11

Real-World Applications of Multiobjective Optimization

Theodor Stewart[1], Oliver Bandte[2], Heinrich Braun[3], Nirupam Chakraborti[4], Matthias Ehrgott[5], Mathias Göbelt[3], Yaochu Jin[6], Hirotaka Nakayama[7], Silvia Poles[8], and Danilo Di Stefano[8]

[1] University of Cape Town, Rondebosch 7701, South Africa
   theodor.stewart@uct.ac.za
[2] Icosystem Corporation, Cambridge, MA 02138 oliver@icosystem.com
[3] SAP AG, Walldorf, Germany heinrich.braun@sap.com
[4] Indian Institute of Technology, Kharagpur 721 302, India
   nchakrab@iitkgp.ac.in
[5] The University of Auckland, Auckland 1142, New Zealand
   m.ehrgott@auckland.ac.nz
[6] Honda Research Institute Europe, 63073 Offenbach, Germany
   Yaochu.Jin@honda-ri.de
[7] Konan University, Higashinada, Kobe 658-8501, Japan nakayama@konan-u.ac.jp
[8] Esteco Research Labs, 35129 Padova, Italy silvia.poles@esteco.com

Abstract. This chapter presents a number of illustrative case studies of a wide range of applications of multiobjective optimization methods, in areas ranging from engineering design to medical treatments. The methods used include both conventional mathematical programming and evolutionary optimization, and in one case an integration of the two approaches. Although not a comprehensive review, the case studies provide evidence of the extent of the potential for using classical and modern multiobjective optimization in practice, and opens many opportunities for further research.

11.1 Introduction

The intention with this chapter is to provide illustrations of real applications of multiobjective optimization, covering both conventional mathematical programming approaches and evolutionary multiobjective optimization. These illustrations do cover a broad range of application, but do not attempt to provide a comprehensive review of applications.

In examining the case studies presented here, it may be seen that the applications may be distinguished along two primary dimensions, namely:

The number of objectives which may be:
- Few, i.e. 2 or 3 (impacts of which can be visualized graphically);
- Moderate, perhaps ranging from 4 to around 20;
- Large, up to hundreds of objectives

The level of interaction with decision makers, i.e. the involvement of policy makers, stakeholders or advisers outside of the technical team. The level of such interaction may be:
- Low, such as in many engineering design problems (but for an exception, see Section 11.7) where the analyst is part of the engineering team concerned with identifying a few potentially good designs;
- Moderate, such as in operational management or interactive design problems where solutions may need to be modified in the light of professional or technical experience from other areas of expertise;
- Intensive, such as in public sector planning or strategic management problems, where acceptable alternatives are constructed by interaction between decision makers and other stakeholders, facilitated by the analyst.

Not all combinations of number of objectives and level of interaction may necessarily occur. For example, the public sector planning or strategic management problems which require intensive interactions, also tend often to be associated with larger numbers of objectives. In the case studies reported in this chapter, we have attempted to provide summaries of a number of real case studies in which the authors have been involved, and which do between them illustrate all three levels for each dimension identified above. Table 11.1 summarizes the placement of each of the cases along the above two dimensions.

These studies exemplify the wide range of problems to which multiobjective optimization methods can and have been applied. Half of the case studies deal with engineering design problems, which is clearly an important area of application, but even within this category there is a wide diversity. For

Table 11.1. Classification of case studies

Number of Objectives	Level of Interaction	Case study
Few	Low	Aerodynamic design (Section 11.2)
		Industrial neural network design (Section 11.3)
		Molecular structures for drugs (Section 11.4)
Few	Moderate	Medical decision making (Section 11.5)
		Supply chain management (Section 11.6)
Moderate	Moderate	Interactive aircraft design (Section 11.7)
Moderate	Intensive	Land use planning (Section 11.8)
Large	Low	Lens and bridge designs (Section 11.9)

example, we have two examples from aircraft design, but one (Section 11.2) focuses on the trade-off between robustness and cost in aircraft design, while the other (Section 11.7) deals with the need to provide a broad holistic interactive decision support to aircraft designers. Although both applications relate to aircraft design, the issues raised are substantially different so that different sections in this chapter are devoted to each of them.

Other applications range over operational management of supply chains, effective treatment of cancers and conflicts between environmental, social and economic factors in regional planning.

A perhaps less usual application is that described in Section 11.4. Here the multiobjective optimization methods are applied not directly to design, operational or strategic decisions, but to the development of understanding of molecular processes in synthesizing drugs.

11.2 Aerodynamic Design Optimization

11.2.1 Problem Description

Although the number of objectives are typically low (two or three) if the geometrical constraints are not counted, aerodynamic design optimization is a challenging engineering task for a number of reasons. Firstly, aerodynamic optimization often needs to deal with a large number of design parameters. Secondly, no analytical function is available for evaluating the performance of a given design, and as a result many gradient-based optimization techniques are inapplicable. Thirdly, to evaluate the quality of designs, either computationally expensive computational fluid dynamics (CFD) simulations have to be performed or costly experiments have to be conducted. Finally, aerodynamic optimization involves multiple disciplines and more than one objective must be considered.

In recent years, evolutionary algorithms have successfully been applied to single and multiobjective aerodynamic optimization (Obayashi *et al.*, 2000; Olhofer *et al.*, 2000; Hasenjäger *et al.*, 2005). Despite the success that has been achieved in evolutionary aerodynamic optimization, several issues must be carefully addressed.

11.2.2 Methodology

Geometric Representation

Finding a proper representation scheme is the first and most important step toward successful optimization of aerodynamic structures. A few general criteria can be mentioned for choosing an appropriate geometric representation. Firstly, the representation should be sufficiently flexible to describe highly complex structures. An overly constrained representation will produce only

suboptimal results. Secondly, the representation should be efficient, which means that the flexibility of representation can be achieved with a minimum number of free parameters. Inefficient representations may result in an unnecessarily large search space, which reduces the search efficiency of evolutionary algorithms. Finally, the representation should makes it possible to perform a local search. This requirement is important for refining the performance and for reducing search space.

Several methods are available for the geometric representation, such as B-Splines, Bezier curves, and T-Splines. Furthermore, constrained deformation instead of global deformation techniques can be used. An example of constrained deformation is free form deformation or simplified constrained deformation.

Nevertheless, it can happen that no single representation is able to satisfy all the above-mentioned properties. To solve this problem, adaptive representation techniques can be used. The basic idea of adaptive representation is to start the optimization with a relatively compact representation, so that the global search can be conducted first. After that, the number of search parameters can be increased, e.g., by inserting new control points in a B-Spline based representation. Encouraging results have been reported where an adaptive representation for evolutionary optimization of turbine blades has been adopted (Olhofer *et al.*, 2001).

However, adaptive representation is not as straightforward as it appears. On the one hand, it is not trivial to establish when a new point should be inserted, or removed. On the other hand, the insertion or removal of a search point should be neutral to the fitness value. Moreover, an adaptation in representation may degrade the search performance of evolutionary algorithms, for example, for evolutionary strategies with a small population size (Jin *et al.*, 2005).

In multiobjective optimization, even more complex situations can occur. For example, it has been found that in micro heat-exchanger optimization more than one representation is needed to achieve the whole Pareto front (Okabe *et al.*, 2003).

Reduction of Computational Cost

Evolutionary algorithms (EAs) acquire strong search power at the cost of search efficiency. In contrast to gradient-based search methods, EAs often need a large number of quality evaluations to achieve acceptable solutions. This poses serious problems in aerodynamic optimization where each quality evaluation is costly. For example, a full three-dimensional CFD simulation may takes several hours on a high-end computer. To reduce the computational time for evolutionary optimization aerodynamic structures, the following approaches are adopted. Firstly, efficient and scalable evolutionary algorithms need to be developed. An efficient and scalable EA should be able to converge

to an acceptable non-dominated front within a small number of fitness evaluations, and should be insensitive to the increase in search dimension. Secondly, both the CFD simulations and the fitness evaluations should be parallelized. A further step is to take advantage the grid computing techniques that enable us to use available computational resources as efficiently as possible. In addition, computational efficiency can further be improved if EAs are adapted to the parallel or grid based computing hardware architecture. Finally, computationally expensive full simulations can be replaced partly by computationally more efficient reduced simulations, or surrogates. In recent years, metamodeling techniques have been extensively investigated to reduce the computational cost in evolutionary optimization of expensive problems, both for single objective (Jin *et al.*, 2002) and multiobjective optimization (Emmerich *et al.*, 2006). Refer to Chapter 10 on metamodeling for a more detailed discussion of their use in multiobjective optimization.

Metamodels can introduce errors in quality evaluation, which may lead to the convergence to a false minimum (Jin *et al.*, 2000). Therefore, a major concern in using surrogates in evolutionary optimization is to reduce computational cost as much as possible without misleading the evolutionary algorithm. To this end, the meta-model should be properly interleaved with the original fitness function, which is known as evolution control or model management (Jin *et al.*, 2002). A meta-model can be employed in different operations of EAs, such as population initialization, crossover or mutation, evolutionary fitness evaluation, or local search combined with evolutionary search. Different model management techniques have been suggested. In the individual-based techniques, all individuals in the current generation are evaluated with the metamodel. Then, the most promising solutions according to the metamodel are re-evaluated using the original fitness function. In a generation-based evolution control framework, the meta-model is used for fitness evaluation in some of the generations, and the frequency at which the meta-model is used can be adjusted according to the fidelity of the model (Jin *et al.*, 2002). If a metamodel is employed in local search, the trust-region framework (Alexandrov *et al.*, 1998) can be adopted. A comprehensive survey of techniques for using meta-models (surrogates) in evolutionary optimization can be found in Jin (2005).

Robustness Considerations

In aerodynamic optimization, uncertainties in the environment must be taken into account. For example, the Mach number may deviate from the normal condition during the flight. In this case, a robust optimal solution is very much desired. By robustness, it is meant in general that the performance of an optimal solution should be insensitive to small perturbations of the design variables or environmental parameters. In multiobjective optimization, the robustness of a solution can be an important factor for a user in choosing the final solution. Robust solutions can be achieved in evolutionary optimization

by a number of means. One simple approach is to add perturbations to the design variables or environmental parameters before the fitness is evaluated, which is known as implicit averaging, see e.g., Tsutsui and Ghosh (1997). An alternative to implicit averaging is explicit averaging, which means that the fitness value of a given design is averaged over a number of designs generated by adding random perturbations to the original design. One drawback of the explicit averaging method is the number of additional quality evaluations needed, which may make the approach impractical. Partly to solve this problem, metamodeling techniques have been considered (Ong *et al.*, 2006; Paenke *et al.*, 2006). A slightly different approach is to find the solution with the maximal allowed deviation given the allowed performance deterioration (Lim *et al.*, 2007). One potential advantage of this methods is that no assumptions need to be made concerning the noise distribution (as needed in the averaging based approaches).

Search for robust solutions can also be treated as a multiobjective task, i.e., to maximize the performance and the robustness simultaneously. These two tasks are very likely conflicting, and therefore, Pareto-based multiobjective methods can be employed to find a number of trade-off solutions. Refer to Jin and Branke (2005) for a more detailed discussion on evolutionary search for robust solutions.

11.2.3 An Example

We present here an example of evolutionary multiobjective optimization of a three-dimensional (3D) turbine stator blade used in gas turbines. Two objectives are taken into account in the optimization. The first objective is the average pressure loss, which indicates the energy efficiency of the blade. The second objective, as suggested in Hasenjäger *et al.* (2005), is the variation of the pitch-wise static outlet pressure.

The 3D geometry of the blade is represented by two sections of closed cubic B-splines, namely, a hub section and a tip section, each consisting of 25 control points, as illustrated in Fig. 11.1. In the representation, the first three and the last three control points of the closed B-splines are overlapping, resulting in 22 control points. In addition, since the hub and tip sections are supposed to lie on a cylindrical surface, the z-coordinate of the control points is fixed and not optimized during the evolution. As a result, 88 design parameters in total (x and y coordinates of 22 control points) are to be optimized by the evolutionary algorithm.

