EngOpt 2008 - International Conference on Engineering Optimization
Rio de Janeiro, Brazil, 01 - 05 June 2008.

Navigation on a Pareto-optimal front utilizing gradient information in interactive multiobjective optimization

Henri Ruotsalainen, Elina Madetoja and Jari Hämäläinen

Department of Physics, University of Kuopio, P.O. Box 1627, FI-70211 Kuopio, Finland.
Henri.Ruotsalainen@uku.fi, Elina.Madetoja@uku.fi, Jari.Hamalainen@uku.fi

1. Abstract
In this paper, utilizing gradient (i.e. derivative) information to navigate through multiobjective optimization solutions is studied. Targets in multiobjective optimization are conflicting, and thus it is impossible to satisfy all of them at the same time. Therefore, new ways to assist the solution process and navigation through a Pareto-optimal front in multiobjective optimization are needed. When using gradient-based multiobjective optimization methods, gradients are needed to be calculated for the optimization problem solver. However, existing gradient information can be utilized thorougher: we present two different ways to control and direct the optimization process interactively by utilizing gradient information. First, gradients can be used for approximating a Pareto-optimal front, and secondly, calculating trade-offs between the optimization targets. Results from the numerical examples indicate that these approaches enable more efficient search for the best possible solution because the approaches predict how the conflicting objectives will behave and they indicate what changes to the current solution are productive to make during the optimization process. This decreases the number of uninteresting solutions calculated, and better solutions can be obtained by finding advantageous trade-offs.
2. Keywords: Nonlinear multiobjective optimization, Interactive methods, Gradient information, Trade-off information, Decision Support.

3. Introduction
In multiobjective optimization, there are several conflicting targets that cannot reach their optima simultaneously. Thus, finding the optimal solution is not trivial, and a comparison and decision making between different compromise solutions, so-called Pareto-optimal solutions (forming a Pareto-optimal front), is not an easy task for the human mind. Therefore, new ways to support decision making are needed.

In this paper, gradient (i.e. derivative) information is used to assist solution processes and navigation through a Pareto-optimal front in multiobjective optimization. When using gradient-based optimization solvers (with scalarized multiobjective optimization problems), gradients are needed to be calculated some way. However, existing gradient information can be utilized more carefully to assist the decision making process. First, a computationally efficient meta-model for approximating a Pareto-optimal front can be produced using the gradient information. With the approximated Pareto-optimal front, behavior of the Pareto-optimal solutions can be predicted in a certain area, which can be useful information in decision making. Similar ideas of approximating Pareto-optimal fronts have been presented also in the literature [1, 2, 3, 4].

Second, gradient information can be employed during an interactive solution process to generate so-called trade-off information. Here, a decision maker (a person who is supposed to have better insight into the problem considered) utilizes the trade-off information to predict the most profitable direction where to look for the best Pareto-optimal solution. That is, when one target is getting better, how to maintain feasible levels of other targets. The concept of trade-off is used in the context of multiobjective optimization because the Pareto-optimal solutions of the conflicting targets are mathematically incomparable and one has to sacrifice in some objective in order to gain in some other objective, and this is called trading-off. In other words, this kind of trade-off information describes interdependencies between objective functions and how their values change locally with respect to others. Furthermore, we want to present the trade-off information to the decision maker as clearly as possible so that interpreting and utilizing it becomes easier and the desired solution is easy to choose. The concept of trade-off have been widely discussed in the literature, see e.g. [5, 6, 7, 8, 9, 10, 11, 12, 13], but here we concentrate on supporting the decision making process with the trade-off information such as in [14].

Results from the numerical examples indicate that presenting an approximation of a Pareto-optimal front and the trade-off information to the decision maker makes the optimization process more intuitive. Since decision makers are able to predict how the solutions may behave, directing the solution process comes more efficient and he/she can learn about the interrelationships between the objectives. Thus, the number of uninteresting solutions computed decreases. The support provided by gradient information is extremely welcome especially when the evaluation of objectives requires solving of computationally costly mathematical simulation models. That is the case in real world industrial problems, see e.g. [15], in which it is important to reduce the number of trial-and-error experiments to be made.

Rest of the paper is organized as follows: Next, there is a section in which multiobjective optimization is presented. After that, the utilization of gradient information is explained. In this section, both, an approximation of a Pareto-optimal front and the use of trade-off information are described. Next section contains two examples: An academic example with two conflicting objectives, and a real world example having three conflicting targets. The final section is devoted for conclusions.

4. Multiobjective optimization approach

In general, a multiobjective optimization problem can be defined as follows [16]

$$\begin{array}{ll} \text{minimize} & \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})\} \\ \text{subject to} & \boldsymbol{x} \in S, \end{array} \tag{1}$$

where $\boldsymbol{x}$ is a vector of decision variables from the feasible set $S \subset \mathbf{R}^n$ defined by box, linear and nonlinear constraints. We can denote a vector of objective function values or an objective vector $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$. We denote the image of the feasible set by $\boldsymbol{f}(S) = Z$ and call it as a feasible objective set. If some objective function f_i is to be maximized, it is equivalent to consider minimization of $-f_i$.

