear, when the computation time is enlarged; but others remain because the single sub-Hamiltonians differ in size and thus are optimised at a different temperature range.

This can be seen in Figure 6.9, where the energy and heat capacity of the

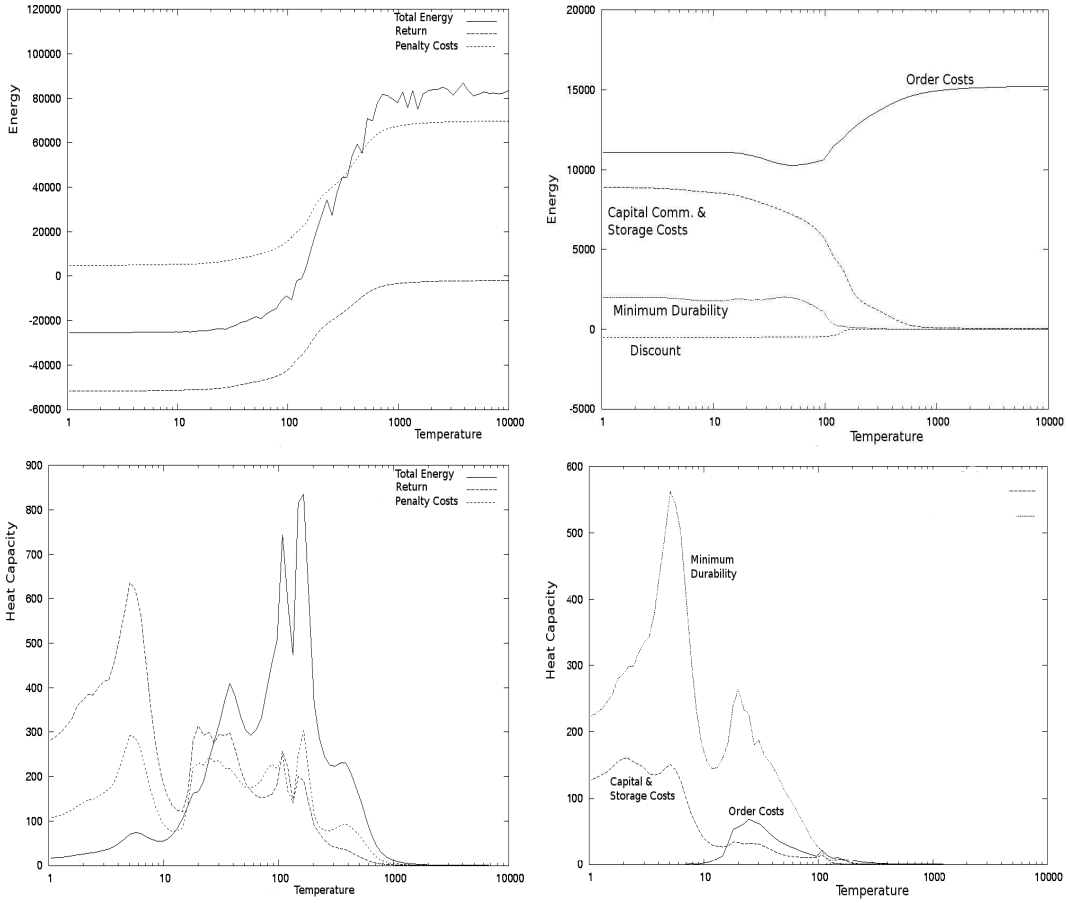

Figure 6.9: Components of energy and heat capacity concerning an (s,Q)-policy

different sub-Hamiltonians of a (s,Q)-policy are shown. In contrast to the previous optimisation run the measured values have been increased with a factor 10 (up to 100 000), the number of lattice sweeps from 10 to 50 for $10 \leq T \leq 1000$; the end temperature T_{end} was set to 1 in order to save some computation time in the temperature range with little turbulence; the total computation time for this setting was approximately 20 hours. The single heat capacities have several peaks because of their interaction with other terms of the Hamiltonian. For example the order costs are going down, because of the algorithm trying to rise the order quantity. But then the minimum durabilities are getting important and the algorithm has to find a compromise between the lowering of the order costs and the cost increase due to exceeding minimum durabilities.

In order to have an estimation of the value of a historically calculated policy, it has been applied to future periods of the past, just like in section 6.2.4. The application of a historically optimal (s,Q)-policy has the following results

depending on the type of the used data and the parameter constellation:

Application of a (s,Q)-Policy to Future Periods	Real Sales	Distribution_1	Distribution_10	Single Items
3	-7045.71	-5294.38	-7083.17	-9340.69
4	-6442.87	-5177.83	-7819.67	-2601.62
5	-7718.80	-5173.20	-8894.82	-3954.17
6	-9602.22	-4829.58	-10850.27	-5150.64
7	-6291.29	-5749.17	-10866.00	2746.50
8	-7308.35	-5778.44	-12578.19	2204.20
9	-5798.53	-5794.30	-15321.42	2251.25

Table 6.4: Comparison of optimal (s,Q)-policies (with different parameters) to future periods

"Distribution_1" means that based on a constructed frequency distribution of section (5.2) a data series of 10 periods is generated; therewith the optimal (s,Q)-policy is determined. Analogue "Distribution_10" means that ten data seriess are generated; then the optimal policies of the single series are determined and applied to the other nine. Depending on the criterion (here the median), the policy with the best value is chosen. In detail each policy has 10 possible returns; by comparison of the different medians the best one is taken. Alternatively the best or worst value of each *return-configuration* can be compared, depending on the preferences of the decision maker. "Real Sales" means that just the blank historic sales figures are taken for the determination of the optimal order policy. "Single" stands for an optimisation without capacity restriction and overall item discounts. In the short run a single-item optimisation can be better than the other ones, if the existing capacity restriction is not too harsh; but normally this is random and in the long run a single-item optimisation always turns out to be bad. In contrast to this the optimisation with constraints leads to stable (negative energy $\equiv$ positive return) values. In this case Distribution_10 leads to permanent improvements from period to period, whereas Real Sales and Distribution_1 are staying on the same level. Normally the application of calculated policies to future periods tends to result in this sequence: best is Distribution_10, then Distribution_1, Real Sales and Single as a bad solution. But this cannot be generalised for each set of data and parameters. The soccer grades for example fluctuate very strong and thus the optimisation with Real Sales leads to the best results, whereas the construction of frequency distributions is less optimal. Depending on the constraints, "Single" just randomly leads to good results. Therefore several tests of the underlying

data have to be made in order to determine the best proceeding. Anyhow it is clear that the application to future periods is not as good as the optimisation of historic values, because the future is never identical to the past. This can be seen in the next table:

Application of different Policies to Future Periods	(s,S)-policy	(t,S)-policy	(s,Q)-policy
3	-2409.74	-4266.24	-7083.17
4	-3152.61	-4610.28	-7819.67
5	-1450.66	-5818.31	-8894.82
6	-3417.60	-7460.31	-10850.27
7	-3410.32	-8159.12	-10866.00
8	-7308.35	-9457.35	-12578.19
9	-8406.89	-11142.62	-15321.42
Optimisation Results with 10 Historic Values	-23485.24	-13758.70	-21961.79

Table 6.5: Comparison of different policies to future periods and optimisation results

Here different policies are compared, which have been optimised with Distribution_10; for the application of the policies to future periods the return is always below the historic optimisation value. Sometimes the difference is small and sometimes it is relatively big, depending on the future developments. From a theoretical point of view it is hard to evaluate the economic impact of this. A perfect policy can't be determined, because of the unknown future; but if the results are acceptable for a *real* inventory, this has to be tested in practice with *real* parameters. That is an important result: it is always possible to determine the optimal policy of the past. If the sales don't fluctuate too strong, it can be highly assumed that the determined policy will lead to good results in the future. In the comparison above the (s,S)-policy is a little bit better than the (s,Q)-policy concerning the optimisation results. But in the application to future periods it is quite different: (s,S) is much below (s,Q) *and* also worse than (t,S). This means that the optimisation results are not necessarily the best indicator to show the best policies for the future. A (t,S)-policy is mostly very stable, because t can only take a few values and thus a determined policy has a high probability for acceptable returns in the future.

Conclusion

Due to the presented results it can be said that physical optimisation is definitely a valuable tool for helping to determine a good order policy. Of course the underlying circumstances have to be considered before determining the optimal policies. The main problem is the application to future periods: it is always uncertain, whether a historically optimal policy will be good in the future. Therefore the best proceeding would be to optimise a sales forecast. Forecasting is a highly researched area and many methods are available. Based on this, an optimal policy could be calculated by applying physical optimisation to an acceptable forecast. In the determination of the best order policy for given sales figures it will be hard for other methods to beat the physical optimisation algorithm.

The next step would be to optimise a real inventory. Therefore a cooperation with a company is necessary, in order to determine parameters like storage costs, order costs and anything else which is important for the optimisation. Then over a certain period of time a comparison has to be drawn between the current order policy and the one determined by physical optimisation.