To evaluate the performance of a given design, 3D CFD simulations have to be performed. In our work, a 3D Navier-Stokes flow solver, HSTAR3D (Arima *et al.*, 1999) is employed, which usually takes from two to four hours on an AMD Opteron 2 GHz double processor depending on the convergence speed of the fluid dynamics. To reduce computation time, a two-level parallel computing architecture has been adopted. At the first level, fitness evaluations for each individual in the population are parallelized using the master-slave
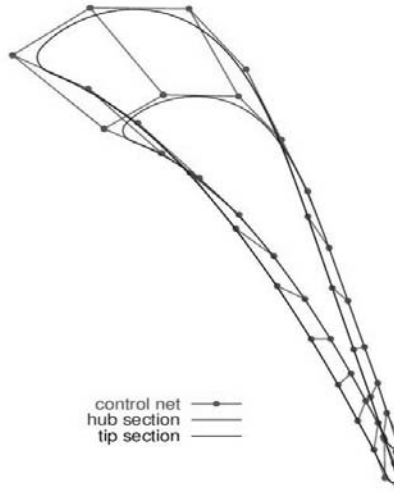
Fig. 11.1. Three-dimensional representation of a blade using B-splines.

model. At the second level, each CFD simulation is again parallelized on four computers using the node-only model. Consequently, if the population size is P, the needed number of computing nodes will be $4P+1$.

An efficient model-based evolutionary multiobjective optimization algorithm, the regularity modeling multiobjective estimation of distribution algorithm (RM-MEDA) (Zhang *et al.*, 2008), has been employed for the optimization of the 3D turbine blade. RM-MEDA is in principle a variant of estimation of distribution algorithms (EDA) (Larranaga and Lozano, 2001). Instead of using Gaussian models, a first-order principle curve has been used to model the regularly distributed Pareto-optimal solutions complemented by a Gaussian model. As demonstrated in Jin *et al.* (2008), by modeling the regularity in the distribution of Pareto-optimal solutions, the scalability of the EDA can be greatly improved. Furthermore, unlike most EDAs, which require a large population size, RM-MEDA performs well even with a small population size. In this example, a population size of 20 has been used.

The optimization results from two independent runs are plotted in Fig. 11.2, in each of which the population has been evolved for 100 generations. Note, however, that the population was initialized not randomly, but with solutions from previous optimization runs using weighted aggregation approaches (Hasenjäger *et al.*, 2005). Compared to the results reported in Hasenjäger *et al.* (2005), we see that the non-dominated solutions obtained by the RM-MEDA are better in terms of both coverage and accuracy.
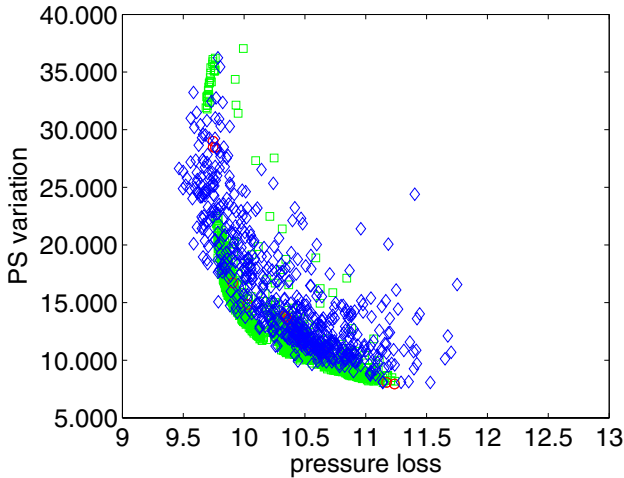
Fig. 11.2. Solutions obtained from two independent evolutionary runs. The diamonds denote the solutions from the first run, and the squares from the second run.

11.3 Design of a Neural Network for an Industrial Blast Furnace

11.3.1 Problem Description

Intuitively one can conceive of a neural network that is simultaneously attempting to satisfy two different requirements: it should be able to reproduce the data in an accurate manner and simultaneously, it should engage a small number of neural connections, primarily to avoid over-training of the data set. These two objectives in many cases could be conflicting with each other. As expected, as the number of nodes becomes smaller, the training error tends to shoot up, and the converse usually remains true as well. In term of these two objectives, one can thus think of working out a Pareto tradeoff, where each solution in the Pareto frontier denotes a neural network of a unique architecture with a unique set of weights. A procedure for evolving such frontiers through a Predator-Prey type multi-objective Genetic Algorithm (Li, 2003) has been demonstrated in a recent article (Pettersson *et al.*, 2007a) and was elaborately tested on the highly nonlinear data from an industrial iron making blast furnace shown schematically in Figure 11.3.

The aim of the study reported in Pettersson *et al.* (2007a) was to evolve a neural network that would optimally predict the carbon, sulfur and silicon contents of the hot metal produced in the blast furnace as a function of a number of process parameters. What was attempted there was simultaneously to minimize (i) the training error of the network (E) and (ii) the required

number of active connections in the lower part of it (N). The idea was to tinker with the architecture of the lower part of the network, and to treat their corresponding weights as variables influencing the objective functions, as further elaborated in Figure 11.4. The trade off situation between E and N is expected to be represented as a Pareto-frontier.

11.3.2 Methodology

The evolutionary process: Here the crossing over was done between two entities both of which are essentially self-sustaining neural networks. This process is further elaborated in Figure 11.5. A self-adaptive real-coded mutation was performed on the weights, which draws its inspirations from Differential Evolution (Price *et al.*, 2005).

The multi-objective algorithm used in this study utilized a Moore neighbourhood inhabited by two distinct species: the predators and the preys. The preys are a family of sparse neural networks, initiated randomly as a population, and they evolved in the usual genetic way. The members of the prey population differed from each other both by the topology of the lower part connections and the corresponding weight values. The predators in this algorithm are a family of externally induced entities, which do not evolve, and the major purpose of their presence is to prune the prey populations based upon the fitness values. A two dimensional lattice was constructed as a computational space and both the predators and the prey were randomly introduced there, where each of them would have its own neighbourhood. The basic idea propagated in this algorithm inherits some of the concepts of cellular automata in Moore's neighbourhood. However, unlike cellular automata, here the lattice here does not denote the discretized physical space; it is just a mathematical construction that facilitates a smooth implementation of this algorithm. Further details are available in the original work (Pettersson *et al.*, 2007a).

The method seems to have worked better when the initial population is deliberately generated in the vicinity of the estimated nadir region. The progress of the rank-one members is captured in Figure 11.6 and a computed Pareto frontier is shown in Figure 11.7. Each discrete point in the frontier denotes a neural net with a different ability of prediction than the others. Some typical examples are shown in Figure 11.8. As the ultimate choice between them remains the task of the decision maker, the conservative middle ground 'B' shown in Figure 11.7 should be adequate for most applications.

This novel method of multi-objective analysis is not just to benefit the steel industries: basically it is robust enough to handle noisy data irrespective of their sources. Very recently this methodology has been augmented further through the use of Kalman filters (Saxén *et al.*, 2007), and it has also been effectively utilized for identifying the most important in-signal in a very large network (Pettersson *et al.*, 2007b), rendering it of further interest to the soft computing researchers at large.
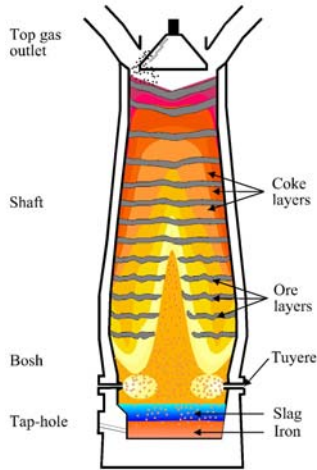
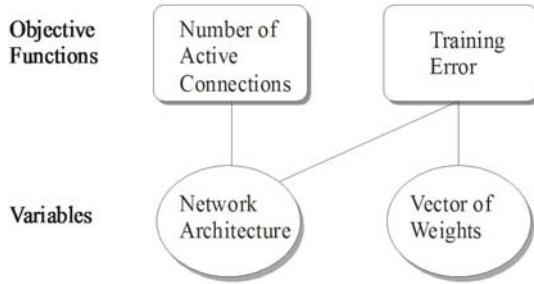**Fig. 11.3.** Schematic diagram of an iron making blast furnace.

Fig. 11.4. Multi-objective formulation of the neural network.
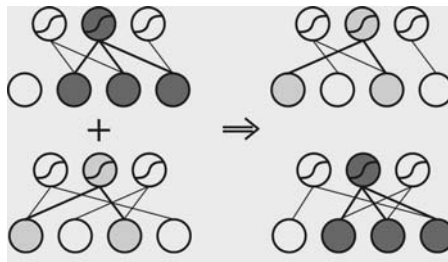

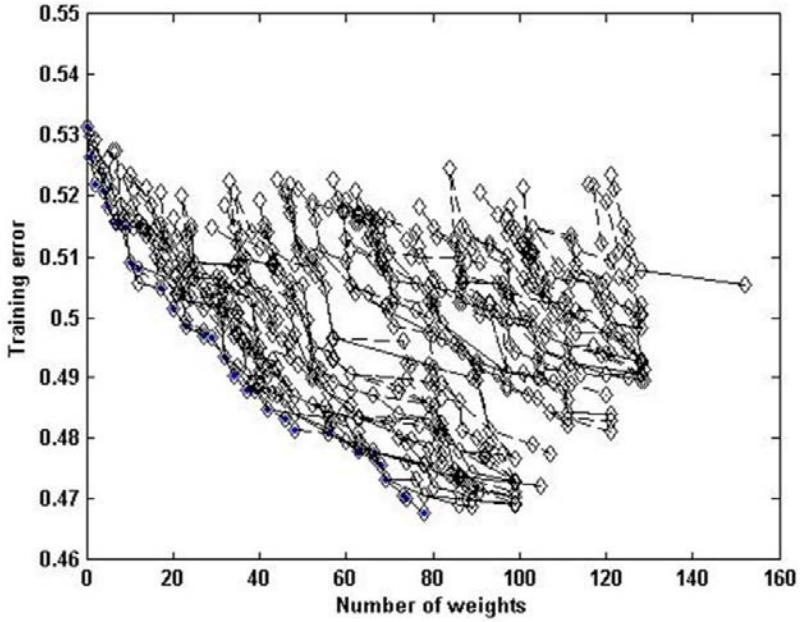**Fig. 11.5.** The crossover scheme. The shaded regions are participating in the crossover process.

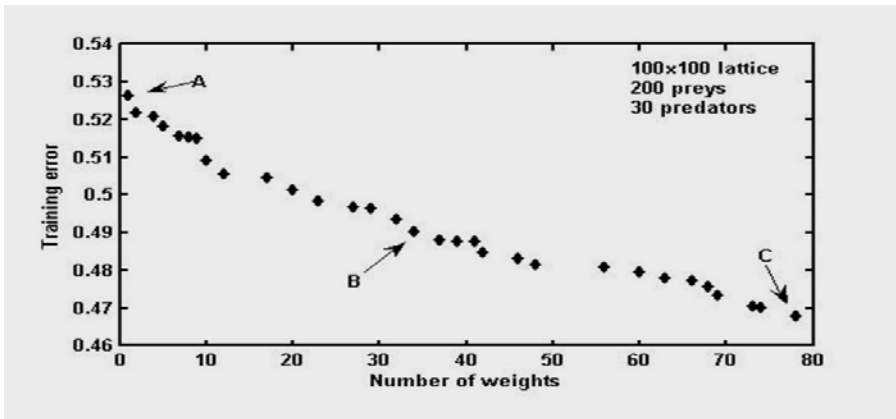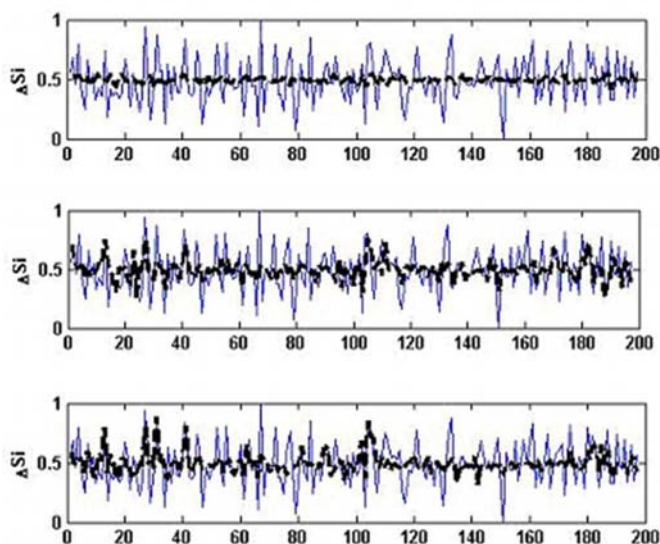**Fig. 11.6.** Movement of rank 1 population in different generations.