In multiobjective optimization, optimality is understood in the sense of Pareto-optimality [16]. A decision vector $\boldsymbol{x}^* \in S$ is Pareto-optimal, if there does not exist another decision vector $\boldsymbol{x} \in S$ such that $f_i(\boldsymbol{x}) \le f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one index j. These Pareto-optimal solutions constitute a Pareto-optimal front, i.e. a Pareto-optimal set. From a mathematical point of view, all of these Pareto-optimal solutions are equally good and they can be regarded as equally valid compromise solutions of the problem considered. Because vectors cannot be ordered completely, there exists no trivial mathematical tool in order to find the most satisfactory solution in the Pareto-optimal front.

Because all the solutions are equally good, an expert of the problem (from the application field) known as a decision maker is typically needed in order to find the best or most satisfying solution to be called the final one. The decision maker can participate in the solution process and in one way or the other, determine which one of the Pareto-optimal solutions is the most desired to be the final solution. It is often useful for the decision maker to know the ranges of objective function values in the Pareto-optimal front. An ideal objective vector gives lower bounds for the objective functions in the Pareto-optimal front and it is obtained by minimizing each objective function individually subject to the constraints. A nadir objective vector giving upper bounds of objective functions in the Pareto-optimal front is usually difficult to calculate, and, thus, its values are usually only approximated, for example, by using pay-off tables, see for more [16].

Sometimes, the many methods developed for multiobjective optimization are divided into four classes according to role of the decision maker [16]. First, there are methods where no decision maker is available and where the final solution is some neutral compromise solution. The three other classes are a priori, a posteriori and interactive methods, where the decision maker participates in the solution process before it, after it or iteratively, respectively. We concentrate on the last-mentioned class in this paper. Interactive methods make possible the decision maker to control the solution process iteratively and learn about the conflicting targets during optimization. This approach also provides shorter computing times, because the decision maker directs the solution process the way he/she wants and only such solutions he/she is interested in are generated. In this paper, by adding the potential offered by gradient information, the solution process can be made even more intuitive and easier to understand.

5. Utilization of gradient information

In this section, we discuss about two different ways how gradient information can be used in supporting the interactive decision making process. First, we approximate a Pareto-optimal front with help of Taylor's polynomial in which gradient information is needed. Second, we discuss how trade-off information, which has been computed with help of gradients, can be utilized when navigating through a Pareto-optimal front during an interactive optimization process.

5.1. Approximating a Pareto-optimal front using gradient information

When using gradient-based optimization solvers (with scalarized multiobjective optimization problems), gradient information is needed to be calculated. In addition, it can be used in different ways to assist the decision maker. With the gradient information we are able to produce a computationally efficient meta-model for approximating a Pareto-optimal front. With the approximation, behavior of the Pareto-optimal solutions can be predicted in a certain area, which can be useful information in decision making.

In this study, we use Taylor's formula to find a successive approximation to a Pareto-optimal front. Taylor's formula expresses a function as an infinite sum of terms calculated from the values of its derivatives at a single point. Taylor's formula (a polynomial approximation and an error (remainder) term) can be written as

$$\boldsymbol{P}(\boldsymbol{x}) = \sum_{l=0}^{p} \frac{1}{l!} d^l \boldsymbol{f}(\boldsymbol{a}; \boldsymbol{x} - \boldsymbol{a}) + \frac{1}{(p+1)!} d^{p+1} \boldsymbol{f}(\xi; \boldsymbol{x} - \boldsymbol{a}), \tag{2}$$

where $p \in \mathrm{N}$ and d^l denotes l order of derivatives. In Taylor's formula, the first term (expressed as a sum) gives a polynomial approximation of function $\boldsymbol{f}$ in the neighborhood of point $\boldsymbol{a}$, and its accuracy can be estimated with the second term even if the point ξ is unknown.

In Eq.(2), polynomial approximation of function $\boldsymbol{f}$

$$\boldsymbol{P}(\boldsymbol{x}) = \sum_{l=0}^{p} \frac{1}{l!} d^l \boldsymbol{f}(\boldsymbol{a}; \boldsymbol{x} - \boldsymbol{a}), \quad \boldsymbol{x} \in \mathrm{R}^n \tag{3}$$

is a Taylor's polynomial of a function $\boldsymbol{f}$ in point $\boldsymbol{x}$. Using Eq.(3), an approximation of a Pareto-optimal front can be formed (see, e.g. Figure 1). In the Figure 1, there are different order ($p = 0, \ldots, 3$) polynomial approximations of a Pareto-optimal front based on Taylor's polynomial. Polynomial $\boldsymbol{P}(\boldsymbol{x})$ describes a Pareto-optimal front better than other polynomial of degree p.