6.4 Physical Structures in Inventory Control

After the basic explanations of spin glasses, physical and other optimisation algorithms, the theory of inventory control and the optimisation itself, the similiarities between the different areas of physics and inventory control shall be illustrated in detail.

6.4.1 Equivalence of the Systems

The energy function of a spin glass doesn't have a known and fixed ground state and can be compared to the objective function of an optimisation problem. In a combinatorial optimisation problem there is also a high number of local minima near the global optimum. In physics this means that the system is energetically degenerated. The ground state of spin glasses is degenerated, because of magnetic forces which operate against each other. But also an inventory problem has competing components: the customer service rises when the stock (and thus the capital committment) is high and vice versa. Therefore not every inventory strategy can fulfil the requirements of all components and frustration is the consequence. Frustration is one of the basic similiarities between spin glasses and inventory optimisation and leads to the degeneration of the ground state.

Other aspects are the manifold restrictions of an inventory problem like capacity or budget restrictions. These external influences of the inventory problem

are equivalent to an external magnetic field of a spin glass. Because of these equivalences the same optimisation algorithms can be used.

6.4.2 Optimisation Methods

Materials with a disturbed structure can be transferred to a perfect one by a slow annealing process; this was reproduced by Metropolis with a computer simulation. Out of this simulated annealing, threshold accepting and other algorithms have been developed to find the ground state of a solid state body. Beside crystals those methods were applied to spin glasses, whose ground state is unknown. Then 1983 Kirkpatrick et al. [KGV83] proposed to use the algorithms for economic optimisation problems, too.

Normal iterative methods that just accept improvements, strongly depend on the start configuration. In physics this corresponds to a fast annealing of a solid state body; thus the solutions are mostly bad and the system is fixed in a local optimum. In contrast to normal methods, physical optimisation algorithms can accept a local worsening at high temperatures and thus surmount an energy barrier. For low temperatures the probability to leave an energy barrier goes down. Because of the local worsening, physical optimisation algorithms have a greater capability to find the global optimum than the iterative methods of operations research. And the concept of constraints and penalties makes it possible to consider external restrictions in the simulation.

But the physical algorithms have further advantages: the measurement of physical variables can make a statement about the interaction of the different elements of the system; besides it can help to fix the system parameter, for example the weighting of the penalty function.

6.4.3 Equivalence of the System Variables

Fundamental Units of the System and Interactions The elementary unit of a spin glass is the spin. In the Heisenberg-model the spin can freely rotate in three dimensions; it is fixed by its x-, y- and z-component. In the inventory problem the elementary unit is the order of an item; for a complete description of the unit each parameter has to be known. In the case of a (s, Q)-policy for example, following has to be known: item, safety stock and order quantity. Both systems are built on a huge number of those elementary and interacting units. The magnetic exchange energy is caused by the RKKY-interaction. Depending on the distance of adjacent spins, the interactions trigger a parallel or antiparallel position of the spins. A measure for the alignment of the spins is the energy of the system; the energy is minimal, if the alignment fulfils all interactions.

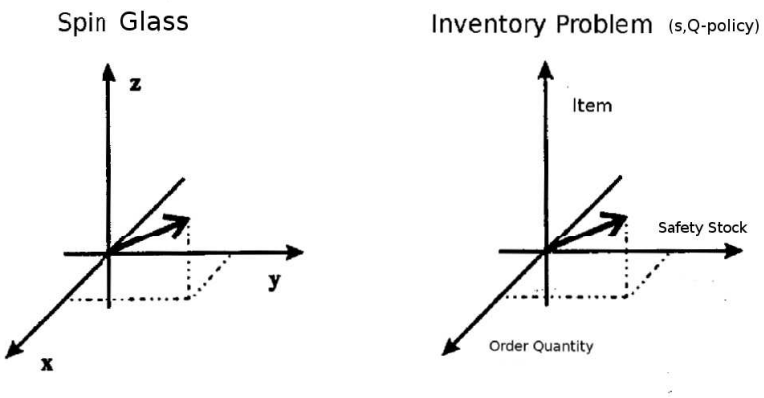

Figure 6.10: Equivalences between spin glass and inventory problem

The interaction in an inventory problem is caused by restrictions, which effect all items. Because of a capacity constraint for example, the space for one item is limited by those of the others. Thus the order quantity and order time of the different items influence each other. And also discounts for the collective order of different items cause interactions like in a spin glass.

Temperature For the spin glass and the inventory problem the temperature is a parameter, which controls the backward steps in the solution quality, in order to surmount local minima on the way to the global optimum. The temperature limits the freedom of the system: the higher the temperature, the more solutions are possible.

Energy and Objective function The probability of a worsening depends on its quantity, which has to be evaluated. For spin glasses the evaluation is given by the energy function, for the inventory problem by the objective function, which consists of several cost and penalty functions.

External Restrictions In the physical case the constraints are given by outer electric oder magnetic fields. Those fields influence the positions of the spins and thus the minimum of the energy. Economically those restrictions come from plausibility considerations. For example some items have to be stored in a full container and thus the order quantity has to be quantised; there can be several reasons for this quantisation, e.g. the handling. Furthermore capacity restrictions lead to constraints. They can be changed in the middle- or long-run, if they are internal; if they are exernal there is just an indirect influence and no change is

possible. The important aspect of those restrictions is that they strongly influence the solution space, just like external fields in a spin glass. The adherence of those constraints can be reached by the introduction of additional terms in the energy or objective function.

Configuration and Mutations A change of the system is caused by a so called mutation. For the Ising-model of a spin glass a mutation is a spin flip. For the inventory problem a mutation is the change of a parameter, which determines the problem. A change of the order quantity or the safety stock is a mutation. In both cases the mutation leads to a neighbour-configuration in the configuration space. The neighbourhood of configurations is based on their similarity, what means a wide compliance. In a spin glass a configuration is characterised by the alignments of the spins in the system. The configuration of an inventory problem is determined by the parameters. But there is one difference: in a spin glass each solution is allowed with different probability; in a combinatorial optimisation problem the solution space can be a sub-space of the configuration space and not every configuration is possible.

6.4.4 Differences

The differences between both systems lie in the purpose of the models and their details. The spin glass model is idealised, in order to get a manageable model despite the complex mechanisms. But the inventory problem has to be represented as exactly as possible, because the solutions of the simulation have to be realistic; otherwise there is no benefit.

Interval of the Possible Energy Changes The $\pm J$-model is a strongly simplified model of a spin glass: the values of the spins and the couplings are +1 or -1. Therefore the possible energy changes are restricted to a limited number. The inventory problem is very inhomogeneous, because the single parameters can have very different values. Thus the energy changes by one mutation can be in a great interval.

Significance of Non-Equilibrium Effects For spin glasses the thermal equilibrium is important to get reliable expectation values. The system size can be adjusted and extrapolated to infinite systems. But for an optimisation problem the system size is hardly changeable. In principle it is possible to divide the problem into subproblems; but only if a separation is possible, the optimised sub-solutions can be merged. From a certain system size non-equilibrium effects

cannot be prevented, because computation time is always limited. But it seems to be possible to produce good results without a thermal equilibrium.

Significance of Surface Effects In spin glass physics the systems have periodic boundary conditions, in order to simulate infinite systems and to prevent surface effects. In the inventory problem the frame has a strong influence on the system, because it is temporally restricted within an optimisation period. Therefore periodic boundary conditions are not possible.

Chapter 7

Physical Optimisation by Comparison

7.1 Genetic Algorithm

7.1.1 Implementation

In order to have a contrast to physical optimisation also a genetic algorithm was used to optimise the modeled inventory problem of sections 6.1 and 6.3. The model is quite similiar, but of course there are a few differences. At first the possible solutions of safety stock and order quantity are formulated as **binary code** in order to use the genetic operations of mutation and crossover successfully. Then the energy-function of physical optimisation has to be changed to a **fitness-function**. The two functions are identical for one solution, but a little bit different for the whole system, because genetic algorithms produce several solutions for one item. Beside this and the genetic algorithm itself the inventory model is the same; in a first step the (s,S)-policy of a "Standard Parameter Configuration" (6.1.3) shall be optimised.

7.1.2 Simulation Results

The simulation was executed for different parameters of the genetic algorithm; the results can be different depending on the observed variable. After a few simulations the mutation rate was set to 5%, because it doesn't influence the results too strongly, if it is not much *higher* or set to *zero*. For recombination a **2-point** crossover was used: one crossover for the coding of the safety stock and one for the order quantity. **Selection** as the third genetic operation was defined in two ways: **Roulette**- and $(\mathbf{N}, \mu)$-selection. Both selection methods have the same results; the Roulette selection needs just a few generations more to reach

the optimum. In order to preserve good solutions, the parents of one generation can survive, too; this means that they can reproduce themselves. The crucial point of the simulation is the coding of the solutions: if the range of possible values is too high, the algorithm has difficulties to find the optimum.