Fig. 11.7. A typical Pareto frontier presented in (Pettersson *et al.*, 2007a).

Fig. 11.8. Data prediction through three networks A (top), B (middle) and C (bottom). The lighter lines denote actual observations for a period of 200 days and the darker lines are the predicted values provided in (Pettersson *et al.*, 2007a).

11.4 Molecular Docking

11.4.1 Problem Description

The docking of a highly flexible small molecule (the ligand) to the active site of a highly flexible macromolecule (the receptor) is described in this Section. See Morris *et al.* (1998); MacKerell Jr. (2004) for a more detailed discussion of the problem. The ability to predict the final docked structure of the intermolecular complex has a great importance for the development of new drugs as docking modifies the biological and chemical behavior of the receptor. Most of the current docking methods account only for the ligand flexibility and consider the receptor as a rigid body because the inclusion of receptor flexibility could involve thousands of degrees of freedom. Current research in this field is faced with this problem. The application described here focuses on a different aspect of the docking procedure: the optimization methodology applied to find the best docked structure. The application of a multi-objective approach to the docking problem based on the Pareto optimization of different features of a docked structure is proposed. It is shown that this approach allows for the identification of the dominating interactions that drives the global process.

A drug performs its activity by binding itself to the receptor molecule, usually a protein. In their bounded structure, the molecules gain complementary chemical and geometrical properties that are essential for the therapeutic

function of the drug. The computational process that searches for a ligand that best fits the binding site of the receptor from a geometrical and chemical point of view is known as molecular docking.

A molecule is represented by its atoms and the bonds connecting them. Atoms are described mainly by their Van der Waals radius that roughly defines their volume; bonds are described by their lengths (the distance between atoms), by the angle between two consecutive bonds and by their conformational state (the dihedral angle between three consecutive bonds). Molecules are not static systems. At room temperature they perform a variety of motions each one having a characteristic time scale. Since the time scales of stretching (changes in bond lengths) and bending (changes in bond angles) have greater time scales than conformational motions (changes in dihedral angles), bond lengths and bond angles can be considered fixed. Thus, from the docking point of view, only conformational degrees of freedom are important.

Typically, ligands have from 3 to 15 conformational degrees of freedom; their values define the conformational state of the ligand. Receptors have typically from 1000 to 3000 conformational degrees of freedom, so the dimension of the complete search space for best docked conformation becomes computationally unaffordable even for routine cases. The most widely used simplification is to consider only the ligand flexibility, so reducing the complexity of the search space.

The different possible ligand conformations are ranked according to their fitness with the receptor. What this fitness stands for is one of the key aspects of molecular docking and differentiates various docking methodologies. Most of the docking fitness functions are based on the calculation of the total energy of the docked structure. Energy based fitness functions are built starting from force fields which represent a functional form of the potential energy of a molecule. They are composed of a combination of different terms that can be classified in bonded terms (regarding bond energies, bond angles, bond conformations) and non-bonded terms (Van der Waals and electrostatic). This energy can be calculated in various ways, ranging from quantum mechanics to empirical methods. Obviously, a more "exact" fitness function as derived from quantum mechanical simulations strongly impacts on the computational complexity and is applicable only for small systems on massive parallel computers; the opportunity to use "rough" empirical models creates the possibility of treating more realistic cases.

In summary, a docking procedure is composed by two main elements: a fitness function to score different conformations for the molecular complex and a search procedure to explore the space of possible conformations. In current docking approaches, the bonded and non-bonded terms both contribute to the fitness function and the optimization has a single objective equal to their weighted sum. The weights are determined by statistical analysis of experimental data. The proposed multi-objective optimization approach incorporates two conflicting objectives, i.e. the concurrent minimization of internal

and intermolecular energy terms, derived from a suitable scoring function, each one corresponding to an objective for the optimization algorithm.

11.4.2 Methodology

MOGA-II

MOGA-II is an improved version of the MOGA (Multi-Objective Genetic Algorithm) of Poloni (Poloni and Pediroda, 1997). It uses smart multi-search elitism for robustness and directional crossover for fast convergence. The efficiency of MOGA-II is controlled by its operators (classical crossover, directional crossover, mutation and selection) and by the use of elitism. The internal encoding of MOGA-II is implemented as in classical genetic algorithms. Elitism plays a crucial role in multi-objective optimization because it helps preserving the individuals that are closest to the Pareto front and the ones that have the best dispersion. MOGA-II uses four different operators for reproduction: one point crossover, directional crossover, mutation and selection. At each step of the reproduction process, one of the four operators is chosen with regard to the predefined operator probabilities.

A strong characteristic of this algorithm is the directional crossover that is slightly different from other crossover operators and assumes that a direction of improvement can be detected comparing the fitness of individuals. A novel operator called evolutionary direction crossover is introduced and it is shown that even in the case of a complex multi-modal function this operator outperforms classical crossover. The direction of improvement is evaluated by comparing the fitness of the individual Ind_i from generation t with the fitness of its parents belonging to generation $t-1$. The new individual is then created by moving in a randomly weighted direction that lies within the ones individuated by the given individual and his parents.

Multi-objective Ligand-Receptor Docking

In this example, the MOGA-II implementation in modeFRONTIER® is used to optimize the docking towards each of the different contributions of the docking program Autodock v. 3.05 (`http://autodock.scripps.edu`) scoring function:

$$\Delta G = C_{\text{CVDW}}\Delta G_{\text{GVDW}} + C_{\text{hbond}}\Delta G_{\text{hbond}} + C_{\text{elec}}\Delta G_{\text{elec}} \\ + C_{\text{tor}}N_{\text{tor}} + C_{\text{sol}}\Delta G_{\text{sol}} \quad (11.1)$$

that tries to estimate the change in Gibbs free energy G involved in passing from the system (receptor + ligand) to the docked system (receptor-ligand). The coefficients C are parametrized from experimental data and set to proper values. *VDW* stands for Van der Waals contribution, *hbond* for hydrogen bonds contribution, *elec* for electrostatic contributions, *tor* for the entropy change if N_{tor} rotatable bonds are connected with heavy atoms, and *sol* for the change in solvation energy.

11.4.3 Results

A "bound docking" experiment was performed: on the basis of the x-ray structure of the complex, the receptor coordinates were separated from those of the ligand, and then an attempt made to reconstruct the original x-ray structure by docking the ligand to the receptor. Starting from the scoring function of equation (11.1), Autodock gives the values for the internal energy of the ligand and for the intermolecular ligand-receptor interaction energy. These two outputs were assigned as the objective of the optimization.

The tests were conducted on PDB code 1KV3 chain A co-crystallized with GDP (`http://www.rcsb.org/pdb`). The resulting Pareto front is reported in Figure 11.9 (in which the units for the axes are Kcal/mol).
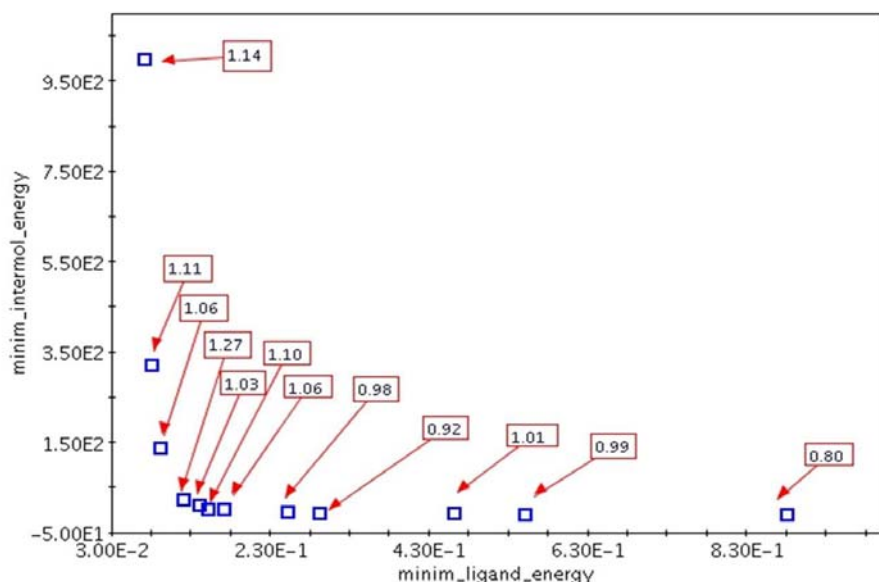
Fig. 11.9. Pareto frontier for the molecular docking problem. Energies are in Kcal/mol. Boxed values represent RMSD in Angstrom between the candidate solution and the original x-ray structure.

The squared values represent the root mean squared deviation (RMSD) in $\mathring{A}$ (angstrom) between the candidate solution and the original x-ray structure. Typically, RMSD values less than 1.5 $\mathring{A}$ are considered as good solutions. It is possible to note that in this case the docking process is mainly driven by the intermolecular energy. This information could be useful for a deeper understanding of the effective relative influence of the contributions of the scoring function to this particular docking process. From a practical point of view, it could also be useful for the design of a tailored scoring function for the

docking of similar drug candidates. Also note the presence of a "knee" point (RMSD=1.27 $\mathring{A}$). This is a particularly interesting solution of the docking problem in which a small improvement in the minimization of the ligand energy leads to a large deterioration of the intermolecular energy.

11.5 Radiotherapy Treatment Planning

11.5.1 Problem Description

Cancer is one of the most significant health problems worldwide. In industrialised countries it is the second most common cause of death and more than 5 in every 1,000 people are diagnosed with some type of cancer every year. The main treatment form besides surgery and chemotherapy is radiation therapy. It is estimated that 50% of all patients diagnosed with cancer would currently benefit from radiotherapy.

Ionising radiation is used to damage the DNA and interfere with division and growth of cancer cells. Radiation therapy exploits the fact that cancerous cells are more susceptible to radiation than healthy cells. The goal of radiotherapy treatment planning is therefore to ensure that enough radiation is delivered to the targeted region to kill the cancerous cells while surrounding anatomical structures are spared.

In the past it was possible for a physician manually to design a treatment that took full advantage of the available technology. Modern procedures use a technique called Intensity Modulated Radiotherapy (IMRT). This technique uses a multileaf collimator to shape the beam and control, or modulate, the intensity that is delivered along a fixed beam direction. IMRT allows patients to receive complicated treatments and the number of options that are available in IMRT places the optimal planning of a treatment outside the realm of human awareness. Because of this complexity of the planning process, treatment planning is segmented into a three-phase process that first selects beam directions then decides an intensity map (exposure times, fluence) for the directions selected in phase one, and finally chooses a delivery sequence that efficiently administers the treatment. Computer assisted optimisation methods are needed in each phase and since the end of the 20th century these problems have attracted the interest of the Operations Research community. Surveys on optimisation methods for the three phases can be found in Ehrgott *et al.* (2008a); Shao (2005); Ehrgott *et al.* (2008b), respectively. In the following we will concentrate on the intensity optimisation problem and assume that beam directions are given.

In 2000, we started a collaboration with the Physics Section of the Oncology Department at Auckland City Hospital to work on treatment planning problems. Treatment planners spend between 30 minutes and several hours on one single case. This is because the available planning system (like almost all commercially available ones worldwide) requires a trial and error approach.

Apart from desired dose levels in the tumour and surrounding critical structures, so called "importance factors" for these entities need to be specified as input. The software then employs heuristics to find a good treatment plan, which is presented to the planner. If it is unsatisfactory the importance factors have to be changed and the process will be repeated. Treatment planners are aware of the inefficiency of this approach. So the goal was to investigate the possibility of a planning system that would calculate several plans right away and provides decision support for choosing an appropriate one.

11.5.2 Methodology

Mathematical models for the intensity optimisation problem are based on the discretisation of the body and the beams. The body is divided into volume elements (voxels) represented by dose points. Voxels are cubic and their edge length is defined by the slice thickness and resolution of the patient's CAT images and is in the range of a few mm. at most. Deposited dose is calculated for one dose point in every voxel and assumed to be the same throughout the voxel. A beam is discretised into beam elements (bixels). Their size is defined by the number of leafs of the collimator and the number of stops for each leaf. The number of voxels may be tens or hundreds of thousands and the number of bixels can be up to 1,000 per beam. The relationship between intensity and dose is linear, i.e., $d = Ax$ where x is a vector of bixel intensities. The entries a_{ij} of A represent the rate at which dose is deposited in voxel i by bixel j. Finally, d is a dose vector that represents the discretised dose distribution in the patient. The computation of the values a_{ij} is referred to as dose calculation.