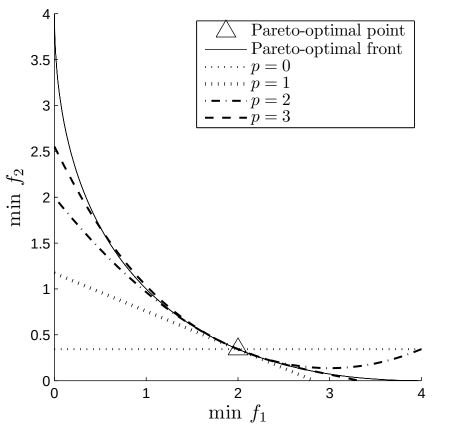

Figure 1: Approximations of a Pareto-optimal front based on Taylor's polynomial for a multiobjective optimization problem formulated in [17].

If we have derivative information of degree p, we can find a good Taylor's polynomial approximation to a Pareto-optimal front. Nevertheless, in practice (in the real world problems), we usually have only first order of derivatives (p=1), and thus the approximation is not accurate but it gives information of local changes. Then, the Taylor's polynomial is of the form

$$\boldsymbol{P}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{a}) + d^1\boldsymbol{f}(\boldsymbol{a}; \boldsymbol{x} - \boldsymbol{a}). \tag{4}$$

Based on the presented first order linear Taylor's polynomial approximation (4), new points (approximating Pareto-optimal points) can be generated to the neighborhood of the Pareto-optimal point without solving the original problem. *The main idea is that a nonlinear approximation of the Pareto-optimal front between the Pareto-optimal points can be done based on the Pareto-optimal points and the approximated Pareto-optimal points by using e.g. the least square method.* In a two-objective case, at least two Pareto-optimal points and two approximated Pareto-optimal points between these points are needed, and, in three dimensions, at least three Pareto-optimal points and six approximated points between these points are needed. Thus, an algorithm can be written:
Algorithm 1: *Let us assume that we have k Pareto-optimal solutions forming a set denoted by PP. Let AP be a set of approximated Pareto-optimal points.*

1. *Set* $AP = \phi$.
2. *For each* $f_i \in PP$.
   (a) *Select k - 1 nearest points from PP* $\setminus\{f_i\}$ *and denote them by* $\widehat{f}_j \in \widehat{PP}_i,\ j = 1, \ldots, k-1$.
   (b) *Put also* f_i *in* $\widehat{PP}_i$ *and check that a similar set does not exist* $\widehat{PP}_k,\ k = 1, \ldots, i-1$. *If there is a similar set, then go to 1.*
   (c) *Calculate using Taylor's polynomial approximation new points as follows: For each* f_j *in* $\widehat{PP}_i$
      i. *calculate k - 1 new approximated Pareto-optimal points (one toward each point in* $\widehat{PP}_i$ *except* f_j*), and*
      ii. *put the points in AP.*
   (d) *Fit a proper approximation for a Pareto-optimal front using* $\widehat{PP}_i \cup AP$.
3. *Show the approximations obtained and ask if more accurate approximation of a Pareto-optimal front is needed. If not, then exit.*
4. *Calculate a new Pareto-optimal solution with a multiobjective optimization solver and put it in PP. Continue from 1.*

Two examples of using this algorithm are presented in the Results section.

For nonlinear problems, first order Taylor's polynomial is accurate approximation only in some finite neighborhood of the current solution. In that sense, a fitted approximation of the Pareto-optimal front based on only a few Pareto-optimal points (and a few approximated points) can be even misleading in some cases; if the Pareto-optimal points are not closely-spaced, the approximation between these points can be inaccurate. However, a skillful decision maker can obtain more information from the approximation than only from the Pareto-optimal points, and naturally the piece-wise fitted nonlinear approximation of the Pareto-optimal front comes more accurate when the number of Pareto-optimal points calculated increases. In addition, when using this approach, the Pareto-optimal front should be continuous.

5.2. Navigating through a Pareto-optimal front using trade-off information

Trade-off information has been traditionally used as a part of a multiobjective optimization method. In our approach, we concentrate on utilizing the trade-off information as an aid which supports decisions, i.e. our aim is to produce additional value to the used multiobjective optimization method which makes the decision making easier. Trade-off information idea is possible to use in all classification-based methods and reference point methods [14] (interactive methods), where preference information given by the decision maker is needed in specifying a reference point or classifying the objective functions in different classes. Our target is to support the decision maker in the selection of the next reference point or making the next classification. Consequence of this, the number of iterations needed is reduced. Thus, the whole interactive solution process can be shortened which saves time of the decision maker and reduces the number of Pareto-optimal solutions needed to be calculated.