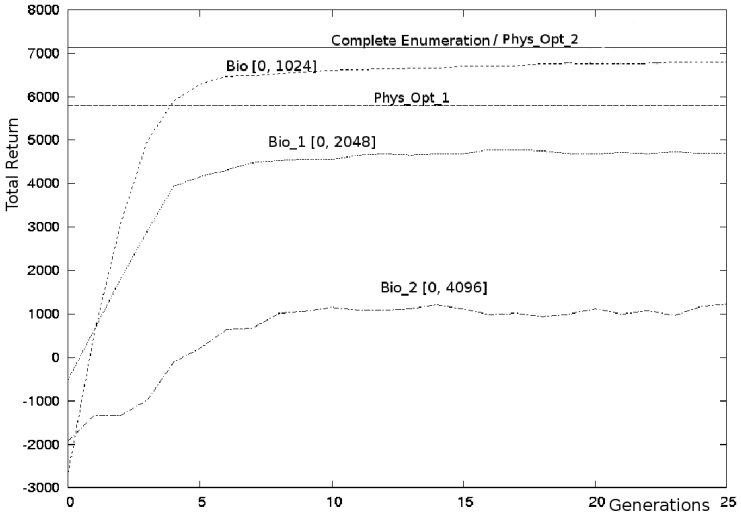

Figure 7.1: Comparison of complete enumeration with physical and genetic optimisation of a (s,S)-policy

In a first approximation the range for saftey stock and order quantity was set to $[0, 1024]$; this means that both variables were coded each with strings of 10 bits. In this case the results are very good and just 5% away from the global optimum. For the finding of this solution about 50 individuals per generation were reproduced 25 times. But if the solution space is enlarged and the binary code of the order quantity has just one additional bit, the reached solution is already 30% away from the global optimum; for two additional bits it is even 70%. This means that the GA gets trapped in a local optimum. A longer computation time with more generations and individuals per generation couldn't change this effect for the chosen implementation. In contrast to this the optimisation with a physical algorithm can reach different levels. This can have several reasons, but here two different kind of moves have been used: *Phys_Opt_1* is the result of an algorithm with a less sophisticated move; it mostly reaches a local optimum, which is 20% away from the the global one, irrespective of the extension of the solution space. *Phys_Opt_2* is another move which reaches the global optimum. The difference between both is that the steps of Phys_Opt_1 just have the same length, whereas Phys_Opt_2 combines smaller with bigger steps. Because of this Phys_Opt_1 is trapped in local minima; Phys_Opt_2 can escape and is able to

reach the global minimum. Figure 7.1 shows a comparison between the *level* of complete enumeration of all possible order policies, the genetic algorithm for a different range of order quantities and the *level* of physical optimisation; of course only the "genetic" graphs change with the generations.

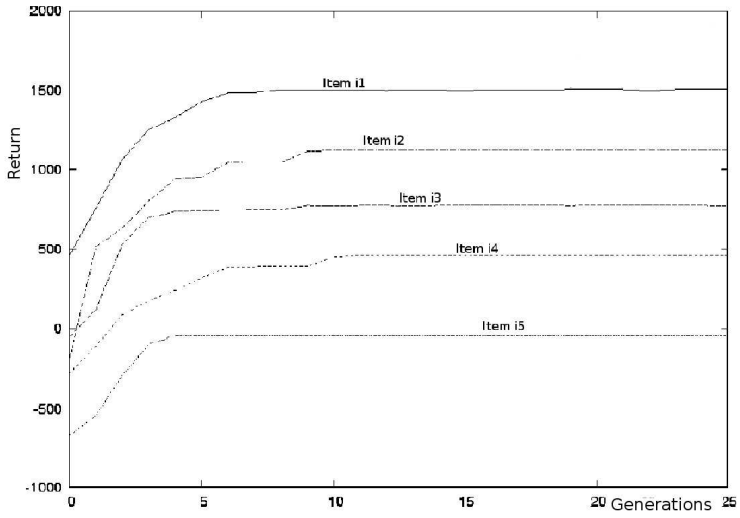

Figure 7.2: Return of different items

Figure 7.2 illustrates the development of different returns of a few items during the optimisation. It can be seen that the main part of the optimisation is done within the first 10 generations; after that there are just smaller improvements. This doesn't change much, when the inventory model of section 6.3 is optimised with a genetic algorithm. Beside the additional model parameters and the number of generations the algorithm is designed very similiar: 2-point crossover, roulette-selection, 50 individuals in one population, parents can survive one generation. In Figure 7.3 the results of the optimisation of a (s,Q)-policy with different sales figures are demonstrated: the physical and genetic optimisation of random sales figures on the left side is approximately on the same level for the best number of bits, which is here 14 ($\equiv [0, 2^{14}] = [0, 16384]$). For a higher number of bits, like 15 or 16, the solutions are degrading. On the right side in Figure 7.3 the simulation was done with soccer grades representing sales figures.

For those kind of data genetic optimisation could not reach the level of the physical one. It is difficult to give a final answer to this, because many are possible. Surely the structure of the soccer grades plays a role; maybe the physical algorithm can easier cope with different kinds of data. But for a detailed answer more investigations have to be done.

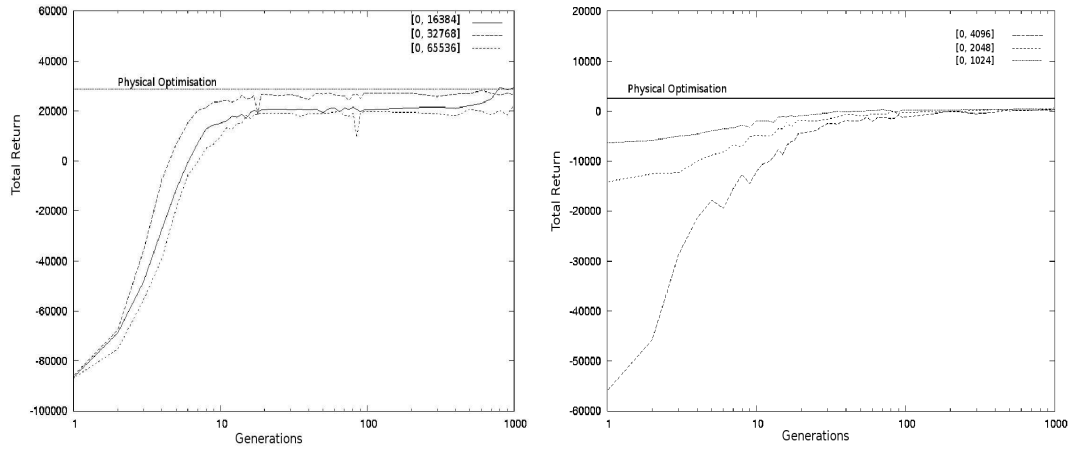

Figure 7.3: Comparison of physical and genetic optimisation of a (s,Q)-policy with random sales (left) and "soccer" sales (right)

7.1.3 A new Optimisation Algorithm ?

It might be possible to develop a new optimisation algorithm as a synthesis of physical and biological algorithm elements. For example a population of several solutions could be generated with simulated annealing and then used for the start configuration of a genetic algorithm. Another possibility would be to run a physical algorithm with binary coded solutions. Thus the genetic operations mutation and crossover could be used to walk through the phase space; maybe the results would be better.

[PR03] already tested a GA-SA hybrid algorithm for a lot sizing and scheduling problem. It works in two phases: in the first phase, the genetic algorithm generates the initial solutions (only once) randomly. The GA then operates on the solutions using selection, crossover and mutation operators to produce new and hopefully better solutions. After each generation the GA sends each solution to the SA (second phase) for further improvement. The neighbourhood generation scheme used in SA is a single-insertion neighbourhood scheme. Once the SA is executed for a solution of the GA, another solution of the GA is passed to SA. This process continues until all solutions of the GA in one generation are exhausted. Once the SA is executed for all solutions in one generation of GA, the best solutions of population size obtained from SA are the solutions of GA for the next generation. The GA and SA exchange continues until the required number of generations is completed. The results indicate that SA performs better than GA and the proposed GA-SA performs better than SA. The SA algorithm reaches a steady state after about two to fourteen generations. The proposed GA-SA hybrid algorithm reaches a steady state after about seven to eight generations.

7.2 Results of the Research Community

7.2.1 Overview

Single-item dynamic lot-sizing referring back to Wagner-Whitin and diverse lot-sizing heuristics are included in today's operations management textbooks and material requirements planning software systems. The interdependencies between multiple items are mostly considered in dependent demand systems like **material requirements planning** (MRP) and **distribution requirements planning** (DRP), which are essential modules in state-of-the-art advanced planning systems (APS) and have been among the earliest implementations of inventory concepts in enterprise resource planning. Lot-sizing models have attracted researchers for almost a century and many results of inventory research have been implemented in APS to resolve the trade-off between different types of costs.