   While most optimisation models in the medical physics literature have a single objective, they do try to accommodate the conflicting goals of destroying tumour cells and sparing healthy tissue. Almost all can be interpreted as weighted sum scalarisations of multi-objective programming models, where the weights are the importance factors mentioned above. Almost all of these multi-objective models are convex problems, so that their efficient sets can be mapped to one another. We decided to use a multi-objective version of the model of Holder (2003), which has some nice mathematical properties. Here A is decomposed by rows into A_T, A_C, and A_N depending on whether a voxel belongs to the tumour, critical structures, or normal tissue. Accordingly, TUB and TLB are vectors of upper and lower bounds on the dose delivered to the tumour voxels; CUB is a vector of upper bounds for the critical structure voxels; and NUB a vector of upper bounds for the remaining normal tissue voxels. The objectives of the model are to minimise the violation of any of the lower and upper bounds and can be stated as shown in (11.2). αUB, βUB, and γUB are parameters to restrict the deviations to clinically relevant values.

$$\min\{(\alpha, \beta, \gamma) : TLB - \alpha e \leqq A_T x \leqq TUB, A_C x \leqq CUB + \beta e,$$
$$A_N x \leqq NUB + \gamma e, 0 \leqq \alpha \leqq \alpha UB,$$
$$\min_i CUB_i \leqq \beta \leqq \beta UB, 0 \leqq \gamma \leqq \gamma UB, 0 \leqq x\}, \qquad (11.2)$$

where e denotes a vector of ones of appropriate dimension.

We will denote the feasible set of (11.2) by X and its image in the objective space by Y. In (11.2) we have a multi-objective linear programme with three objectives, a large number of variables (order of thousands), and a very large number of constraints (order of hundred thousands). Under these circumstances we never did try to solve the problem with simplex methods, as it is known that the number of efficient basic feasible solutions can be very large, even for moderately sized problems. Moreover, treatment planners will never use the intensity maps to decide on a treatment, but always look at the dose distribution. The obvious choice was to try and solve the problem in the three-dimensional objective space.

To that end Benson's outer approximation algorithm (Benson, 1998) was implemented. With this method 2D planning problems (i.e. on a single CAT slice) could be solved, but the experiments indicated that 3D problems would require prohibitive computation times. It was therefore necessary to consider approximate solution of the problem. Discussions with physicists on whether that would be acceptable from a radiotherapy point of view provided valuable insights. We discovered that dose calculation is imprecise because the mathematical models to compute the entries of A are inexact since they cannot exactly capture the specific tissue composition in individual patients. The medical physicists assured us that it is acceptable to work with precision of about 0.1 Gy (Gy, for Gray, is the physical unit for radiation dose).

This allowed us to consider ε-efficient solutions of (11.2). It was possible to adapt Benson's algorithm in such a way that it does guarantee the construction of an ε-nondominated set, the modified algorithm is described in Shao and Ehrgott (2008). Solving the problems approximately reduced the computation times dramatically. Figure 11.10 (a) and (b) show the ε-nondominated set of a 2D problem with 986 voxels and 1140 bixels. For $\varepsilon = 0.1$ the problem had 152 nondominated extreme points and was solved in 20 minutes. For $\varepsilon = 0.005$ it took 9 hours to compute 1,989 nondominated extreme points.

11.5.3 Interactive Scheme

From the planners' point of view the whole set of nondominated points is not very useful, since it is infinite. Also, for the same reason of imprecision in dose calculation mentioned above, planners would not distinguish between plans if they differ only by very small amounts. It is necessary to select a finite set of nondominated points (efficient solutions). The nondominated extreme points and associated basic solutions have only mathematical relevance, but no clini-
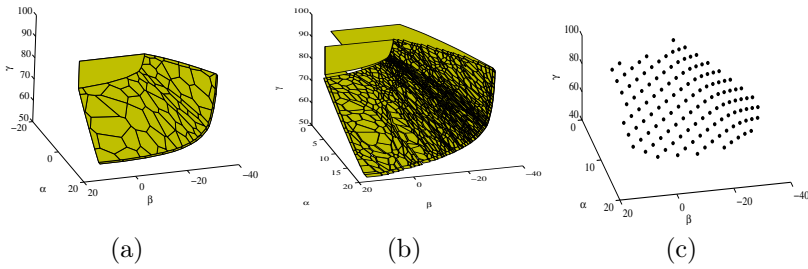
Fig. 11.10. ε-nondominated set with $\varepsilon = 0.1$ (a) and $\varepsilon = 0.005$ (b) and set of representative nondominated points (c).

cal meaning. The selection of plans should represent the whole nondominated set, but guarantee a certain minimal difference between the points. We developed a method to determine a representative subset of nondominated points in Shao and Ehrgott (2007). The method first constructs an equidistant lattice of points placed with distance d on a simplex S (the reference plane) that supports Y at the minimiser of $e^T y$ over Y and such that $Y \subset S + \mathbb{R}^3_{\geqq}$. For each lattice point q an LP

$$\min\{t : q + te \in Y, t \geqq 0\}$$

is solved. If the optimal value is $\hat{t}$, the point $q + \hat{t}e$ is tested for nondominance. It can be shown that the distance between remaining nondominated points is between d and $\sqrt{3}d$. Figure 11.10 (c) shows a representative set for the same example shown in Figure 11.10 (a) and (b).

Since the representative points are all nondominated the planners now have a choice between several plans. By the theory of linear programming, we know that they are all optimal solutions of some weighted sum problem using importance factors as used in current practice. Moreover, the whole range of such solutions is represented. To support planners in the choice of a plan, visual aids are necessary within a decision suppor system. Planners are used to judging the quality of a plan by looking at isodose curves and dose volume histograms (DVH). The former are colour-wash pictures showing curves of equal dose superimposed on CAT pictures. The latter are plots of the percentage of tumour and critical structures against dose levels, see Figure 11.11.

The representative set of solutions (treatment plans) is stored in a database and input to the software CARINA (Ehrgott and Winz, 2008) which first proposes a balanced solution of (11.2) (with as equal as possible values of α, β, γ) displaying the corresponding DVH and isodose plots as well as some information on available trade-offs. The planner can then specify changes (going to a neighbouring solution, searching for solutions with specific values, or for solu-
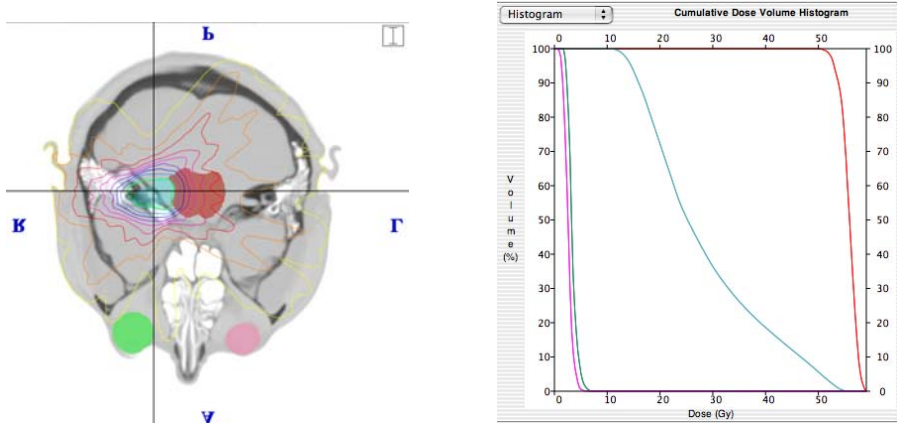
Fig. 11.11. Isodose curves and dose-volume histogram for a brain tumour treatment plan.

tions satisfying some thresholds). This process is continued until the planner accepts a treatment plan.

The interaction with the treatment planner is therefore ex-post, allowing the likely time consuming plan calculation to be decoupled from plan selection. As a consequnce, plan selection becomes faster as it is based on information retrieval from databse, a real-time operation. Moreover, the specification of dose levels is more natural than the "guessing" of importance factors.

11.5.4 Remarks

It is interesting to note that the optimisation model (11.2) tries to characterise dose distribution by a few numbers, whereas the quality is actually judged by the whole DVH. This, of course, is not part of the model. Attempts to specify some points on the DVH curves as constraints in optimisation models exist. But they lead to mixed integer programming models that at this time cannot be solved as multi-objective models.

Throughout the project radiotherapists have been involved in the project. This had several advantages. We obtained valuable information on radiotherapy practice and could ensure to develop usfeul tools that would be accepted by the actual users. Fears that we intend to replace people by software could easily be laid to rest once the radiotherapists understood that we never thought it is possible to replace their role in the treatment design, but that we could improve the planning process. Finally, we made sure to use the tools they are accustomed to work with every day.

The work on this project has thus far resulted in an academic software that allows solution of the multi-objective linear programme (11.2) for 2D and smaller sized 3D problems. The software has been developed in close consultation with treatment planners at Auckland City Hospital. Further work

needs to address numerical issues arising in large 3D problems. Before actual use in clinical practice a lengthy and costly approval process needs to be completed for which the support of a medical software company will be required.

11.6 Supply Chain Management

11.6.1 Problem Description

In supply chains often thousands of individual decisions need to be made and coordinated. Due to the high degree of complexity successive planning approaches are therefore often chosen in practice.

Figure 11.12 shows typical planning tasks that arise in supply chains. These planning tasks are arranged in two dimensions. The first dimension is the "supply chain process". In this dimension, planning tasks are arranged focusing on the most important processes following the flow of goods in supply chains. These are procurement, production, distribution and sales. The second dimension is the "planning horizon". In this dimension the planning tasks are distinguished by their temporal impact on the supply chain. These may be strategic decisions with a long-term impact or operational decisions, which have only an immediate impact in the near future (short-term).

The strategic network planning module covers the long-term decisions across all supply chain processes. It supports the user to determine the structure of the supply network (plant location, distribution system) as well as the product program. Although its results are important for the long-term
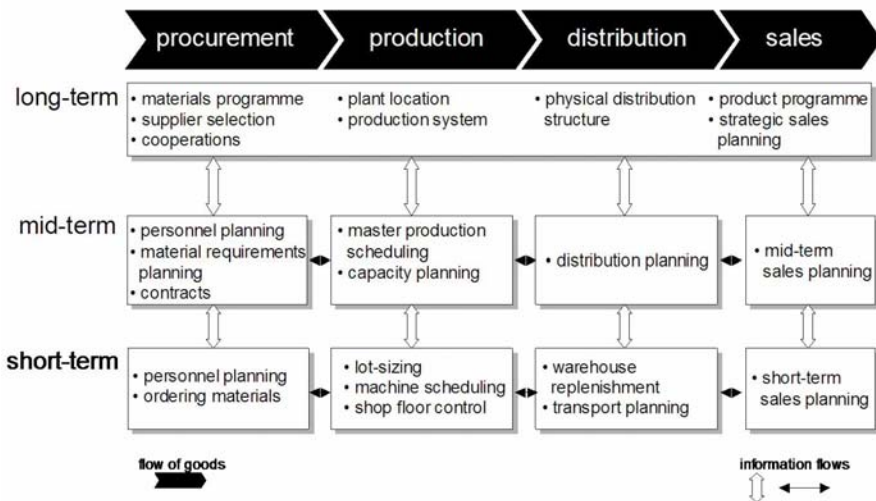

Fig. 11.12. Supply Chain Planning Matrix Fleischmann *et al.* (2005), p. 87.

profitability of supply chains, it is often not a core functionality of Advanced Planning Systems (APS). This is because APS are primarily built to support daily business, whereas strategic decisions are only reviewed periodically and most often not within the regular organization, but rather on a project basis.

The master planning module coordinates procurement, production and distribution on a mid-term level. Its major decision support is about sourcing: Which product is produced at which location and when? Thus, in this module the master production schedule is fixed. However, it is important to anticipate the key characteristics of the lower (short-term) planning levels within this module, because otherwise inconsistent plans (for procurement, production and distribution) will result on the lower planning level.

In the area of distribution and transport planning, distribution related planning tasks are addressed, the latter on a more detailed level (e.g., scheduling of transports, vehicle loading and routing). Production planning and scheduling on the other hand are the two modules that support production related issues in the short-term planning horizon. Finally, purchasing and material requirements planning support the (short-term) procurement of materials and components.