Here, the concept of trade-off is presented based on [5]. The ratio of change between points $\boldsymbol{x}$ and $\boldsymbol{x}^*$ involving objective functions f_i and f_j can be defined by

$$T_{ij}(\boldsymbol{x}, \boldsymbol{x}^*) = \frac{f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}^*)}{f_j(\boldsymbol{x}) - f_j(\boldsymbol{x}^*)}, \quad \boldsymbol{x},\ \boldsymbol{x}^* \in S, \tag{5}$$

where $f_j(\boldsymbol{x}) \neq f_j(\boldsymbol{x}^*)$. If $f_l(\boldsymbol{x}) = f_l(\boldsymbol{x}^*)$ for all $l \neq i, j$, is T_{ij} called partial trade-off between vectors $\boldsymbol{x}$ and $\boldsymbol{x}^*$. If $f_l(\boldsymbol{x}) \neq f_l(\boldsymbol{x}^*)$ for at least one $l \neq i, j$, then T_{ij} is called total trade-off.

Using the ratio of change $T_{ij}(\boldsymbol{x}, \boldsymbol{x}^*)$ we can define total trade-off rate at the point $\boldsymbol{x} \in \mathbf{R}^n$ to direction $\boldsymbol{d}$ as a limit

$$T_{ij}(\boldsymbol{x}, \boldsymbol{d}) = \lim_{\alpha \to 0} T_{i,j}(\boldsymbol{x} + \alpha \boldsymbol{d}, \boldsymbol{x}), \tag{6}$$

where $\boldsymbol{d} \neq 0$ is a feasible direction. That is, there exists $\alpha_0 > 0$ such that $\boldsymbol{x} + \alpha\boldsymbol{d} \in S$ for all $\alpha \in [0, \alpha_0]$. If $\boldsymbol{d}$ is a feasible direction such that there exists $\overline{\alpha} > 0$ satisfying $f_l(\boldsymbol{x} + \alpha\boldsymbol{d}) = f_l(\boldsymbol{x})$ for all $l \neq i, j$ and for all $0 \leq \alpha < \overline{\alpha}$, then the corresponding t_{ij} is called a partial trade-off rate. In continuously differentiable case, the total and partial trade-off rates can be formulated [16]

$$t_{ij}(\boldsymbol{x}, \boldsymbol{d}) = \frac{\nabla f_i(\boldsymbol{x})^T \boldsymbol{d}}{\nabla f_j(\boldsymbol{x})^T \boldsymbol{d}} \tag{7}$$

and

$$t_{ij}(\boldsymbol{x}) = \frac{\partial f_i(\boldsymbol{x})}{\partial f_j}, \tag{8}$$

respectively, where $\nabla f_j(\boldsymbol{x})^T \boldsymbol{d} \neq 0$ and $\partial f_j \neq 0$.

Figure 2 shows the concept of trade-off in the two objective function case in objective space. In the Figure 2, the set $Z = \boldsymbol{f}(S)$ denotes the image of the feasible set and its thicker boundary indicates the Pareto-optimal objective vectors set $z \in \boldsymbol{f}(E)$, that is, the Pareto-optimal front. The rate-off rate related to the objective vector $\boldsymbol{z}$ is shown by an arrow. This means that if we want to improve objective f_2 by amount Δf_2 we can approximate the impairment in objective f_1 by Δf_1. As it can be seen in Figure 2, the rate-off rate in some point is only a linear approximation and, thus, can only be used in some finite neighborhood of the point considered.

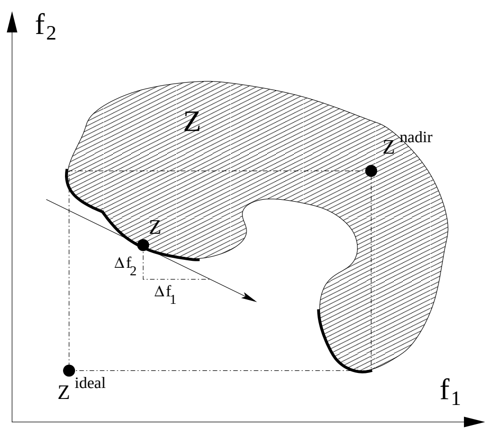

Figure 2: An illustrated trade-off for two objectives.

In this paper, by trade-off information we mean trade-off rates. In addition, we are interested in trade-off rates only at the Pareto-optimal points. Thus, in above presented trate-off definitions, a feasible set S is replaced by a Pareto-optimal set E. At every point $\boldsymbol{x} \in E$ we can formulate the trade-off rate matrix

$$M(\boldsymbol{x}, \boldsymbol{d}) = \begin{bmatrix} t_{11}(\boldsymbol{x}, \boldsymbol{d}) & \dots & t_{k1}(\boldsymbol{x}, \boldsymbol{d}) \\ \vdots & \ddots & \vdots \\ t_{1k}(\boldsymbol{x}, \boldsymbol{d}) & \dots & t_{kk}(\boldsymbol{x}, \boldsymbol{d}) \end{bmatrix}. \tag{9}$$

The trade-off matrix (9) reflects sensitivities between objectives when we are moving to the direction $\boldsymbol{d}$ from the point $\boldsymbol{x}$. A similar matrix $M(\boldsymbol{x})$ can be defined also for partial trade-off rates. With partial trade-off, we can study how one objective impairs if we want to improve one other objective by one unit, and other objectives stay unchanged at the same time. In the case of total trade-off, we can consider how all other objectives change if we improve one objective by one unit. When we have only two objectives, are the total and partial trade-off rates naturally the same. Computing the trade-off rates is presented in [14], for example.