Three main aspects of multi-product lot-sizing coordination are extensively discussed in the literature:

1. joint replenishment problems (JRP)
2. capacitated lot-sizing and scheduling problems
3. warehouse scheduling problems

For each of the three coordination problems, the literature can be classified along the main criteria

- discrete and continuous time
- deterministic and stochastic demand

Replenishment of multiple items from a single supplier is called **joint replenishment**. The savings realised by joint replenishment can be significant. The joint replenishment problem (JRP) is to determine an inventory replenishment policy that minimises the total cost of replenishing multiple items from a single supplier. The total cost depends on the cost of holding items in inventory, the cost of placing an order and the demand. The cost of placing an order includes a fixed cost of preparing an order and a handling cost associated with each item in the order. The problem is to find the optimal grouping of items for each order.

If the interaction between multiple products results from competition for a common and capacitated manufacturing facility, **lot-sizing and scheduling** problems have to be solved. There are many features of lot-sizing and scheduling problems and the manner in which they are treated by operational researchers; therefore many models exist with different features: for example the presence of single or multiple machines. In the latter case these can be parallel machines (in

a single stage), machines in sequence (i.e. multi stage) or even parallel multistage machines. The formulation can involve set-up costs and set-up times that can be fixed, or vary by product or be sequence-dependent. Another feature is the demand, which can be constant, or vary over regular periods, or vary over irregular periods. Due to the challenging research issues and the practical relevance in various industries, the majority of coordinated lot-sizing models and algorithms has evolved in this field.

If the inventories resulting from lot-sizes cope for limited warehouse capacity or a limited budget, the warehouse-scheduling problem has to be analysed. Many contributions for continuous time, constant demand models are straightforward extensions of the economic order quantity model with dedicated capacity and shared capacity with staggered orders. Staggering the replenishment times of lots is essential to unlock the benefits of sharing warehouse capacity across products where the capacity released by demands for one product can be used to accommodate inventories from the replenishment of other products. These benefits appear in contrast to warehousing strategies with space being dedicated to each product.

Only a few contributions have appeared that deal with the deterministic, discrete time, dynamic demand lot-sizing problem with shared capacity. For the special case of linear costs as considered for most practical applications, this property reduces to the well known zero-inventory property as in the Wagner-Whitin case where a product is only replenished if the inventory level equals zero and if units are replenished, the corresponding lot includes the demands of consecutive periods. Embedded in a LP-relaxation algorithm [DP90] suggests a smoothing method that firsts determines independent lot-sizing policies for the multiple items and in a second step remove infeasibilities by shifting replenishments. Apart from scheduling problems, metaheuristics that approach multi-item warehouses or inventories are rare, although it is a common problem in practice and only small instances of this problem can be solved with exact algorithms in reasonable time. Therefore a few results of optimisation with metaheuristics in this area are presented.

7.2.2 Different Papers

Fitness landscape analysis of dynamic multi-product lot sizing problems with limited storage. In the paper of [GRM06] the benefits of several mutation operators and that of recombination by means of a fitness landscape analysis are evaluated. The obtained results shall be useful for optimisation practitioners who design a metaheuristic for finite-horizon discrete-time lot-sizing with dynamic deterministic demand and a joint warehouse capacity constraint. This paper analyses the effectiveness of different mutation operators for multi-item

lot-sizing under warehouse capacity constraints. Further, the global structure of the search space is analysed in order to predict the problem difficulty for recombination-based search.

The results underpin the necessity to stagger lots when solving the lot sizing problem. Lot staggering can be realised by shifting all order periods of a single product back or forth a period. If capacity is highly constrained, fine-tuned changes should also be considered, e.g. by shifting ordering decisions or merging and joining consecutive orders. In the light of these findings, not only popular operators should be used; additionally the effects of problem specific mutation operators should be investigated. Further, the fitness distance analysis by random walks starting at local optima indicates that local minima are not randomly distributed in the search space.

Genetic Algorithm for Inventory Lot-Sizing with Supplier Selection Under Fuzzy Demand and Costs. In this paper of [RD06] a multi-period inventory lot sizing scenario, where there are multiple products and multiple suppliers, is solved with a real parameter genetic algorithm. It is assumed that demand of multiple discrete products is known (but not exactly) over a planning horizon and transaction cost is supplier dependent, but does not depend on the variety nor quantity of products involved and holding cost is product-dependent and there are no capacity restrictions and no backlogging is allowed. Because of uncertainties in demand and inventory costs, demand and all costs are considered as fuzzy numbers. The problem is formulated as a fuzzy mixed integer programming, converted and then solved with a Real Parameter genetic algorithm. The results determine what products to order in what quantities with which suppliers in which periods. The methodology can be extended with some modifications to the complicated inventory and supply chain models, i.e. models with deterioration, discount, variable replenishment, etc. formulated in crisp, fuzzy or fuzzy-stochastic environments.

Design of a Retail Chain Stocking Up Policy with a Hybrid Evolutionary Algorithm. [ARCSR06] address the joint problem of minimising transport *and* inventory costs of a retail chain that is supplied from a central warehouse. A hybrid evolutionary algorithm is proposed where the delivery patterns are evolved for each shop, while the delivery routes are obtained employing the multistart sweep algorithm. The experiments performed show that this method can obtain acceptable results consistently and within a reasonable timescale. The results are also of a lower cost than those obtained by other strategies employed in previous research. Furthermore, they confirm the interest of addressing the optimisation problem jointly, rather than minimising separately inventory and transport.

Genetic algorithm and Hopfield neural network for a dynamic lot sizing problem. [MJ06] addresses a dynamic lot sizing problem (DLSP) of a single item with capacity constraint and discount price structure. The general statement of the problem considers a situation where the demand is dynamic and deterministic, the storage capacity is limited and there is an overstock cost associated with the additional storage of the items. Here, the shortage of the item includes a high shortage cost, and purchasing cost includes the ordering cost and the discount rate. A dynamic programming (DP) algorithm is developed to derive the optimal solution, and the optimality of the GA and Hopfield neural network (HNN) are tested against DP. Although the well-known DP of Wagner-Whitin is capable of providing an optimal solution for single stage lot sizing problems, it suffers from its high computational complexity. Thus a genetic algorithm (GA) and HNN have been designed for DLSP to get best trade-off between solution quality and computational time.

DP, which follows an enumerative procedure, provides the optimal solution. But its procedural steps involve cumbersome computation when the size of the problem (either planning horizon or lot sizes) increases, thus limiting its application potential. The GA model for DLSP is capable of providing optimal or near optimal solutions with reasonable computational time. GA compared with HNN is far superior and closer to DP. The attempt made in this paper of [MJ06] provides a base for developing the HNN approach for lot sizing problems. A computational study shows that GA is capable of producing satisfactory results for various sizes of problems. HNN produces satisfactory results only for small size problems, and inferior solutions have been observed for large size problems. Experiment suggests that HNN model involves too many control parameters. Each one has its own range, depending upon its significance to the problem related with their energy components. Adjustment of HNN control parameters and combining HNN with simulated annealing and Boltzmann machines would improve the accuracy of the produced solution by a great extent.

Evolutionary optimisation of hedging points for unreliable manufacturing systems. In the paper of [MP05] an evolutionary stochastic optimisation procedure has been proposed to estimate the optimal hedging points (i.e. optimal inventory levels) for unreliable manufacturing systems producing either single product-types or multiple product-types under crisp-logic control. The methodology has been validated by comparing the hedging points produced by evolutionary algorithms with those obtained from the theoretical long-run solutions. It has been shown that the evolutionary stochastic optimisation procedure can be used to obtain prioritised optimal hedging points, i.e. hedging points when the cost weightings are different among the different products. The pro-

posed methodology is not restricted to unreliable manufacturing systems with exponentially distributed random machine failures and repairs, but is applicable to such random events with other distribution characteristics.

Evaluation of a (R,s,Q,c) Multi-Item Inventory Replenishment Policy through Simulation. The replenishment problem faced in the paper of [CM97] is stochastic in nature, with warehouse and transportation constraints present. Since several items are ordered at the same time, it is necessary to consider a (R, s, Q, c) model to find the solution. The (R, s, Q, c) model can be stated as: review the inventory level every R units of time, if the inventory is less than or equal to s you must-order Q; if the inventory is less than or equal to c you can-order (Q - c). The complexity of this multi-item inventory problem requires a fast and reliable method of determining the operating conditions that optimise the inventory control. Simulation techniques can be effectively used to determine an adequate ordering policy for this type of problems. Several ordering options were analysed and compared to find the policy that best accomplishes the firm's organisational objectives. The developed simulation model allows the dynamic change in the demand pattern for each item of the inventory. The results of these simulations were compared statistically and revealed that the implementation of the multi-item replenishment policy can reduce total investment and maximise customer service, while maintaining the business efficiency.