The capabilities offered within mySAP Supply Chain Management (SCM) extend far beyond the scope of this article. The key functionalities we will describe in the following are highlighted in Figure 11.13, which is based on the generic supply chain planning matrix (Figure 11.12). They are part of the SAP Advanced Planner and Optimizer (SAP APO), which is the advanced planning component within the mySAP SCM solution. For more information on SAP APO refer to Bartsch and Bickenbach (2001); Dickersbach (2005).
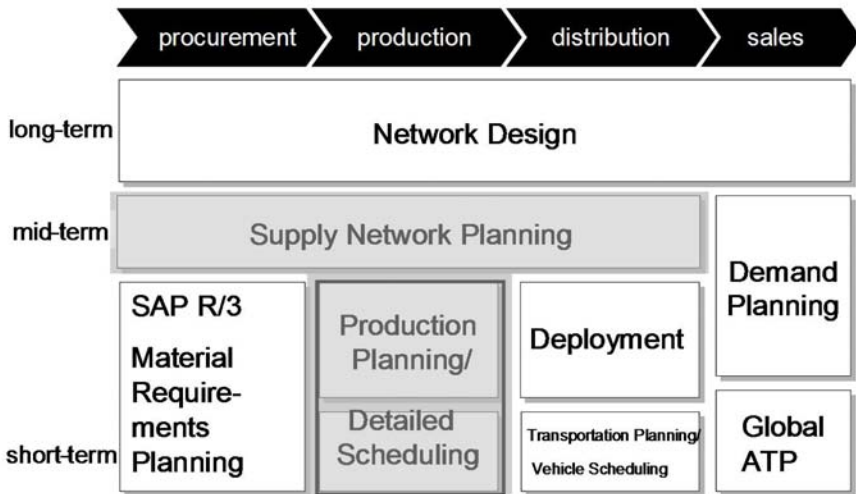

Fig. 11.13. Supply chain planning matrix using mySAP SCM terminology.

Solving a planning problem of this complexity in its entirety within one planning step is neither feasible from an algorithmic perspective nor sensible from a planning process point of view. A hierarchical decomposition of the complete planning problem into a master planning and a production planning and scheduling part addresses planning complexity as well as planning process design issues. In the following, we describe how the planning problem is partitioned into a master planning and a production planning part and which business requirements are addressed on which planning level.

11.6.2 Methodology

Multi Location Optimization in Supply Network Planning (SNP)

In SAP APO, the master planning process is implemented in the Supply Network Planning (SNP) module. SNP offers a multitude of functionalities, not all of which can be described in the limited scope of this article. More details on the SNP module can be found, among others, in Dickersbach (2005). The SNP model contains all relevant locations, i.e. production plants and distribution centres, in the supply network. SNP determines which of the plants produces which quantities of which products in which time periods. On a rough level, SNP also determines which production alternative is used at a specific plant, for instance with regard to ingredients and general process characteristics.

To reduce the complexity of the master planning model, not all products are considered in the SNP optimization run. The selection is made by flagging specific products as not relevant for SNP planning. SNP planning takes into account all products produced in a location, all products for which a stock transfer between locations is possible, externally procured active ingredients, goods for resale and selected forming auxiliaries. Not relevant to SNP are most raw materials, most forming auxiliaries as well as packaging materials. A similar logic is used for resources. Only bottleneck resources are selected for SNP planning.

The concentration on key products and bottleneck resources also results in a significant simplification of the recipes[1] used in SNP, which are derived from the more detailed recipes used in Production Planning and Detailed Scheduling (PP/DS) and the attached enterprise resource planning (ERP) system. Furthermore, compared to the recipes used on the PP/DS level, not all setup and cleaning operations are considered in SNP recipes. Small setup operations are normally neglected while key setup activities which are relevant for campaign planning on bottleneck resources due to their long duration or high costs are considered. To account for the resource capacity consumed by small setup and cleaning operations, a loss factor is applied to calculate the resource capacity for SNP planning.

---

[1] In APO, a recipe is commonly referred to as PPM (production process models) or PDS (production data structure).

One of the main aspects of the SNP planning process is the cost-based plan determination. The following cost types are used to build a cost model which represents the business scenario of value base planning:

- Penalties for not meeting customer demand / forecast,
- Penalties for late satisfaction of customer demand / forecast (location product specific)
- Penalties for not meeting safety stock / safety days' supply requirements (location product specific, linear or piecewise linear)
- Storage cost (location product specific)
- Penalty for exceeding maximum stock level / maximum coverage (location product specific, linear or piecewise linear)
- External procurement cost (linear or piecewise linear, location product specific)
- Handling in / out cost (location product specific)
- Transportation cost (transportation lane, product and means of transport specific, linear or piecewise linear)
- Variable production cost (production process specific, linear or piecewise linear)
- Fixed production cost / setup cost (production process specific)
- Resource utilization cost (resource specific)
- Costs for additional resource utilization (e.g. use of additional shifts, resource specific)
- Cost for falling below minimum resource utilization.

The definition of the cost model is of crucial importance for controlling the behaviour of the SNP optimizer. One of the central questions is whether to maximize service level – which usually means using high penalties for non and late delivery – or to maximize profits – which requires use of realistic sale prices. In the case study scenario, the non delivery cost levels reflect real sale prices sufficiently close to enable a profit maximization logic.

Another important feature of the case study scenario and the resulting cost model is inventory control. High seasonality effects and long campaign durations necessitate considerable build-up of stocks. To avoid an unbalanced build-up of stock, soft constraints for safety stock and maximum stock levels are used. To achieve an even better inventory levelling across products and locations, piecewise linear cost functions for falling below safety stock as well as for exceeding maximum stock levels are employed. In SNP optimization all revelevant constraints can be considered, including

- capacities for production, transportation, handling and storage resources,
- maximum location product specific storage quantities,
- minimum, maximum and fixed production lot sizes,
- minimum, maximum and fixed transportation lot sizes,
- minimum production campaign lot sizes.

An optimization model which considers all these constraints – especially those which can only be modelled using binary or general integer variables – can be highly complex.

Production Planning and Detailed Scheduling (PP/DS)

The short term planning process is dealt with in the Production Planning and Detailed Scheduling module within SAP APO.

PP/DS focuses on determining an optimal production sequence on key resources. In PP/DS, a more detailed modelling than on the SNP planning level is chosen. On the basis of the results determined in SNP optimization, a detailed schedule which considers additional resources and products is created. This schedule is fully executable and there is no need for manual planner intervention, even though manual re-planning and adjustments are fully supported within the PP/DS module. An executable plan can only be ensured by considering additional complex constraints in PP/DS optimization. These additional constraints include:

- Time-continuous planning
- Sequence-dependent setup and cleaning operations.

As the value based planning part is handled within SNP, the PP/DS optimizer uses a different objective function than the SNP optimizer. The following goals can be weighted in the objective function, which is subject to minimization:

- Sum of delays and maximum delay against given due dates
- Setup time and Setup cost
- Makespan (i.e. time interval between first and last activity for optimizing the compactness of the plan)
- Resource cost (i.e. costs associated with the selection of alternative resources)

The main objective of the PP/DS optimizer run in the scenario at hand is to minimize setup times and costs on resources without incurring too much delay against the order due dates. For some resource groups, resource costs are also used to ensure that priority is given to the 'best' (i.e. fastest, cheapest, etc.) resources.

11.6.3 Remarks

We have seen that both in Supply Network Planning and in Detailed Scheduling there are a huge number of objectives to be minimized. However these objectives can be mastered by forming a 4-level hierarchy.

On the root or top level, two dimensions of the second level can be differentiated: Service degree and real costs. The objective of real costs differentiates at the third level between for example:

- storage costs: the minimization of inventory by weighting the inventory of each storage location by an estimated cost factor
- safety costs: the minimization of the risk of getting our of stock by weighting the risk of each storage location by a cost factor
- setup costs: the minimization the overhead of change over for each resource by weighting each change over by a cost factor.

The objective of service degree differentiates at the third level between for example:

- delay penalties: the minimization of delay for each demand or customer order by weighting the priority of the customer
- non delivery penalties: the minimization of non delivery for each demand or customer order by weighting the priority of the customer.

Only for the top level are weighting factors not appropriate. The planner wants to see several alternative solutions of the Pareto front of these two objectives: By how much would costs increase if we wish to achieve a better service level? A high service level is clearly an important objective, but there is no direct cost measure for a delay. Summarizing in Advanced Planning for Supply Chain Management we can focus on an optimization problem with just two objectives maximizing service degree while minimizing the costs. In particular, two dimensional visualization of the Pareto front and representative solutions of this front are needed and sufficient.

11.7 Interactive Processes for Aircraft Design

11.7.1 Problem Description

Interactive Evolutionary Computation (IEC) has started to capture the fascination of researchers from fields as diverse as art, architecture, data mining, geophysics, medicine, psychology, robotics, and sociology. Takagi (2001) outlined many of these applications in his overview paper. However, to this day only very few researchers have applied IEC to the problem of engineering and design of complicated artifacts. While the main reason for the slow pace of adoption in engineering is mostly open for speculation, it is partially a result of the field's reluctance to accept new methods, like Genetic Algorithms, as well as the field's already heavy reliance on automated optimization processes that leave decidedly little room for subjectivity. While the reliance of engineers on analysis tools requires interactive evolutionary techniques to utilize them in the fitness generation, it is also true that many design decisions in practice are made through gut feel and intuition rather then analysis. Recognizing that fact, this example identifies an IEC approach to design that allows for automatic fitness calculation through analysis as well as selection and fitness assignment by the human designers directly.

There are few things humans build that are more complicated than aircraft. Not only are the reliability requirements enormous, given the fatal consequences of failure, but the system itself strides a multitude of areas in physics, such as aerodynamics, thermodynamics, mechanics, and materials. This convolution of disciplines has led historically to a very sequential design process, tackling the various disciplinary issues separately: aerodynamicists only tried to maximize the performance of the wing (or even just an airfoil), propulsion engineers tried to build the largest engines, structural engineers tried to build the sturdiest airframe, while material scientists attempted to only utilize the lightest and sturdiest materials. As a consequence, the design process itself was a highly inefficient iterative process of ever changing airplane configurations, only reconciled by rare, experienced individuals that were proficient in all (or at least many) disciplines. As these people retired, and significant computational power became available, a new design process emerged, attempting to satisfy the concerns of all disciplines concurrently: Multidisciplinary (Design) Optimization. MDO is inherently a multicriteria optimization problem, since each discipline contributes at least one objective function that potentially conflicts with the objective(s) of the other disciplines. The following example demonstrates the ability of one MCDM technique, Interactive Evolutionary Design, to address the difficult task of balancing the different disciplinary objectives when determining the preliminary design configuration of a Supersonic Business Jet.

Figure 11.14 outlines the interactive evolutionary design process employed for this application example (see Bandte and Malinchik, 2004, for background discussion). After the problem is set-up by defining design variables, objectives and constraints and sufficient feasibility has been established, a GA is being interrupted after several generations to display the current population via spider-graphs and Pareto Frontier displays for objective values as well as visualizations of the aircraft configurations. Based on this information the
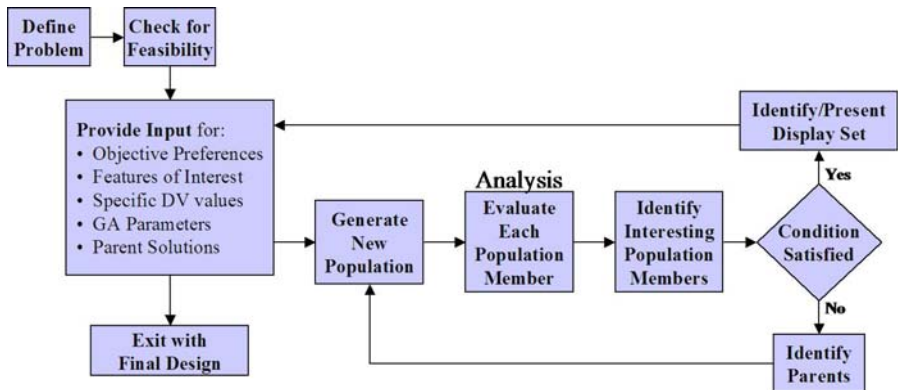

Fig. 11.14. Integrated interactive evolutionary design process.

designer can make some choices regarding objective preferences and features of interest, redirecting search and influencing selection respectively. To limit the scope of this example, only the redirection of the search through objective preferences is being implemented here. However, as exemplified later, designer selection of features of interest is an important part of interactive evolutionary design and should not be neglected in general. The following sections lay out in detail all tasks performed over several iterations for this example.