Two different ways to present the trade-off information to the decision maker are suggested in this paper. A straightforward approach is to show the numerical trade-off rate information from the trade-off matrix (9). Another way to present the trade-off rates to the decision maker is to make some simplifications to the numerical trade-off rate information presented. That is because sometimes it might be enough for the decision maker just to know whether the trade-off between objectives is below, equal or above neutral rate of change. In such a case, a so-called arrow matrix visualization can be used [14], which is demonstrated in the next section.

For nonlinear problems such as in this paper, the trade-off rate matrix values are often enough accurate approximations only in some finite neighborhood of the current solution. The proper neighborhood is problem specific and sometimes difficult to characterize. That is why the decision maker have to be a skillful expert on his/her field to use the trade-off information.

6. Results

Similar ideas for presenting information and supporting the decision making process have been used only with a reference point method [14]. Therefore, we tested these ideas with a classification based method. In these examples (one academic and one real world), a classification-based interactive multiobjective optimization method NIMBUS [18] was used.

6.1. Example 1: An academic example

Here, we used an example presented in [17] in which there were two conflicting objective functions ($f_1 : \mathbf{R} \to \mathbf{R}$ and $f_2 : \mathbf{R} \to \mathbf{R}$) to be minimized at the same time:

$$f_1(x) = x^2 \tag{10}$$

and

$$f_2(x) = (x - 2)^2. \tag{11}$$

Thus, the problem is as follows

$$\begin{array}{ll} \text{minimize} & \{f_1(x), f_2(x)\} \\ \text{subject to} & -10^5 \le x \le 10^5. \end{array} \tag{12}$$

In this case, we wanted to present useful information to the decision maker as much as possible to support he/she in his/hers decisions with a minimal computational effort. Using the interactive classification-based optimization method, after every classification a new Pareto-optimal solution was computed and gradient information could be utilized.

Using the Algorithm 1 presented in the previous section, an approximation of a Pareto-optimal front was generated during the iterative solution process. First, a decision maker calculated two Pareto-optimal solutions with the interactive multiobjective optimization method. He wanted to obtain two different kinds of solutions, and thus he made two different classifications: he firstly preferred objective function f_1 and secondly objective function f_2. Two different kind of solutions were obtained (Figure 3 on the upper-left). Using Taylor's polynomial with first order derivatives, a linear approximation of the Pareto-optimal front in the neighborhood of the Pareto-optimal points was made. With this approximation,

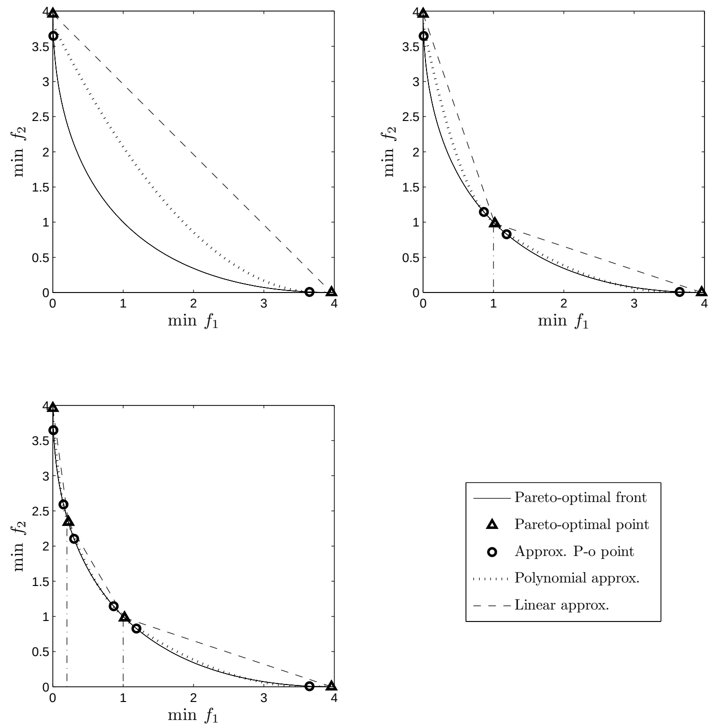

Figure 3: A piece-wise nonlinear approximation of a Pareto-optimal front based on gradient information for two objectives.