Application and Comparison of Physical and Conventional Optimisation Methods in the area Material Procurement. U. Gebauer [Ge97] compared threshold accepting (TA) to other methods for the optimisation of a (t, S)-policy with a fixed total order quantity. Following algorithms have been evaluated:

- Groff
- Part-Period-Balancing (PPB)
- Least-Unit-Cost (LUC)
- Silver-Meal
- Wagner-Whitin
- Savings

The solution structure of those algorithms depends on the used operation figures. In relation to the costs per period Groff, LUC, PPB and Silver-Meal have similiar good results. The Savings algorithm is worse than the others, because it tries to

consider the complete planning horizon and to reduce the order costs; but then the capital commitment costs rise and produce worse solutions. In contrast to this, Groff and Silver-Meal try to minimise the capital commitment costs and increase the order costs. A similiar solution structure can be found in all other methods, which try to find a balance between both possibilites. The solutions with the lowest costs are produced by Wagner-Whitin and threshold accepting.

Although the computation time of TA is small enough for the day-to-day business, those algorithms have a better performance in this special case of determining the optimal policy. But that's not a great advantage, because there are several disadvantages:

- The storage capacity cannot be included.
- The total order quantity is fixed and can't be changed.
- No stochastic lead times are possible.
- The often necessary order quantisation leads to a worsening of the solution.

There are a lot of other requirements, which cannot be included by those algorithms. Threshold accepting and physical optimisation algorithms in general can do this and that is their potency. Other methods like the Dixon algorithm and linear programming can consider the restrictions, but are not able to optimise the total order quantity and take a fixed value for their calculation.

Optimisation and Simulation. [BLR04] made a comprehensive analysis of metaheuristic optimisation methods. Particularly they applied and compared those methods to an inventory problem. At first they used traditional opimisation methods like the regression method or the pattern search method; each method has its special characteristics, e.g. relative to computation time and solution quality. The decisive disadvantage of those methods is that they can get stuck in a local minimum and have no chance to escape from it. Heuristics were the next generation of optimisation methods, which have to cope with the local minimum problem just like the traditional ones. Biethahn didn't deal with heuristics, but he analysed the following metaheuristics:

- tabu-search
- simulated annealing
- ant colony optimisation
- evolutionary algorithms

The basic conclusion of Biethahn is that there can be no definite decision about the best method. The best method strongly depends on the application. In the case of the single item inventory of Biethahn, simulated annealing shows good results with a low computation time. In any case metaheuristics can produce significantly better results than the traditional methods, because even complex solution spaces are no insurmountable barrier; and in contrast to the traditional methods the solution quality does not depend on the start configuration. But of course the implementation effort is higher.

Deriving inventory-control policies with genetic programming. [KT04] applied genetic programming (GP) to search for the optimal structure and the optimal parameters of inventory control policies. For the relatively simple single-echelon deterministic-demand setting, GP was proved to be capable of finding the optimal policy and the optimal parameters of the policy; i.e., GP rediscovered the economic order quantity formula. For the moderately complex single-echelon stochastic-demand setting, GP was able to identify the optimality of the (s,Q)-policy and to find closed-form heuristics that outperform other state of the art closed-form heuristics for the range of parameter values analysed. For the relatively complex multi-echelon stochastic-demand setting, GP found some elements of the optimal policy and found heuristics that outperform heuristics developed by traditional approaches for the range of parameter values analysed.

7.2.3 Mathematical Methods

Mathematical models like linear programming, integer programming and relaxed versions of integer programming with four and five time periods have been addressed in the literature for the lot sizing problem. The application of the mathematical model is limited to small sizes in the case of problems using a deterministic model. The stochastic nature of the controlling parameters in the dynamic lot sizing problem (DLSP) limits the application of mathematical models to the larger size problem.

Inventory-control policies are typically derived analytically, and this requires advanced mathematical skills and can be quite time-consuming. Despite this and the often limited practical relevance, nearly all research in determining the best inventory policy is done with mathematical methods; especially the (s,S)-inventory system is one of the main issues. There have been several studies on the determination of s and S variables. Veinott and Wagner studied the single item inventory model considering setup cost, demand lead times, and discount rates by a large order. They demonstrated the optimality of the proposed model and the methodology to decide s and S variables. Sivazlian proposed an algorithm to minimise the total cost consisting of inventory cost, ordering cost, and shortage

cost in assumption that demand is the Gamma distribution in the single item inventory system. The values of s and S were found using the graph. Snyder assumed that demand is the normal distribution in the single item inventory system.

Most of the previous works for the (s,S)-inventory system (and generally in the area of inventory control with mathematical methods) were studied in the single-item system. Just a few researchers deal with multi item inventories. [AS06] for example proposed a a multi-item ordering model in the (s,S) inventory system. Thereby an order range was introduced and its effectiveness demonstrated by numerical experiment.

7.2.4 Delineation

Before the development of metaheuristics just mathematical methods and heuristics have been used to optimise an inventory system. Nowadays metaheuristics generally have a great proportion in optimising different types of problems, because of their flexible adaption to complex situations. In lot sizing and scheduling of single- and multi-echelon inventories metaheuristics already have been proven to be efficient and useful. Thereby mostly evolutionary algorithms are used; simulated annealing or tabu search are less represented.

However, in the determination of order policies metaheuristics are scarcely established. Just a few authors like [BLR04] and [KT04] make some research in this area. Predominantly, mathematical methods are used with rather restrictive assumptions like special stochastic distributions; besides, their application to practical problems is very narrow due to the complex calculation.

In this dissertation a quite realistic model of a dynamic multi-item inventory with discounts, capacity restrictions, customer service, minimum durabilities, order and storage costs has been developed and afterwards different inventory policies have been optimised with simulated annealing and a genetic algorithm. Also interdependencies between the different items have been considered. The policies and metaheuristics have been compared to one another.

Thus the contribution of this work to current research is *the application of metaheuristics for determining the optimal order policy in a multi-item inventory under widely realistic constraints*. It could be proved that simulated annealing and the genetic algorithm have *respectable* and *comparable results* and can be applied in practice. Also the dependencies of the optimisation on the available data could be shown.

Chapter 8

Summary

This dissertation is the attempt to optimise an inventory system with physical methods. It mainly deals with the following aspects in simulating and optimising the economic problem of an inventory process:

1. Formulation of the mathematical model
2. Development of the computational program
3. Analysis of the theoretical results

Unfortunately not many real data have been available and thus at least another work is necessary to verify practically the value of the achieved results in co-operation with a company. But from a theoretical point of view the results of this work are mostly positive. Thus physical optimisation definitely is a useful tool in the area of inventory control; it consists of two main parts: the forecasting of future demand and the determination of the most efficient order policy. Due to this the dissertation has been structured in the same way. At first a forecasting model was developed and tested; then a inventory model (containing the basic features) was contructed and also checked. The fundamental results can be summarised as follows.

8.1 Forecasting

The smaller and less successful part of this dissertation is about forecasting. It was possible to make a good short term forecast for multiple items in consideration of capacity or budget restrictions; but a medium term forecast considering correlations between data series failed, because the correlations didn't have a great effect on the forecast. The short term forecast was made for soccer grades and compared to the moving average and the exponentially weighted average as

two practical standard methods of forecasting. The comparison has shown that the different physical models are better or equal to the compared standard methods. Beside the integration of capacity restrictions, another model was developed that additionally tries to minimise the risk of high deviations in the forecast. Altogether a short term forecast with special constrainsts is accessible for physical optimisation.

For the medium term forecast a distribution based on historic values was developed. The distribution was used for generating future values as forecast; after that the correlations of the future sales figures have been optimised with a physical algorithm. For the highly correlated soccer grades it could not be proved that the optimisation of correlations has any effect on the forecast. The constructed distribution has shown some value in inventory optimisation, but is of no use in forecasting.

8.2 Inventory Optimisation

The main part of this dissertation is the optimisation of a widely realistic multi-item-inventory by different order policies. In a first step a simplified inventory model was implemented with returns, order-, storage-, capital commitment- and penalty costs. For this "standard model" (without constraints considering all items) physical optimisation is not absolutely necessary, because the most important configurations simply can be enumerated in special cases. But before further aspects have been implemented it could be tested, whether the algorithm is working. After this the model was more and more enlarged and adjusted to reality. At first a capacity restriction was introduced; that makes it already impossible to optimise the problem with standard methods, because of the large and discrete number of possible solutions. Furthermore discounts and minimum durabilities have been established. In varying constellations all those parameters are the kernel of the underlying inventory model. Then the best way to refill the inventory is determined. Therefore three different standard order policies have been used: sS, tS and sQ (s: safety stock, S: upper order limit, Q: order quantity, t: re-order period). Each policy was applied to different types of data and mostly optimised with simulated annealing (SA) as the best physical algorithm. Beside SA threshold accepting (TA) and a genetic algorithm (GA) was programmed for optimising the order policies. The overall proceeding is shown in Figure 8.1.

The simulation results of this proceeding are very good and thus it can be summarised that physical optimisation and other metaheuristics are a useful tool for the organisation of an inventory. Of course there are differences depending on the used data, order policies, parameters and algorithms itself. Thus there have been no problems in optimising random sales figures or those of a steel company.