As in any design problem, the first step is to define the independent parameters, objectives and constraints, as well as evaluation functions that describe the objectives' dependencies on the independent variables. For this interactive evolutionary design environment, this step also identifies the genotype representation of a design alternative, the fitness evaluation function, influenced by the objectives, and how to handle design alternatives that violate constraints.

The supersonic business jet is described by five groups of design variables, displayed in a screen shot presented in Figure 11.15. The first group, general, consists of variables for the vehicle, some of which could be designated design requirements. The other four groups contain geometric parameters for the wing, fuselage, empennage, and engine. The engine group also entails propulsion performance parameters relevant to the design. All in all, the chromosome for this supersonic business jet contains 35 variables that can be varied to identify the best solution.

A mix of economic, size, and vehicle performance parameters were chosen as objectives in this example, with a special emphasis on noise generation, since it is anticipated to be a primary concern for a supersonic aircraft. Hence, for the initial loop the boom loudness and, as a counter weight, the acquisition cost are given slightly higher importance of 20%, while all other objectives are set at 10%. Furthermore, certain noise levels could be prohibitively large and prevent the design from getting regulatory approval. Hence, some of the noise objectives have to have constraint values imposed on them. In addition to these constraints, the design has to fulfill certain FAA requirements regarding take-off and landing distances as well as approach speed. Furthermore, the amount of available fuel has to be more than what is needed for the design mission. Finally, fitness is calculated via a weighted sum of normalized objective values, penalized by a 20% increase in value whenever at least one constraint is violated. Note that the "best solution" is identified as the one with the lowest fitness, i.e. objective function values. All constraints, objectives, normalization values and preferences are also displayed in Figure 11.15.

11.7.2 Methodology and Interactive Scheme

Run GA

Since the initial objective preferences were already specified at problem definition, the GA can be executed next without requiring further input from the designer. The GA chosen for this example is one of the most general found
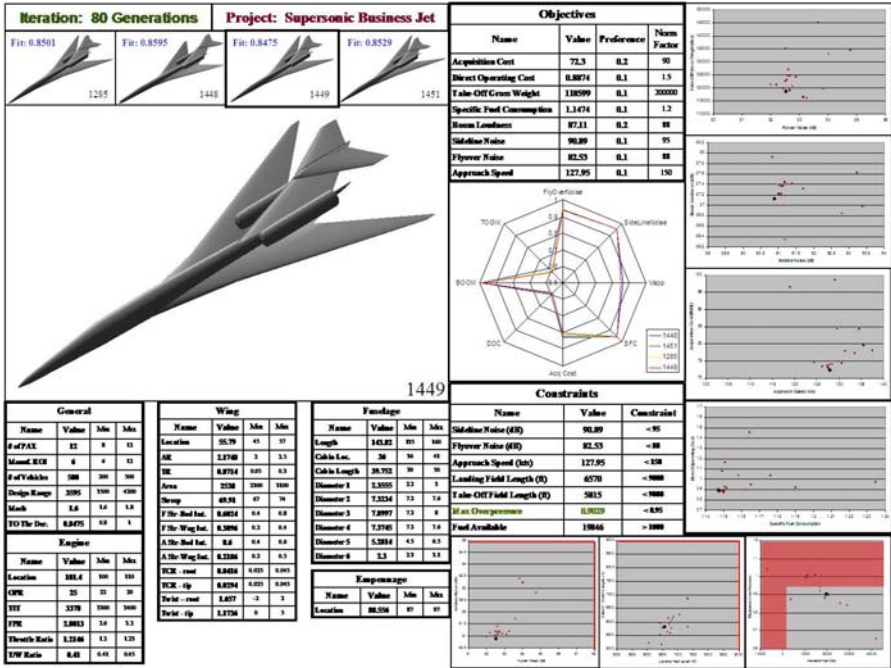
Fig. 11.15. Screen shot: Display of information after 80 generations.

in the literature Holland (1975); Mitchell (1996); Haupt and Haupt (1998). It has a population size of 20 and makes use of a real valued 35-gene representation, limited to the ranges selected at problem set-up. New generations are created from a population with a two individuals elite pool and proportionate probabilistic selection for crossover. The crossover algorithm utilizes a strategy with one splice point that is selected at random within the chromosome. Since the design variables are grouped in logical categories, this crossover algorithm enables, for example, a complete swap of the engine or fuselage-engine assembly between parents. Parent solutions are being replaced with offspring. Each member of the new population has a 15% probability for mutation at ten random genes, sampling a new value from a uniform distribution over the entire range of design variable values. The GA in this example is used for demonstration purposes only and therefore employs just a small population. A population size of 50 to 100 seems more appropriate for a more elaborate version of the presented interactive evolutionary design approach.

Display Information

Once the GA has executed 80 generations, it is interrupted for the first time to display the population of design alternatives found to this point. The designers are presented with information that is intended to provide maximum insight

into the search process and the solutions it is yielding. In order to allow for a reasonable display size of the aircraft configuration, only the four best design alternatives, based on fitness and highest diversity in geometrical features, are presented in detail on the top of the left hand side of the display. A screen shot of the displayed information is presented in Figure 11.15, highlighting the individual with the best/lowest fitness, which is also enlarged to provide the designers with a more detailed view of the selected configuration. The design variable values for the highlighted alternative and their respective ranges are presented below this larger image, completing the "chromosome information" on the left hand side of the pane.

On the right hand side of the pane, the designers can find the objective and constraint information pertaining to the population and the highlighted individual. On the top, a simple table outlines the specific objective values for the highlighted alternative, as well as the objective preferences and normalization factors used to generate the fitness values for the current population. Below the table, a spider graph compares the four presented alternatives on the basis of their normalized objective values, while to the right four graphs display the objective values for the entire population, including its Pareto frontier (highlighted individual in black). Below the spider chart, a table lists the constraint parameter values for the highlighted alternative as well as the respective constraint values. A green font represents constraint parameter values near, orange font right around, and red font way beyond the constraint value. Finally at the bottom, three graphs display the population with respect to its member's constraint parameter values as well as the infeasible region, superimposed. These graphs in particular indicate the level of feasibility in the current population.

Provide Input

This step represents the central interaction point of the human with the IEC environment. Here they process the information displayed and communicate preferences for objectives, features of interest in particular designs, whether specific design variable values should be held constant in future iterations, what parameter setting the GA should run with in the next iteration (e.g. a condition that identifies the end of the GA iteration), or whether specific design alternatives should serve as parents for the next generation.

Analyzing the data provided, it is noticeable that all objectives except for the boom loudness are being satisfied well. Consequently, in an attempt to achieve satisfactory levels for the boom loudness in the next iteration, its preference is increased to 30%, reducing the acquisition cost's importance to 10%. This feedback is provided to the GA via a pop-up screen (not displayed here) that allows the designer to enter the new preference values for the next iteration. With this new preference information the GA is executed for another 80 generations.

Second Iteration, 160 Generations

After the GA has executed an additional 80 generations, it is apparent from the objective values that the last set of preferences did not emphasize the boom loudness and sideline noise enough, since boom loudness did not improve (from 87.11 to 87.12 dB) and the sideline noise got worse (from 90.89 to 93.65 dB). Consequently, for the next iteration the importance of both is increased to 35% while all other objectives are reduced to 5%.

Third Iteration, 240 Generations

Examining the population after another 80 generations yields that the last set of preferences still did not emphasize the boom loudness enough, since boom loudness improved only marginally (from 87.12 to 86.96 dB). On the other hand, sideline noise did improve significantly (from 93.65 to 90.94 dB), so that for the next iteration all emphasis can be given to boom loudness. To keep the score even for all other objectives, they are being kept at 5% with boom loudness at 65%.

Fourth Iteration, 400 Generations

For this iteration 160 more generations were executed to produce the population displayed in the screen shot presented in Figure 11.16. In part due to the longer GA run, a very good solution, #7172, is found after 400 generations with largely improved values for almost all the objectives. This result is somewhat surprising, considering most objectives had only a 5% level of preference and the one objective with 65%, boom loudness, improved only marginally.

This result can be attributed to an exemplified effect from summing correlated objective function values that are caused by similar design alternatives (with similar design variable settings) exhibiting similarly good (or bad) values for all objectives except boom loudness. The fact that boom loudness is a conflicting objective, specifically with sideline noise, can also be observed from the pronounced Pareto frontier in the second objective chart from the top.

However, the presented solution after 400 generations seems to satisfy the objectives better than the published solution in Buonanno *et al.* (2002), generated by MATLAB©'s Fmincon function (The MathWorks Inc. (2008)). So it could be concluded that none of these objective values are dramatically out of sync or range and the presented individual is the final solution.

11.8 Land Use Planning

11.8.1 Problem Description

The work described here was motivated by problems of land use allocation or re-allocation in the Netherlands. Land which is already intensely developed
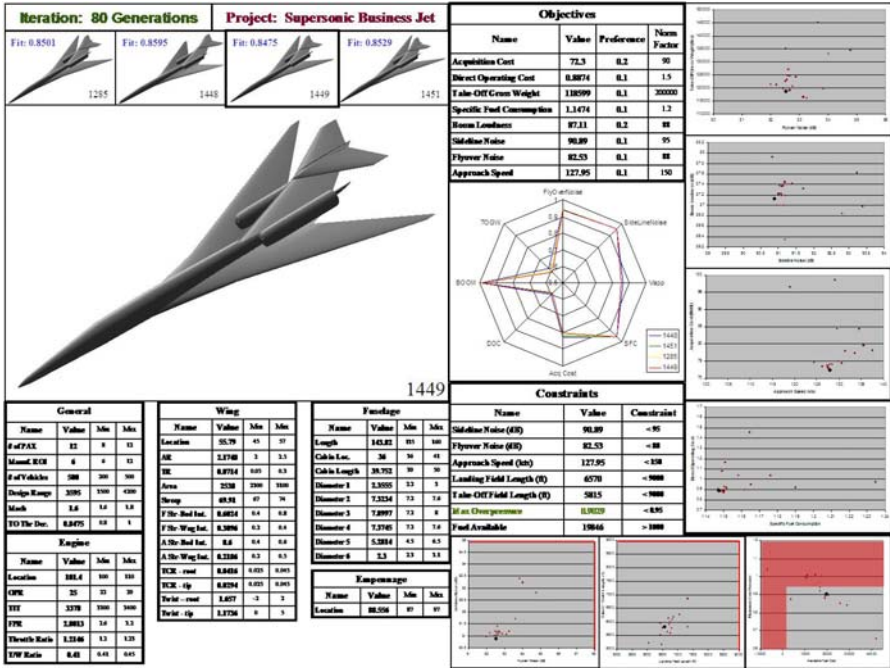
Fig. 11.16. Screen shot: Fourth iteration after 400 generations.

has often to be redeveloped to meet current needs for agriculture, residence, recreation and industry, while at the same time recognizing conservation goals (including possible restoration of some land to approximately pristine conditions). The initial model development was based on a specific region near Amsterdam (the Jisperveld), as briefly described in Section 5 of Stewart *et al.* (2004). However, the longer term intention is to build the model into a general land use planning decision support system (LUDSS). The function of such an LUDSS would be to generate a small number of plans which can be evaluated holistically by decision makers or planners. They would then indicate which features of the plan they like or dislike, which would lead to a readjustment of goals in the LUDSS and the generation of a new solution. This process may be repeated until planners are satisfied that no substantial further improvements are likely.

The model represents the region under consideration by a rectangular grid of (say) $R \times C$ equal-sized cells. It is then assumed that one and only one land use (from a set of Λ possible uses) is allocated to each grid. Formally, we define binary variables $x_{rc\ell}$, such that $x_{rc\ell} = 1$ if land use ℓ is allocated to cell (r,c) and $x_{rc\ell} = 0$ otherwise. For ease of notation we shall denote the three dimensional array of all $x_{rc\ell}$ values by $\mathbf{x}$. Typical constraints on $\mathbf{x}$ would include exclusions of certain land uses from certain cells (corresponding

$x_{rc\ell}$ set to zero) and upper and lower bounds on the total area allocated to a particular land use (i.e. on $\sum_{r=1}^{R} \sum_{c=1}^{C} x_{rc\ell}$).

Some objectives relate to directly quantifiable costs and benefits, and tend to be additive in nature. For example, if all such objectives (without loss in generality) are expressed as costs then:

$$f_i(\mathbf{x}) = \sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{\ell=1}^{\Lambda} \beta_{rc\ell}^i x_{rc\ell}$$

where $\beta_{rc\ell}^i$ is the cost in terms of objective i associated with allocating land use ℓ to cell (r, c).