two approximating Pareto-optimal points were calculated between the real Pareto-optimal points. Now, the decision maker had four points (two real Pareto-optimal points and two approximated), and a nonlinear second order polynomial approximation of the Pareto-optimal front could be made to these four points using the least square method (Figure 3 on the upper-left). For comparison, there is a linear approximation between these two Pareto-optimal points presented in the same figure. The optimization process was iterative: now the decision maker could calculate a new Pareto-optimal point between two existing points, and a new Taylor's polynomial approximation of the Pareto-optimal front in the neighborhood of this new point could be done. From this Taylor's polynomial approximation, new Pareto-optimal points were approximated. Now, the decision maker was able to generate two separate second order polynomial approximations of the Pareto-optimal fronts using the least square method in both sides of the new Pareto-optimal point (Figure 3 on the upper-right). In this way, he obtained a piece-wise nonlinear approximation of the Pareto-optimal front which accuracy naturally increased when the number on Pareto-optimal points increased (Figure 3 on the lower-left, with four Pareto-optimal points). In this way, by following the Algorithm 1, the Pareto-optimal points and their gradient information were utilized in forming a good approximation of the Pareto-optimal front with computing less Pareto-optimal points with the computationally costly real model (which is the case in real world industrial problems).

In addition, based on the gradients calculated, the trade-off matrix was formed. In Table 1, there is an example of the trade-off information obtained in the Pareto-optimal point presented in the Figure 1. This information was shown to the decision maker after every iteration (i.e. in each Pareto-optimal point). To avoid information burden to the decision maker, the information presented was reduced by using an arrow matrix visualization as discussed in the previous section. In Figure 4, an example of

Table 1: Matrix of total trade-off rates in the Pareto-optimal point presented in the Figure 1.

	f_1	f_2
f_1	1	-0.414
f_2	-2.414	1

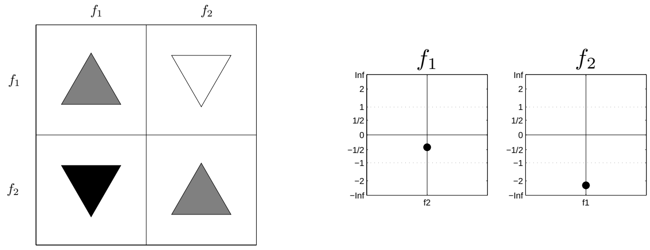

Figure 4: Visual rate of change arrows and visual compromise bars for two objectives.

arrow matrix visualization is presented. In this example, threshold values used in rate of change arrows were the following: White color represented small change in objective values (rate of change was from -0.5 to 0.5), and light gray meant neutral change (rate of change was from -2 to -0.5 or from 0.5 to 2). In addition, significant change was expressed with black color (rate of change less than -2 or more than 2). If the arrow points up, the objective improves, and if the arrow points down, the objective impairs. In this particular Pareto-optimal point, it can be seen that if both objectives are equally important to the decision maker, it could be profitable to try to improve f_1 because f_2 impairs only a little at the same time. Instead, if the decision maker tries to improve f_2, will f_1 worsen significantly.

With this information presented to the decision maker, the optimization process could be made more efficient and the decision maker could better understand the interrelationships between targets without numerical burden. Therefore, less Pareto-optimal solution needed to be computed, and thus solving the optimization problem came easier and faster.

6.2. Example 2: A real world example

In this example, a radiotherapy treatment planning case is studied. There are three objective functions:

$$f_1(\boldsymbol{x}) = ||D_{PTV} - D(\tilde{\boldsymbol{x}})||_{L_\infty(PTV)}, \tag{13}$$

$$f_2(\boldsymbol{x}) = ||D(\tilde{\boldsymbol{x}})||_{L_2(OAR)} \tag{14}$$

and

$$f_3(\boldsymbol{x}) = ||D(\tilde{\boldsymbol{x}})||_{L_2(NT)}, \tag{15}$$

where L_∞ is L_∞-norm and L_2 is L_2-norm. The objective function f_1 describes the maximum dose deviation from a desired dose D_{PTV} in the tumor (PTV) and we want to minimize it. The objective functions f_2 and f_3 are the averaged doses in the organ at risk (OAR) and in normal tissue (NT), respectively, to be minimized, too. The desired dose D_{PTV} was scaled to 100%. As one can easily understand, the targets are conflicting, so all the treatment planning targets cannot reach their minima at the same time. Achieving the desired dose D_{PTV} in the PTV is not possible without affecting some other regions because the radiation must travel through NT to reach the PTV, for example (see, Figure 7. In Figure 7, dark grey area is the PTV, light grey area is the OAR, and these areas are surronded by NT (white). Therefore, the problem is as follows

$$\begin{array}{ll} \text{minimize} & \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})\} \\ \text{subject to} & 0 \leq \boldsymbol{x} \leq 1. \end{array} \tag{16}$$

In this example, the Algorithm 1 was used to generate an approximation of a Pareto-optimal front such as in the first example. By following the Algorithm 1, Pareto-optimal points and their gradient information was utilized in forming a nonlinear three dimensional approximation of the Pareto-optimal front with computing less Pareto-optimal points with the computationally costly real model. By studying the approximation (Figure 5), the decision maker could obtain a clear picture on his mind how the