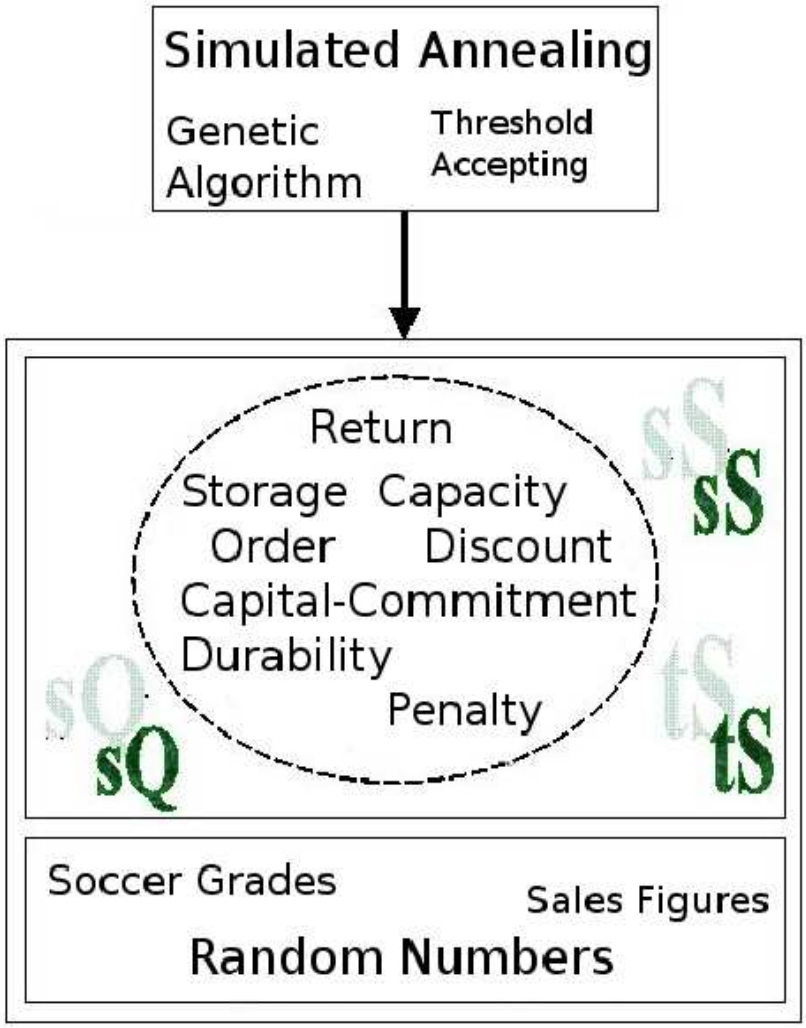

Figure 8.1: Scheme of proceeding

But soccer grades are highly fluctuative and thus it is harder to determine a good policy. The quality of the different order policies depends on the set of parameters: sometime the sS-policy is the best, sometimes one of the others. There is no basic rule, which decides what policy is better in a special case; it only can be said that a tS-policy is less good than the others. But that is only true for historic values ! If the historically determined policy is applied to future periods, the tS-policy often turns out to be equal to the others, sometimes it is even better. Therein lies another important point of the analysis: it is no problem to determine the best policy afterwards; but it strongly depends on the concrete situation, whether the old policies are optimal in the unknown future.

Thus it is necessary to re-calculate the policy after a while and to support it by other considerations like future developments of the market. In case of the random numbers and the steel company the application to future periods resulted in positive return, sometimes as high as in the past. But for the soccer grades, the return was always negative in the long run; in short or middle term the values have been mostly acceptable. This illustrates again the necessity of a permanent review. Most of the simulation was executed with simulated annealing. Threshold accepting is a little bit faster, but also 5 % worse in the results. Because computation time mostly is not too scarce, SA should be used as the standard algorithm in optimising an inventory problem. For a good solution, several hours of computation time are enough; more time improves the results, but only slightly. In order to have a comparison to physical optimisation, a genetic algorithm (GA) was introduced in section 7.1. The results are on a similiar level, sometimes slightly below the physical algorithm. For future studies it would be interesting to know more about differences and similarities between SA and GA. Also other meta-heuristics like tabu search or ant colony optimisation could be applied to this inventory problem and compared afterwards. Finally analogies between spin glasses and inventory optimisation have been identified in 6.4.

8.3 Conclusion

In total the following conclusion can be drawn: As stated above, physical optimisation is at least a very useful tool in determining the optimal order policy of an inventory. The cost parameters can be varied in simulation and thus it can be examined how the order policy will change. For example a company can find out by simulation, whether a bigger inventory is profitable. Physical optimisation is less useful in forecasting; in some cases it could be applied, but normally there are other forecasting methods available, which have the same or a better quality. Apart from the examined areas, a physical algorithm might be used to optimise different established forecasting methods. Those methods often have several parameters, which have to be tuned in relation to the available data: the different methods are tested with alternating parameters and then the best one is taken. The task to determine the best forecasting method is highly complex and thus physical optimisation could make a valueable contribution.

For the complete process of an inventory system it would be the best to make a medium term forecast with the best fitting forecasting method at first. Then the optimal policy for this forecast can be determined with simulated annealing.

Bibliography

[ARCSR06] A.I. Esparcia-Alcazar, L. Lluch-Revert, M. Cardos, K. Sharman, C. Andres-Romano: *Design of a Retail Chain Stocking Up Policy with a Hybrid Evolutionary Algorithm*, in *Evolutionary Computation in Combinatorial Optimization*, Springer Verlag, Berlin/Heidelberg, 2006

[Ar04] J.S. Armstrong: *Principles of forecasting*, Kluwer, Boston et al., 2004

[AS06] B. Ahn, K.-K. Seo: *A multi-item ordering model in the (s, S) inventory system*, in *The International Journal of Advanced Manufacturing Technology*, Springer Verlag, London, 2006

[Ba98] P. Baumgartner: *Vergleich der Anwendung Neuronaler Netze und Genetischer Algorithmen zur Lösung von Problemen der Finanzprognose*, Universität St. Gallen, Dissertation, 1997

[BB89] D. Bartmann, M.J. Beckmann: *Lagerhaltung*, Springer Verlag, Berlin Heidelberg New York, 1989

[Bi99] J. Biethahn: *Simulation als betriebliche Entscheidungshilfe*, Physica-Verlag, Heidelberg, 1999

[BLR04] J. Biethahn, A.Lackner, M.Range: *Optimierung und Simulation*, Oldenbourg Verlag, München Wien, 2004

[BH02] K. Binder, D.W. Heermann: *Monte Carlo Simulation in Statistical Physics*, Springer Verlag, Berlin-Heidelberg, 2002

[BR01] R.J. Brooks, S. Robinson: *Simulation - Inventory Control*, Palgrave, Operational Research Series, New York, 2001

[Br98] J. Britze: *Anwendung von Methoden der Statistischen Physik auf Optimierungsprobleme der Materialplanung*, Universität Regensburg, Diplomarbeit, 1998

[CM97] C. Cerda, A. Espinosa de los Monteros F.: *Evaluation of a (R,s,Q,c) Multi-Item Inventory Replenishment Policy through Simulation*, Proceedings of the 1997 Winter Simulation Conference, 1997

[Co85] R.W. Cottle: *Mathematical programming essays in honor of George B. Dantzig*, North-Holland, Amsterdam, 1985

[CS03] Y. Collette, P. Siarry: *Multiobjective Optimisation*, Springer Verlag, Berlin Heidelberg, 2003

[DD04] W. Domschke, A. Drexl: *Einführung Operations Research*, Springer Verlag, Berlin-Heidelberg, 2004

[DP90] P.S. Dixon, C.L. Poh: *Heuristic procedures for multi-item inventory planning with limited storage*, IIE Transactions 22, 1990

[DPST06] J.Dreo, A. Petrowski, P.Siarry, E.Taillard: *Metaheuristics for Hard Optimization*, Springer Verlag, Berlin Heidelberg, 2006

[DS90] G. Dueck, T. Scheuer: *Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing*, J.Comp.Phys. **90**, 161, 1990

[DoS04] M. Dorigo, T. Stützle: *Ant Colony Optimization*, The MIT Press, Cambridge Massachusetts, 2004

[Fr00] P. Francois: *Flexible Loßgrößenplanung in Produktion und Beschaffung*, Physica Verlag, Heidelberg, 2000

[Fre07] P. Frerichs: *persönliches Gespräch mit P. Frerichs (INFORM GmbH)*, Aachen, April 2007

[FH91] K.H. Fischer, J.A. Hertz: *Spin glasses*, Cambridge University Press, Cambridge, 1991

[Ge97] U. Gebauer: *Anwendung und Vergleich physikalischer und herkömmlicher Optimierungsverfahren im Bereich der Materialbeschaffung*, Universität Regensburg, Diplomarbeit, 1997

[GG98] R. Gabriel, P. Gluchowski: *Management Support Systeme*, Studienkurs FernUniversität Hagen, 1998