As initially described by Aerts *et al.* (2005), however, a critical management objective is to ensure that land uses are sufficiently compact to allow integrated planning and management. Aerts *et al.* (2005) introduce essentially one measure of compactness, related to the numbers of cells adjacent to cells of the same land use. This concept was extended in our work by means of a more detailed evaluation of the fundamental underlying management objectives. Defining a *cluster* of cells as a connected set of cells allocated to a single land use, three measures of performance for each land use type were identified as follows:

- *Numbers of clusters for each land use, C_ℓ:* These measure the degree of fragmentation of land uses, and minimization of the number of clusters would seek to ensure that areas of the same land use are connected as far as possible.
- *Relative magnitude of the largest cluster for each land use:* Maximization of the ratio $L_\ell = n_\ell^L / N_\ell$ is sought, where n_ℓ^L and N_ℓ are respectively the number of cells in the largest cluster and the total numbers of cells allocated for land use ℓ. If multiple clusters are formed, then it would often be better to have at least one large consolidated cluster, than for all clusters to be relatively small.
- *Compactness of land uses, denoted by R_ℓ,* defined by a weighted average across all clusters for land use ℓ of the ratio of the perimeter to the square root of the area of the cluster. This measure should be minimized as a compact area for one land use (e.g. a square or circular region) may be easier to manage than a long thread-like cluster.

The above measures define an additional 3Λ objectives, as the compactness goals need to be achieved for each land use individually. Furthermore, the calculation of C_ℓ, L_ℓ and R_ℓ require the execution of a clustering algorithm, so that these additional objectives are non-linear and computationally expensive. The total number of objectives is thus $k = k_0 + 3\Lambda$, where k_0 is the number of additive objectives.

11.8.2 Methodology

In view of the large number of objectives, it is not practical to seek to represent the efficient frontier in full. The approach adopted was thus based on sampling the efficient frontier by optimizing an aggregate measure of performance subject to the constraints on **x**, for each of a number of different aggregations. The aggregation chosen was that of the *scalarizing function* introduced by Wierzbicki (1999) in the context of his *reference point* methodology, except that we chose a smoother function than that based on the maximum operator. Thus for any given *reference point* (which can be viewed as a set of goals or aspiration levels for each objective), say $g_1, g_2, \ldots, g_k$, the scalarizing which is to be minimized is defined by:

$$\sum_{i=1}^{k} \left[\frac{f_i(\mathbf{x}) - I_i}{g_i - I_i} \right]^4 \qquad (11.3)$$

where I_i is the ideal (best achievable measure of performance) for objective i.

Constrained minimization of (11.3) with respect to **x** is a non-linear combinatorial optimization problem, with the added complexity that most of the functions cannot be evaluated explicitly in closed form (but are are derived as outputs from a clustering algorithm). Stewart *et al.* (2004) describe a special purpose genetic algorithm (GA), designed to exploit a number of special characteristics of the land use planning problem models, both in the generation of population elements and in the execution of cross-overs (see cited reference for details).

It is interesting to emphasize at this point that the chosen methodology includes elements from conventional multiobjective optimization and from evolutionary optimization, thus representing an integration of the two themes of the present volume.

11.8.3 Interactive Scheme and Results

In implementing the algorithm within an LUDSS, the process starts by selecting one or more tentative reference points, perhaps a central reference point (all goals positioned midway between ideals and worst performance levels) and a selection of reference points which favour each individual goal in turn. Each individual solution generated will be efficient (to the level of optimization accuracy achieved by the GA), and so represents a point on the efficient frontier. In response to assessments by the user as to the direction in which it is desired to move the solution, the reference points are adjusted and the optimizations repeated. In this way, the user is able incrementally to explore the efficient frontier until such time as a satisfactory solution, or short list of possible solutions, is found.

A detailed case study in the use of this system is given in Janssen *et al.* (2007). An illustration of the manner in which the interactions may progress

is given in Figure 11.17, which presents three land use maps. The numbers in the maps indicate nine potential land use types, namely: 1. Intensive agriculture; 2. Extensive agriculture; 3. Residence; 4. Industry; 5. Day recreation; 6. Overnight recreation; 7. Wet natural area; 8. Water (recreational); and 9. Water (limited access).
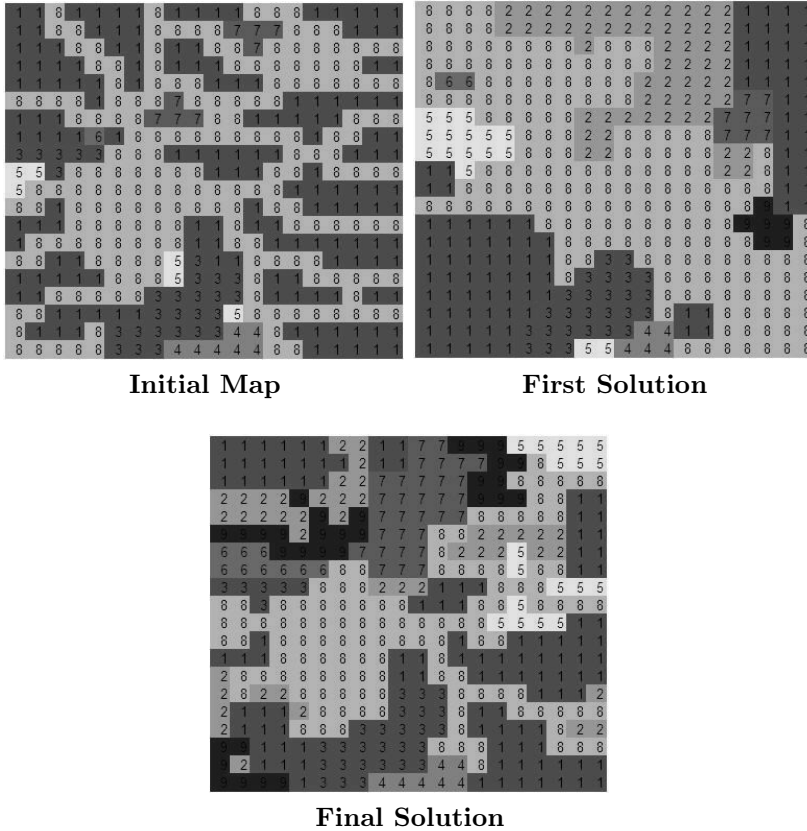

Initial Map   First Solution

Final Solution

Fig. 11.17. Three land use maps generated from the LUDSS.

The upper left hand map displays the original land use pattern. The upper right hand map was generated in the first optimization step, and provides a very compact allocation of land uses. However, the costs were deemed to be too high, largely because of the extent of agricultural land reclaimed from wet areas. For this reason, the priority on the cost attribute was increased. Some more fragmentation of the land uses was then re-introduced, much of which was acceptable except for that of the agricultural land. Also, the values associated with conservation goals were found to be unsatisfactory. Adjustment

of these priorities led after the 8th iteration to the lower map in Figure 11.17, and this was found to represent a satisfactory compromise.

11.9 Engineering Design Problems with Large Numbers of Objectives

11.9.1 Problem Description: Cable-Stayed Bridges

Cable-stayed bridges are gaining much popularity due to their beautiful shape. During and after construction, this kind of bridge needs to have the cable length adjusted in order to attain errors of cable tension and camber (the configuration of the girders of the bridge) within some allowable range.

To this end, the following criteria are considered (Nakayama *et al.* (1995)):

- residual error in each cable tension,
- residual error in camber at each node,
- amount of shim adjustment for each cable,
- number of cables to be adjusted.

Since the change of cable rigidity is small enough to be neglected with respect to cable length adjustment, both the residual error in each cable tension and that in each camber are assumed to be linear functions of the amount of shim adjustment.

Let us define n as the number of cables in use, $\Delta T_i \ (i = 1, \ldots, n)$ as the difference between the designed tension values and the measured ones, and x_{ik} as the tension change of i-th cable caused from the change of the k-th cable length by a unit. The residual error in cable tension caused by the shim adjustment is given by

$$p_i = |\Delta T_i - \sum_{k=1}^{n} x_{ik} \Delta l_k| \qquad (i = 1, \ldots, n)$$

Let m be the number of nodes, $\Delta z_j \ (j = 1, \ldots, m)$ the difference between the designed camber values and the measured ones, and y_{jk} the camber change at j-th node caused from the change of the k-th cable length by a unit. Then the residual error in the camber caused by the shim adjustments of $\Delta l_1, \ldots, \Delta l_n$ is given by

$$q_j = |\Delta Z_j - \sum_{k=1}^{n} y_{jk} \Delta l_k| \qquad (j = 1, \ldots, m)$$

In addition, the amount of shim adjustment can be treated as objective functions of

$$r_i = |\Delta l_i| \qquad (i = 1, \ldots, n)$$

The upper and lower bounds of shim adjustment inherent in the structure of the cable anchorage are as follows:

$$\Delta l_{Li} \le \Delta l_i \le \Delta l_{Ui} \qquad (i = 1, \ldots, n). \tag{11.4}$$

11.9.2 Methodology

Now we have a multi-objective optimization problem in which $p_1, \ldots, p_n$, $q_1, \ldots, q_m$ and $r_1, \ldots, r_n$ are to be minimized under the constraint (11.4). Some large scale bridges have around 100 cables at each side, so that the problem results easily in a very large number of objective functions. For this multi-objective optimization problem, engineers in bridge construction have tried to apply goal programming (Charnes and Cooper, 1961), in which they want to get a desirable solution by adjusting weights imposed on criteria. However, it has been pointed out in the literature (e.g. see Nakayama, 1995) that this task is very difficult even for simple problems. In addition, the shim adjustment must ususally be done during a relatively short period (say, 2:00 am to 8:00 am) with a stable temperature, because the cable length is greatly affected by change of temperature. Therefore, the decision of cable length adjustment must be made very quickly. Also, due to this reason, the traditional goal programming approach is not satisfactory for practical use in this problem.

On the other hand, an interactive multi-objective programming technique has been developed, called the satisficing trade-off method (Nakayama and Sawaragi, 1984). The method is one of aspiration level approaches to multi-objective optimization, which are observed to be effective in many practical problems because they are very simple and easy to implement and do not require any mathematical consistency of decision makers' judgment, and in addition take aspiration levels of decision makers as a probe rather than weights imposed on criteria.

Figure 11.18 shows the graphical user interface (GUI) for the erection management system of a cable stayed bridge using the satisficing trade-off method. The residual error of each criterion and the amount of shim adjustment are represented by bar graphs. The aspiration level is inputted by a mouse on the graph. After solving the auxiliary min-max problem, the Pareto solution according to the aspiration level is represented by another bar graph in a similar fashion. If the designer is not satisfied with the Pareto solution displayed, he/she can revise the aspiration level by means of mouse operations, and the process repeated.

This procedure is continued until the designer obtains a desirable shim-adjustment. The interactive operation using the GUI is very easy for the designer, and the visual information on trade-offs among criteria is user-friendly. The software has been used for real bridge construction, for example tje Tokiwa Swan Bridge (Ube City) and the Karasuo Harp Bridge (Kita-Kyusyu City) in 1992.

One of the important aspects of such a problem with a large number of objective functions is the graphical user interface. As can be easily seen in Fig.11.18, it is not too difficult for designers to make a trade-off analysis on the basis of the displayed visual information, even in cases with hundreds of objective functions. However, it might be difficult or even impossible to
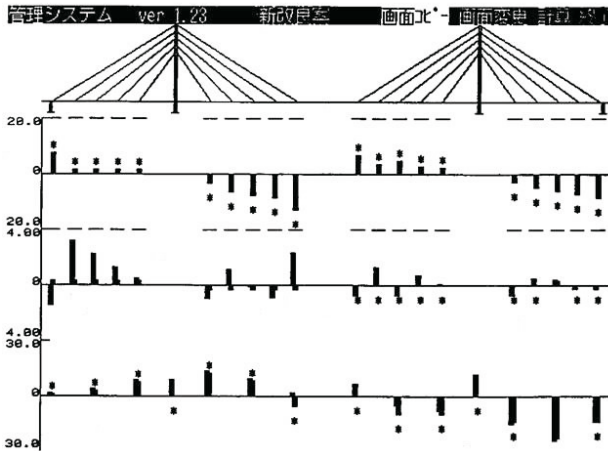
Fig. 11.18. Erection Management System of Cable-stayed Bridge.

grasp the total trade-off context on the basis of numerical information only, for problems with such a large number of objective functions.