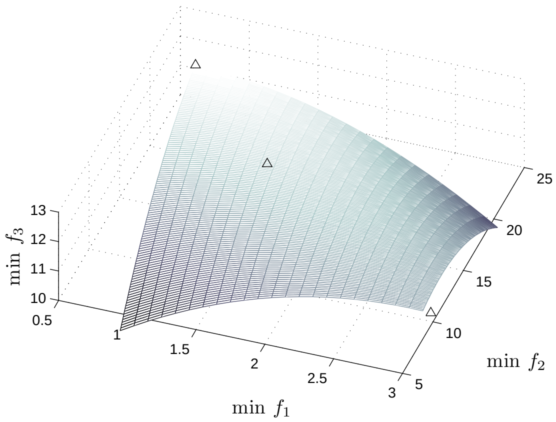

Figure 5: A nonlinear three dimensional approximation of a Pareto-optimal front based on gradient information. Three Pareto-optimal points calculated are marked with △.

conflicting objectives would behave, and what are the interesting regions of the Pareto-optimal front. However, in Figure 5, there could be seen that the approximation is accurate only in some finite neighborhood of the Pareto-optimal solutions: the farthermost parts of the approximated Pareto-optimal front represent unrealistic good solutions (in lower-left corner, for example). Thus, the decision maker should be a skillful expert from his/hers field to interpret the approximated Pareto-optimal front.

In addition to the approximated Pareto-optimal front, trade-off information was shown to the decision maker during the interactive optimization process after every iteration. For example, in Figure 6 (and in Table 2), there is presented the total trade-off information after one iteration. The total trade-off was

Table 2: Matrix of total trade-off rates.

	f_1	f_2	f_3
f_1	1	-5.161	-1.999
f_2	-0.093	1	0.211
f_3	-0.286	1.673	1

used because the decision maker felt that it described better his preferences than partial trade-off: he wanted to see what happens to the other objectives if he improves one of the objectives by one unit. As can be seen in the figure, it would be profitable to try to improve objective f_3. If objective f_3 was improved by one unit, the objective f_2 would also improve and the objective f_1 would impair only a little, for example. For comparison, if he wanted to improve the objective f_1 by one unit, would both f_2 and f_3 impair a lot at the same time. Threshold values used in the rate of change matrix arrows were the same than in Example 1.

With this information presented to the decision maker, he was able to steer the solution process in an efficient way and better treatment plans could be obtained by finding advantageous trade-offs. In addition, less Pareto-optimal solutions was needed to be computed because the decision maker had a clear picture on his mind of the conflicting targets and their behavior and trade-offs. That is why solving the optimization problem came easier and faster. In Figure 7, there is the final solution (treatment plan)

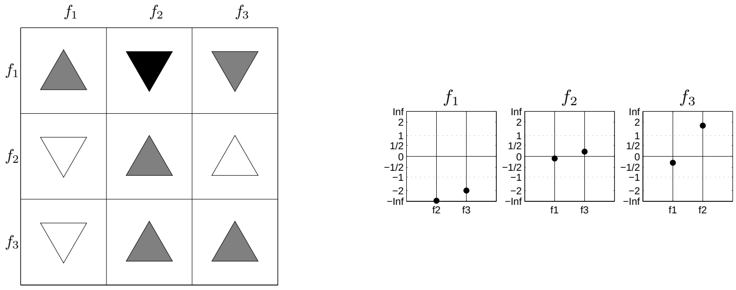

Figure 6: Visual rate of change arrows and visual compromise bars for three conflicting objectives.

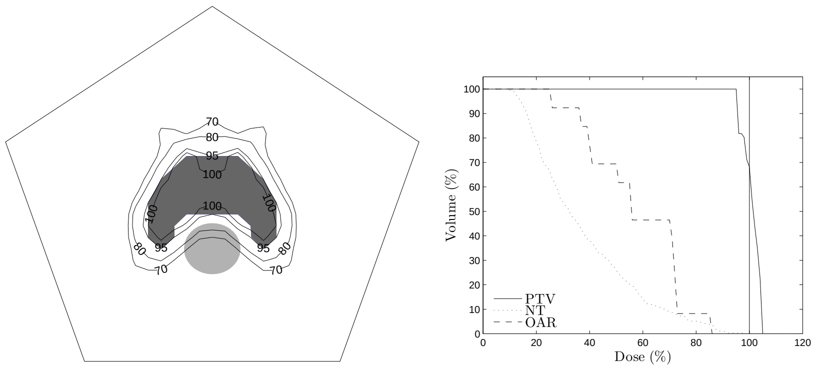

Figure 7: Optimized radiotherapy treatment plan. Left, isodose map and right, dose volume histogram.

presented.

7. Conclusions

In multiobjective optimization, there are several conflicting targets that cannot reach their optima at the same time. Thus, finding the optimal solution is not trivial, and decision making between different compromise solutions is not an easy task. Hence, new tools to support the decision making are needed.