[Go89] D.E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Weasley, New York Sydney, 1989

[GRM06] J. Grahl, A. Radtke, S. Minner: *Fitness Landscape Analysis of Dynamic Multi- Product Lot-Sizing Problems with Limited Storage*, University of Mannheim, Department of Business Administration and Logistics, Technical Report, 2006

[Has00] K. Haase: *Kapitalwertorientierte Bestellmengenplanung bei Mengenrabatten und dynamischer Nachfrage*, Christian-Albrechts-Universität zu Kiel, Institut fur Betriebswirtschaftslehre, Lehrstuhl fur Produktion, 2000

[Ha00] R. Hackl: *Optimierung von Reihenfolgeproblemen mit Hilfe Genetischer Algorithmen*, Universität Regensburg, Dissertation, 2000

[He81] A. Herbein: *Kostenoptimale Bestellpolitiken im Mehr-Produkt-Lager*, Universität Augsburg, Dissertation, 1981

[Ho67] D. Hochstädter: *Stochastische Lagerhaltungsmodelle*, Springer Verlag, Berlin Heidelberg, 1969

[Ho92] J.H.Holland: *Adaption in artificial and natural systems*, The University of Michigan Press, 1992

[HR02] A.K. Hartmann, H. Rieger: *Optimization Algorithms in Physics*, Wiley-VCH Verlag, Berlin 2002

[KGV83] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi: *Optimization by Simulated Annealing*, Science **220**, 671, 1983

[Ki94] W. Kinnebrock: *Optimierung mit genetischen und selektiven Algorithmen*, Oldenbourg Verlag, München, 1994

[Ko93] K. Kopitzki: *Einführung in die Festkörperphysik*, Teubner Studienbücher Physik, Stuttgart, 1993

[KR96] W. Kinzel, G. Reents: *Physik per Computer*, Spektrum Akademischer Verlag, Heidelberg, 1996

[KT04] P. Kleinau, U. W. Thonemann: *Deriving inventory-control policies with genetic programming*, in *OR Spectrum*, Springer Verlag, Berlin Heidelberg, 2004

[LA92] P. J. M. van Laarhoven, E. H. L. Aarts: *Link Simulated annealing : theory and applications*, Dordrecht u.a, Kluwer, 1992

[MA05] P. Mertens, S. Albers: *Prognoserechnung*, Physica Verlag, Heidelberg, 2005

[MJ06] N. Megala, N. Jawahar: *Genetic algorithm and Hopfield neural network for a dynamic lot sizing problem*, in *The International Journal of Advanced Manufacturing Technology*, Springer Verlag, London, 2006

[Mo87] I. Morgenstern: *Spin glasses, Optimization and Neural Networks*, Springer Verlag, Berlin-Heidelberg, 1987

[MP06] P.Y. Mok, B. Porter: *Evolutionary optimisation of hedging points for unreliable manufacturing systems*, in *The International Journal of Advanced Manufacturing Technology*, Springer, London, 2006

[My93] J.A. Mydosh: *Spin Glasses*, Taylor & Francis, London, 1993

[No02] W. Nolting: *Grundkurs Theoretische Physik, Band 6, Statistische Physik*, Springer Verlag, Berlin-Heidelberg, 2002

[Nu93] K.J. Nurmela: *Constructing Combinatorial Designs by Local Search*, Research Report Ser.A No.27, Helsinki University of Technology, 1993

[Ph00] D.T., Pham: *Intelligent optimisation techniques*, Springer Verlag, London et al., 2000

[PR03] S.G. Ponnambalam, M.M. Reddy: *A GA-SA Multiobjective Hybrid Search Algorithm for Integrating Lot Sizing and Sequencing in Flow-Line Scheduling*, in *The International Journal of Advanced Manufacturing Technology*, Springer, London, 2003

[Qu82] A. Quint: *Simulation und Optimierung eines stochastischen Lagerhaltungsmodells*, Johann Wolfgang Goethe Universität Frankfurt am Main, Dissertation, 1982

[RD06] J. Rezaei, M. Davoodi: *Genetic Algorithm for Inventory Lot-Sizing with Supplier Selection Under Fuzzy Demand and Costs*, in *Advances in Applied Artificial Intelligence*, Springer Verlag, Berlin Heidelberg, 2006

[Re73] I.Rechenberg: *Evolutionsstrategie*, Frommann-Holzboog Verlag, Stuttgart, 1973

[Re95] C.R. Reeves: *Modern Heuristic Techniques for Combinatorial Problems*, McGraw-Hill International, London, 1995

[RKR02] W. Rödder, F. Kulmann, H.P. Reidmacher: *Optimierung mit intelligenten Strategien*, Studienkurs FernUniversität Hagen

[Rö99] W. Rödder: *Planungs- und Entscheidungstechniken*, Studienkurs FernUniversität Hagen, 1999

[RMM04] W. Rödder, H. Müller-Merbach: *Einführung in des Operations Research*, Studienkurs FernUniversität Hagen, 2004

[SBK93] H.P. Schwefel, T. Bäck, F. Kursawe: *Naturanaloge Verfahren. Grundlagen und praktischer Einsatz in der Optimierung*, Tutorium Universität Dortmund, 1993

[SC01] A.T. Staggemeier, A.R. Clark: *A Survey of Lot-Sizing and Scheduling Models*, 23rd Annual Symposium of the Brazilian Operational Research Society, Campos do Jordão, Brazil, November 2001

[Sc81] H.P. Schwefel: *Numerical Optimization of Computer Models*, Wiley, Chinchester, 1981

[Sn99] J. Schneider: *Effiziente parallelisierbare physikalische Optimierungsverfahren*, Universität Regensburg, Dissertation, 1999

[Sw96] C. Schneeweiß: *Lagerhaltung*, Studienkurs der FernUniversität Hagen, 1996

[St93] D. Stein: *Spingläser*, aus "Chaos und Fraktale", Spektrum der Wissenschaft, Spektrum Verlag, Heidelberg, 1993

[Sy89] G. Syswerda: *Uniform Crossover in Genetic Algorithms*, Internat. Conf. on Genetic Algorithms, Morgan Kaufmann Publishers, 1989

[SS01] R. Schlittgen, B. Streitberg: *Zeitreihenanalyse*, Oldenbourg Verlag, München Wien, 2001

[Te03] H. Tempelmeier: *Material-Logistik*, Springer Verlag, Berlin u.a., 2003

[VS98] G. Vogel, I. Szasz: *Dtv-Atlas Biologie*, Dt. Taschenbuch Verlag, München, 1998

[Wi02] A. Wimmer: *Anwendung physikalischer Optimierungsverfahren auf zeitlich veränderliche Systeme*, Universität Regensburg, Diplomarbeit, 2002

[Zi05] M. Zizler: *Physikalische Optimierung ausgewählter Portfolio-Probleme*, Universität Regensburg, Diplomarbeit, 2005

Index

$\pm J$-Model, 47
(N, μ)- selection, 70
(s, Q)-model, 94
$\mathcal{NP}$-complete, 21
$\mathcal{NP}$-hard, 21
(s,S)-policy, 94
1-point crossover, 70
2-point crossover, 71

spin glass phase, 40

algorithm, 18
ant colony optimisation, 76
APS, 153
artificial duck, 63
assignment problem, 33
autocorrelation, 109
average case analysis, 20

Boltzmann-distribution, 48
bond-disorder, 47
branch & bound, 11, 28
business informatics, 13

canonical ensembles, 48
capacity restriction, 125
chromosomes, 61
classical lot size model, 88, 97
competition, 44
complete enumeration, 22
complexity, 18
complexity function, 19
computational intelligence, 12
configuration, 16
configuration space, 16
cooling scheme, 56
correlation, 108
costs, 136
crossover, 64
cycles, 79

darwinism, 60
decision support systems, 14
decodation, 66
deoxyribonucleic acid, 61
detailed balance, 51
diploid, 67
discounts, 136
disorder, 44
Dixon - model, 98
DLSP, 156
dominance, 67
DRP, 153
dynamic optimisation, 25
dynamic programming, 11

Edward-Anderson-model, 47
end temperature, 57
energy landscape, 17
enterprise resource planning, 83
ergodicity, 54
evaporation, 77
evolution strategies, 73
evolutionary computing, 74
evolutionary programming, 74
execution information systems, 14
expectation value, 48
exponential cooling, 57

ferromagnet, 39

field cooling, 44
fitness function, 63
forecasting, 99
freezing temperature, 43
frustration, 41

game theory, 12
genetic operations, 68
genetic programming, 74, 75, 159
genetics, 60
Glauber function, 52
golfholes, 55
great deluge algorithm, 56
greedy, 53

Hadley-Whitin, 94
Hamiltonian, 122
heat capacity, 42, 49
Heisenberg-model, 46
heuristic, 23
heuristic lot size methods, 91
heuristics, 11, 24
hill-valley-landscape, 17
hybrid methods, 25, 67

implicit enumeration, 29
importance sampling, 50
input, 18
instance, 18
intermediary crossover, 71
intermediate storage, 84
inventory control, 84
inversion, 72
Ising-model, 45