11.9.3 A Further Application: Lens Design

Another good example with a large number of objective functions is lens design. There are many kinds of lenses such as copier, camera, medical instruments and so on. Above all, lenses in semiconductor chip production are very expensive (of the order of million dollars), and hence have to be designed very carefully.

In lens design, there are around 200 design variables such as

- kinds of glass,
- number of lenses,
- diameter,
- curvature,
- distance between lenses,

and around 400 criteria such as

- cost
- weight
- criteria for images:
  - aberration
  - chromatic
  - spherical
  - astigmatism
  - coma

  – distortion
  – curvature of field
  – color balance
  – resolution
  – MTF
  – CCI

In lens design, there is the further difficulty of nonlinear optimization in addition to the large number of objective functions: Scalarized optimization problems are usually highly nonlinear and highly multi-modal. Moreover, those functional forms are not given explicitly in terms of design variables. Those function values are evaluated on the basis of some kind of simulation (ray trace). Therefore, it is difficult to obtain a global minimum for the objective function.

So far, engineers use specific software in lens design. Their main attention has been directed to how they obtain a global optimum for the scalarized objective function, while the linearly weighted sum scalization function is applied. It will be surely a good subject to investigate how interactive multiobjective optimization techniques can work in lens design.

11.10 Concluding Comments

In what sense are the above applications different to other optimization studies? Clearly, the distinction lies in the multiplicity of objectives which are central to the applications discussed here. In common with more general approaches to multiple criteria decision making (MCDM), those applying the multiobjective methods start by careful problem structuring to identify the underlying objectives, and to represent these in meaningful manner, much as has been described in Chapter 3 of Belton and Stewart (2002).

Some of the approaches reported in the case studies do ultimately seek to identify an overall mathematical objective function as a surrogate measure of performance to be "maximized" or "minimized", but:

• This is done only after careful attention to tradeoffs between objectives and clear recognition that these tradeoffs may change as one explores the decision space;
• The methods are applied interactively, with systematic changes in formulation (revising goals or value tradeoffs) in the light of preference information (as described in other Chapters of this book concerning interactive methods), and thus providing a means of implicit exploration of the Pareto frontier.

Other reported approaches avoid the use of surrogate objective functions, by seeking to identify the Pareto frontier explicitly, leaving the user or ultimate client to explore the options visually before making the final selection. Unfortunately, it is difficult to provide an unambiguous visualization of the frontier

for more more than two or three objectives, although the Chapter 9 in this book seeks to extend such opportunities.

The clear challenge to future research lies precisely in the interface between these implicit and explicit methods of searching the Pareto Frontier. An opportunity may lie in using the interactive methods using surrogate measures of performance for an initial exploration, but using the explicit search methods (linked to appropriate visualization) to refine the exploration across those objectives which are found most critical to the final decisions in the most promising regions of the decision space.

References

Aerts, J.C.J.H., van Herwijnen, M., Janssen, R., Stewart, T.J.: Evaluating spatial design techniques for solving land-use allocation problems. Journal of Environmental Planning and Management 48(1), 121–142 (2005)

Alexandrov, N.M., Dennis, J.E., Lewis, R.M., Torczon, V.: A trust region framework for managing use of approximation models in optimization. Journal on Structural Optimization 15(1), 16–23 (1998)

Arima, T., Sonoda, T., Shirotori, M., Tamura, A., Kikuchi, K.: A numerical investigation of transonic axial compressor rotor flow using a low-Reynolds-number $k-\epsilon$ turbulence model. ASME Journal of Turbomachinery 121(1), 44–58 (1999)

Bandte, O., Malinchik, S.: A broad and narrow approach to interactive evolutionary design – an aircraft design example. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 883–895. Springer, Heidelberg (2004)

Bartsch, H., Bickenbach, P.: Supply Chain Management mit SAP APO. Galileo Press, Bonn (2001)

Belton, V., Stewart, T.J.: Multiple Criteria Decision Analysis: An Integrated Approach. Kluwer Academic Publishers, Boston (2002)

Benson, H.: An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. Journal of Global Optimization 13(1), 1–24 (1998)

Buonanno, M., Lim, C., Mavris, D.N.: Impact of configuration and requirements on the sonic boom of a quiet supersonic jet. Presented at World Aviation Congress, Phoenix, AZ (2002)

Charnes, A., Cooper, W.: Management Models and Industrial Applications of Linear Programming, vol. 1. John Wiley, New York (1961)

Dickersbach, J.T.: Supply Chain Management with APO, 2nd edn. Springer, Berlin (2005)

Ehrgott, M., Winz, I.: Interactive decision support in radiation therapy treatment planning. OR Spectrum 30, 311–329 (2008)

Ehrgott, M., Holder, A., Reese, J.: Beam selection in radiotherapy design. In: Linear Algebra and Its Applications, vol. 428, pp. 1272–1312 (2008a)

Ehrgott, M., Hamacher, H.W., Nußbaum, M.: Decomposition of matrices and static multileaf collimators: A survey. In: Alves, C.J.S., Pardalos, P.M., Vicente, L.N. (eds.) Optimization in Medicine. Springer Series in Optimization and Its Applications, vol. 12, pp. 25–46. Springer Science & Business Media, New York (2008b)

Emmerich, M., Giannakoglou, K., Naujoks, B.: Single and multi-objective evolutionary optimization assisted by Gaussian random field meta-models. IEEE Transactions on Evolutionary Computation 10(4), 421–439 (2006)

Fleischmann, B., Meyr, H., Wagner, M.: Advanced planning. In: Stadtler, H., Kilger, C. (eds.) Supply Chain Management and Advanced Planning. Concepts, Models, Software and Case Studies, 3rd edn., pp. 81–106. Springer, Berlin (2005)

Hasenjäger, M., Sendhoff, B., Sonoda, T., Arima, T.: Three dimensional evolutionary aerodynamic design optimization using single and multi-objective approaches. In: Schilling, R., Haase, W., Periaux, J., Baier, H., Bugeda, G. (eds.) Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems EUROGEN 2005, Munich, FLM (2005)

Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. John Wiley & Sons, New York (1998)

Holder, A.: Designing radiotherapy plans with elastic constraints and interior point methods. Health Care Management Science 6, 5–16 (2003)

Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)

Janssen, R., van Herwijnen, M., Stewart, T.J., Aerts, J.C.J.H.: Multiobjective decision support for land use planning. Environment and Planning B, Planning and Design. To appear (2007)

Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing 9(1), 3–12 (2005)

Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments – A survey. IEEE Transactions on Evolutionary Computation 9(3), 303–317 (2005)

Jin, Y., Olhofer, M., Sendhoff, B.: On evolutionary optimization with approximate fitness functions. In: Genetic and Evolutionary Computation Conference, pp. 786–792. Morgan Kaufmann, San Francisco (2000)

Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Transactions on Evolutionary Computation 6(5), 481–494 (2002)

Jin, Y., Olhofer, M., Sendhoff, B.: On evolutionary optimization of large problems using small populations. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3611, pp. 1145–1154. Springer, Heidelberg (2005)

Jin, Y., Zhou, A., Zhang, Q., Tsang, E.: Modeling regularity to improve scalability of model-based multi-objective optimization algorithms. In: Multiobjective Problem Solving from Nature. Natural Computing Series, pp. 331–356. Springer, Heidelberg (2008)

Larranaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2001)

Li, X.-D.: A real-coded predator-prey genetic algorithm for multiobjective optimization. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 207–221. Springer, Heidelberg (2003)

Lim, D., Ong, Y.-S., Jin, Y., Sendhoff, B., Lee, B.S.: Inverse multi-objective robust evolutionary optimization. Genetic Programming and Evolvable Machines 7(4), 383–404 (2007)

MacKerell Jr., A.D.: Empirical force fields for biological macromolecules: Overview and issues. Journal of Computational Chemistry 25(13), 1584–1604 (2004)

Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press, Cambridge (1996)

Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J.: Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. Journal of Computational Chemistry 19(14), 1639–1662 (1998)

Nakayama, H.: Aspiration level approach to interactive multi-objective programming and its applications. In: Pardalos, P.M., Siskos, Y., Zopounidis, C. (eds.) Advances in Multicriteria Analysis, pp. 147–174. Kluwer Academic Publishers, Dordrecht (1995)

Nakayama, H., Sawaragi, Y.: Satisficing trade-off method for interactive multiobjective programming methods. In: Grauer, M., Wierzbicki, A.P. (eds.) Interactive Decision Analysis – Proceedings of an International Workshop on Interactive Decision Analysis and Interpretative Computer Intelligence, pp. 113–122. Springer, Heidelberg (1984)

Nakayama, H., Kaneshige, S., Takemoto, S., Watada, Y.: An application of a multiobjective programming technique to construction accuracy control of cable-stayed bridges. European Journal of Operational Research 87, 731–738 (1995)

Obayashi, S., Sasaki, D., Takaguchi, Y., Hirose, N.: Multi-objective evolutionary computation for supersonic wing-shape optimization. IEEE Transactions on Evolutionary Computation 4(2), 182–187 (2000)

Okabe, T., Foli, K., Olhofer, M., Jin, Y., Sendhoff, B.: Comparative Studies on Micro Heat Exchanger Optimisation. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC-2003), pp. 647–654. IEEE Computer Society Press, Los Alamitos (2003)

Olhofer, M., Arima, T., Sonoda, T., Sendhoff, B.: Optimization of a stator blade used in a transonic compressor cascade with evolution strategies. In: Parmee, I. (ed.) Adaptive Computing in Design and Manufacture, pp. 45–54. Springer, Heidelberg (2000)

Olhofer, M., Jin, Y., Sendhoff, B.: Adaptive encoding for aerodynamic shape optimization using evolution strategies. In: Congress on Evolutionary Computation (CEC), Seoul, Korea, May 2001, vol. 2, pp. 576–583. IEEE Computer Society Press, Los Alamitos (2001)

Ong, Y.-S., Nair, P.B., Lim, K.Y.: Max-min surrogate-assisted evolutionary algorithms for robust design. IEEE Transactions on Evolutionary Computation 10(4), 392–404 (2006)

Paenke, I., Branke, J., Jin, Y.: Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. IEEE Transactions on Evolutionary Computation 10, 405–420 (2006)

Pettersson, F., Chakraborti, N., Saxén, H.: A genetic algorithms based multiobjective neural net applied to noisy blast furnace data. Applied Soft Computing 7, 387–397 (2007a)

Pettersson, F., Chakraborti, N., Singh, S.B.: Neural networks analysis of steel plate processing augmented by multi-objective genetic algorithms. Steel Research International 78, 890–898 (2007b)

Poloni, C., Pediroda, V.: GA coupled with computationally expensive simulations: tools to improve efficiency. In: Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, pp. 267–288. John Wiley and Sons, Chichester (1997)

Price, K., Storn, R.N., Lampinen, J.A. (eds.): Differential Evolution: A Practical Approach to Global Optimizations. Springer, Berlin (2005)

Saxén, H., Pettersson, F., Gunturu, K.: Evolving nonlinear time-series models of the hot metal silicon content in the blast furnace. Materials and Manufacturing Processes 22, 577–584 (2007)

Shao, L.: A survey of beam intensity optimization in IMRT. In: Halliburton, T. (ed.) Proceedings of the 40th Annual Conference of the Operational Research Society of New Zealand, Wellington, 2-3 December 2005, pp. 255–264 (2005), Available online at `http://secure.orsnz.org.nz/conf40/content/paper/Shao.pdf`

Shao, L., Ehrgott, M.: Finding representative nondominated points in multiobjective linear programming. In: IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making, pp. 245–252. IEEE Computer Society Press, Los Alamitos (2007)

Shao, L., Ehrgott, M.: Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. Mathematical Methods of Operations Research (2008)

Stewart, T.J., Janssen, R., van Herwijnen, M.: A genetic algorithm approach to multiobjective land use planning. Computers and Operations Research 32, 2293–2313 (2004)

Takagi, H.: Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation. Proceedings of the IEEE 89, 1275–1296 (2001)

The MathWorks Inc. (2008)

Tsutsui, S., Ghosh, A.: Genetic algorithms with a robust solution searching scheme. IEEE Transactions on Evolutionary Computation 1(3), 201–208 (1997)

Wierzbicki, A.P.: Reference point approaches. In: Gal, T., Stewart, T.J., Hanne, T. (eds.) Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications, Kluwer Academic Publishers, Boston (1999)

Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A regularity model-based multi-objective estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 12(1), 41–63 (2008)