In this paper, gradient information is used to assist the navigation though a Pareto-optimal front in multiobjective optimization. When using gradient-based optimization solvers (with scalarized multiobjective optimization problems), gradients are needed to be calculated. In addition, this gradient information can be used in different ways to assist the decision maker. First, with the gradient information, we are able to produce a computationally efficient meta-model for approximating a Pareto-optimal front. With this model, behavior of the Pareto-optimal front can be predicted in a certain area. Second, gradient information can be used to generate trade-off information. Here, the point is that a decision maker utilizes the trade-off information to predict the most profitable direction where to steer the optimization process between the conflicting targets.

Presenting the gradient information to the decision maker in different ways makes the optimization process more intuitive, and it decreases the number of uninteresting solutions computed because decision makers are able to predict how the solutions may behave and thus, direct to solution process more efficiently and learn about the interrelationships between the objectives. This kind of aid is extremely welcome especially when simulation models used are computationally costly which is the case usually in real world industrial optimization problems.

8. Acknowledgments

The authors want to thank Jussi Hakanen, Ph.D., Petri Eskelinen, M.Sc., and Prof. Kaisa Miettinen.

9. References

[1] R.V. Tappeta and J.E. Renaud, Interactive Multiobjective Optimization Procedure, *American Institute of Aeronautics and Astronautics Journal*, 1999, 37, 881-889.

[2] R.V. Tappeta, J.E. Renaud, A. Messac and G.J. Sundararaj, Interactive Physical Programming: Tradeoff Analysis and Decision making in Multidisciplinary Optimization, *American Institute of Aeronautics and Astronautics Journal*, 2000, 38, 917-926.

[3] R.V. Tappeta, J.E. Renaud and J.F. Rodríguez, An Interactive Multiobjective Optimization Design Strategy for Decision Based Multidisciplinary Design, *Engineering Optimization*, 2002, 34, 523-544.

[4] K. Klamroth and K. Miettinen, Integrating Approximation and Interactive Decision Making in Multicriteria Optimization, *Operations Research*, 2008, 56, 222-234.

[5] Y.Y. Haimes and V. Chankong, Kuhn-Tucker Multipliers as Trade-Offs in Multiobjective Decision-Making Analysis, *Automatica*, 1979, 15, 59-72.

[6] H. Yano and M. Sakawa, Trade-Off Rates in the Weighted Tchebycheff Norm Method, *Large Scale Systems*, 1987, 13, 167-177.

[7] H. Nakayama, Trade-Off Analysis Using Parametric Optimization Techniques, *European Journal of Operational Research*, 1992, 60, 87-98.

[8] I. Kaliszewski and W. Michalowski, Generation of Outcomes with Selectively Bounded Trade-Offs, *Foundations of Computing and Decision Sciences*, 1995, 20, 113-122.

[9] I. Kaliszewski and W. Michalowski, Efficient Solutions and Bounds on Tradeoffs, *Journal of Optimization Theory and Applications*, 1997, 94, 381-394.

[10] H. Kuk, T. Tanino and M. Tanaka, Trade-Off Analysis for Vector Optimization problems via Scalarization, *Journal of Information & Optimization Sciences*, 1997, 18, 75-87.

[11] M.I. Henig and J.T. Buchanan, Tradeoff Directions in Multiobjective Optimization Problems, *Mathematical Programming*, 1997, 78, 357-374.

[12] J.-B. Yang and D. Li, Normal Vector Identification and Interactive Tradeoff Analysis Using Minimax Formulation in Multiobjective Optimization, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 2002, 32, 305-319.

[13] Eskelinen P. Objective trade-off rate information in interactive multiobjective optimization methods – A survey of theory and applications. *Working Papers W-445*, 2008, Helsinki School of Economics, Helsinki, Available at http://hsepubl.lib.hse.fi/pdf/wp/w445.pdf

[14] Hakanen J and Eskelinen P. Ideas of using trade-off information in supporting the decision maker in reference point based interactive multiobjective optimization. *Working Papers W-405*, 2006, Helsinki School of Economics, Helsinki, Available at http://hsepubl.lib.hse.fi/pdf/wp/w405.pdf

[15] Hämäläinen J, Hämäläinen T, Madetoja E and Ruotsalainen H. CFD-based optimization for a complete industrial process: papermaking. *Optimization and computational fluid dynamics, Eds. Thévenin, D. and Janiga, G., Springer*, 2008

[16] Miettinen K. Nonlinear multiobjective optimization. *Boston: Kluwer*, 1999

[17] Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms. *Proc. of the First International Conference on Genetic Algorithms (ICGA), Ed. Grefensttete J J*, 1987, Hillsdale, NJ, 93-100

[18] K. Miettinen and M.M. Mäkelä, Synchronous Approach in Interactive Multiobjective Optimization, *European Journal of Operational Research*, 2006, 170, 909-922.