JRP, 153

knapsack problem, 33

Landau function, 19
least-unit-cost, 91
linear cooling, 57
linear optimisation, 27
linear optimisation problem , 27
linear programming, 11
linear rank selection, 69
local search method, 78
logarithmic cooling, 57
lot size, 84
lot sizing & scheduling, 153

magnetic moment, 38
magnetism, 38
management information systems, 14
management support systems, 13
Markov process, 51
mean absolute percentage error, 106
mean squared error, 106
meiosis, 62
metaheuristics, 24
Metropolis function, 52
minimum flow problem, 32
mitosis, 62
Monte-Carlo-methods, 49
move, 16
moving average, 103
MRP, 153
multi-echelon inventory, 84, 87
multi-item inventory, 95
multimodal optimisation, 25
multiobjective optimisation, 21, 25
mutation, 63, 72

neighbourhood, 16
next-neighbour, 23
non-polynomial, 20

objective function, 15
operations research, 10
optimisation problem, 15
OR-process, 12
order, 19

paramagnet, 39
parents, 70

pareto points, 22
part-period-balancing, 92
penalties, 16
performance, 24
phase diagram, 40
phase transition, 39, 42
pheromone, 75
PMX-crossover, 71
polynomial, 20
priority numbers, 98
problem, 18
program, 18
proportional selection, 68
pseudo temperature, 56

quadratic programms, 11
queueing theory, 12
quotient programms, 11

random walk, 53
recombination, 70
reproduction, 64
RKKY, 39

search space, 17
selection methods, 68
selection pressure, 60, 73
seperable programms, 11
shortage costs, 85
Silver-Meal, 92
simple sampling, 49
simplex algorithm, 27
simulated annealing, 53
simulation models, 13
single-echelon inventory, 87
site-disorder, 47
slack variables, 27
soccer grades, 114
spin, 38
spin glass, 38
standard configuration, 124
start population, 64
start temperature, 57
state sum, 48
stationary demand, 101
statistical physics, 48
stochastic lead time, 125
stochastic programming, 11
storage cost, 84
storage input, 84
storage output, 84
supremum, 19
survival of the fittest, 60
susceptibility, 42

tabu list, 80
tabu search, 78
threshold accepting, 55
total fitness, 63
transshipment problem, 32
Trigg error, 107
TSP, 75

Value at Risk, 114
variation operators, 64
virtual costs, 15
virtual ants, 75
visibility, 75
Vogel approximation method, 23

Wagner-Whitin-model, 90
water level, 56
worst case analysis, 20

XY-model, 46

zero field cooling, 44

List of Figures

1 Possible outcomes of the annealing process 6
2 Structure of the dissertation 8

1.1 Classification of the dissertation 9
1.2 History of OR 10
1.3 Classification of management support systems [GG98] 14
1.4 Energy landscape 18
1.5 Classification of mono-objective optimisation methods [CS03] . . . 26

2.1 Schematic description of a ferromagnet (left), an anti-ferromagnet (middle) and a paramagnet (right) 39
2.2 Magnetic phase diagram of $Eu_xSr_{1-x}S$ 40
2.3 Schematic plot of the RKKY-interaction (left); Tag of four atoms (right) 41
2.4 Alternating magnetic field susceptibility of $Eu_xSr_{1-x}S$ 42
2.5 Heat capacity and susceptibility for different magnetising forces . 43
2.6 Remanent magnetisation of an AuFe-alloy (left) and a computer simulation(right) 44
2.7 Probability distribution of the energy E 51
2.8 Flow chart of simulated annealing 54

3.1 Structure of the DNA 61
3.2 Route of the duck searching for feeding places <> 63
3.3 Procedure of a genetic algorithm 66
3.4 Decodation of 1000101110001 in the sequence [-10,10] 67
3.5 Proportional selection of an individual 69
3.6 Probabilities for the linear ranking (left); distribution function (right) 70
3.7 Ants finding the shortest way after blocking 76
3.8 Different trajectories blocked or disconnected from the optimum . 81

4.1 The elementary storage transaction 84
4.2 Scheme of single-item-models 88

4.3 Classical lot size model; Left: inventory process. Right: different costs. 89
4.4 Stationary demand patterns 101
4.5 Demand patterns with growth (right) and seasonal influence (left) 103
4.6 Response of a simple exponentially weighted average forecast with $\alpha = 0.2$ [BR01] 106

5.1 Frequency distribution of the deviation between the forecast and the historic periods 113
5.2 Energy and heat capacity of a simulated forecast based on three (above) and 13 (below) historic periods with 439 items and a budget of 680 116
5.3 Comparison of the different forecast methods 117
5.4 Comparison of the different forecast methods 118
5.5 Frequency distribution before (left) and after (right) smoothing . 119

6.1 Sales Figures of two different items i and j over a time of 34 periods 126
6.2 Energy and heat capacity for 1 and 50 items over 34 periods . . . 127
6.3 Energy of the different Sub-Hamiltonians 129
6.4 Energy and heat capacity of the weighted simulation for 50 items and 34 periods 130
6.5 Energy of the different Sub-Hamiltonians with TA 131
6.6 Energy and heat capacity of a (t,S)-policy for 1 item and 34 periods (above); Energy and heat capacity of a (t,S)-policy for 50 items and 17 periods (below) 132
6.7 Energy and heat capacity of a (s,Q)-policy for 50 items and 17 periods 135
6.8 Comparison of different policies concerning energy and heat capacity 140
6.9 Components of energy and heat capacity concerning an (s,Q)-policy 141
6.10 Equivalences between spin glass and inventory problem 146

7.1 Comparison of complete enumeration with physical and genetic optimisation of a (s,S)-policy 150
7.2 Return of different items 151
7.3 Comparison of physical and genetic optimisation of a (s,Q)-policy with random sales (left) and "soccer" sales (right) 152

8.1 Scheme of proceeding 163

List of Tables

1.1 Definition 1 19
1.2 Definition 2 20
1.3 Pivot Format 28
1.4 Simplex - Algorithm 29
1.5 Branch & Bound 30
1.6 Definition of a TSP 31

2.1 Analogy between an optimisation problem and a physical system . 37
2.2 Parameters of the spin glass Hamiltonian 45
2.3 Simulation of the ±J-Model 52

3.1 Genetic algorithm 65
3.2 Basic ant colony algorithm 78
3.3 Pseudocode of local search methods 79
3.4 Pseudocode of tabu search 80

4.1 Deterministic decision problem 90
4.2 Parameters of the Newsboy problem 93
4.3 Variables of an Multi-Item-Inventory 97
4.4 Types of demand forecast based on underlying time unit 100

6.1 Variables of the Inventory-Hamiltonian 124
6.2 Comparison of different policies 133
6.3 Application of sQ- and sS-policy to future periods 135
6.4 Comparison of optimal (s,Q)-policies (with different parameters) to future periods 142
6.5 Comparison of different policies to future periods and optimisation results 143

Acknowledgements

The way to a doctor's degree is always long and busy. Without the help of many others such an enduring project could never be accomplished. First of all I have to express my thanks to Prof. Dr. Ingo Morgenstern. Four years ago I got to know him in a lecture as a friendly, uncomplicated and slightly different professor. Because of his interdisciplinary field of research he is often evaluated as an underdog in the faculty. But just because of this he was the best one to meet for me. In my eyes interdisciplinary research will be the future and Prof. Morgenstern is one of its precursors. He made it possible for me to work on a interdisciplinary and widely unexplored topic; therefore the progress of my work was good and the results are plenty. Beside the scientific teamwork I have to thank him for my time abroad, which really gave me new input and motivation. And especially I enjoined the little talks about non-physical topics like soccer; this made the time of the last years very pleasant. This leads me to my second advisor Prof. Dr. Rainer Gömmel, who is also an overall connoisseur of soccer (therefore I've learnt a lot about soccer !). Without him my dissertation surely would not be possible in the way as it is. Interdisciplinarity is often requested by ministry, but most professors just want to do their own stuff and try nothing new. Fortunately Prof. Dr. Gömmel is different and my project could flower out.

There are many other people, which made a precious contribution to the success of my work. For example my previous colleague Wolfgang Feil, who taught me the basics of optimisation. Then my current room mates Rainer Schuster and Ulrich Meier, which gave me many moments to laugh and joke. And not to forget the secretary Lizy Lazar, who helped me in all sorts of formal stuff. Besides the members of my own workgroup, some others of the adjacent workgroup have to be named, especially Markus, Tom, Flo and Emiliano, which helped me to disperse time, when there was no need or notion for research. In particular those people are necessary, which keep you moving when there seems no force to go any further; therefore I want to thank all friends and my family. Under those I have to highlight Stefan Rinner, who gave me much intelligent advice and shared a lot of sympathy with my work.