Chapter 4. Computational Algorithms of the FGM

Computational algorithms of the FGM are introduced in this Chapter in a simplified way. In contrast to the previous chapters of the book, certain mathematical background is required in this Chapter from a reader. However, complicated mathematical issues of the FGM are not considered here. The mathematical basis of the FGM, which is provided by results that belong to the fields of differential geometry, algebraic topology and the theory of functional spaces, is described in the book (Lotov et al., 1999b).

Let us repeat in short the mathematical statement of the method that has been given already in Section 1.5. We consider a finite number of decision variables x that belong to linear space $\boldsymbol{R^n}$. The variety of feasible decision vectors is denoted by X. The criterion vectors y are elements of linear space $\boldsymbol{R^m}$. Criterion vectors y are related to decision vectors x by a given mapping

$$f : R^n \to R^m .$$

Then, the variety of feasible criterion vectors Y is given by

$$Y = \{ y \in R^m : y = f(x), \, x \in X \}$$

If user is interested in decreasing the criterion values, the non-dominated (efficient, Pareto-efficient, Pareto-optimal) frontier *P(Y)* of the variety Y that is defined as

$$P(Y) = \{ y \in Y : \{ y' \in Y : y' \le y, \, y' \ne y \} = \varnothing \}.$$

To visualize *P(Y)*, we approximate and visualize the variety

$$Y^* = Y + R^m_+ ,$$

where R_+^m is the non-negative cone of $\boldsymbol{R^m}$. The variety Y^* is called the EPH of the variety Y. Visualization of the variety Y^* instead of the variety Y is based on the fact that

$$P(Y^*)=P(Y).$$

The FGM/IDM technique consists in approximation of the variety Y^* using such simple figures as polytopes, balls, boxes and cones, and in its further display using collections of its two-criterion slices (cross-sections). In this Chapter, problems of approximating the varieties Y and Y^* are considered. We start with approximating the variety Y and consider three basic cases listed below. Then, we turn to approximating the variety Y^*.

In the first case, we assume that the set X is a convex polyhedral set and the mapping f is linear, i.e. the model is linear. Note that the set X may be not bounded in this case. Convolution of linear inequality systems can be proposed for constructing the variety $f(X)$ for linear models. This approach is described in Section 5.1.

In the second case, the variety $f(X)$ is supposed to be convex and bounded. In this case, polytopes are used to approximate the variety $f(X)$. The approximation methods are based on evaluation of the support function for the variety $f(X)$. These methods do not depend upon the way of estimating support function, and so they can be applied in more general decision spaces. Methods of this kind are discussed in Section 5.2.

In the third case, the variety $f(X)$ is not supposed to be convex. This feature may be attributed or to non-linear mapping f, or to non-convex variety X, or to both. In this case it is supposed that the decision space $\boldsymbol{R^n}$ has a relatively small number of dimensions and the variety $f(X)$ is bounded. Approximation methods are based on simulation of random feasible decisions. Collections of simple figures such as balls and boxes are used as approximations of *f(X)*. Methods of this kind are discussed in Section 5.3.

The methods for approximating the EPH of the variety *f(X)* are based on the ideas of the methods developed for the three above cases. The same convolution methods can be used in the first case. The sum of a polyhedral set and the non-negative cone is used in the second case, and collections of the sums of balls (boxes) and cones are used in the third case. The problems of constructing the EPH are discussed in Section 4.4.

In this chapter, we use the standard mathematical term “set” instead of “variety”.

4.1 Methods based on convolution of linear inequality systems

In this section, methods for convolution of linear inequality systems are introduced, and their application for constructing the set *f(X)* is described. As it was said earlier, it is supposed that the set X is a polyhedral set of $\boldsymbol{R}^n$, the mapping f: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}^m$ is linear, and the set *f(X)* is constructed by a polyhedral set.

It is supposed that the set X is specified by a linear inequality system. That is,

$$X = \{x \in \boldsymbol{R}^n : Hx \leq h\}, \qquad (1.1)$$

where H is a given matrix and h is a given vector. The set X may be not bounded. A linear mapping f: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}^m$ is specified by a matrix F, such that

$$f(x) = Fx. \qquad (1.2)$$

We construct the set

$$f(X) = \{ y \in \boldsymbol{R}^m : y = Fx,\ Hx \leq h \} \qquad (1.3)$$

in the form

$$Q = \{ y \in \boldsymbol{R}^m : Dy \le d\}.$$

This idea was introduced in (Lotov, 1973 and 1975a). It is based on the opportunity to construct (orthogonal) projections of polyhedral sets.

Projections of polyhedral sets

DEFINITION. Let a set $M \subset \boldsymbol{R}^p \times \boldsymbol{R}^q$ be specified. The set

$$M_w = \{ w \in \boldsymbol{R}^q : \exists v\text{: } (v,w) \in M \}$$

is known as the *(orthogonal) projection of the set M onto* $\boldsymbol{R}^q$.

Let us show how the set $f(X)$ can be represented in the form of a projection. Consider the graph Z of the mapping f: $\boldsymbol{R}^n \to \boldsymbol{R}^m$ of the type (1.2) defined on the set (1.1), i.e.

$$Z = \{ (x,y) \in \boldsymbol{R}^n \times \boldsymbol{R}^m : y = Fx,\ Hx \le h \tag{1.4}$$

Note that the set $f(X)$ defined in compliance with (1.3) may be represented as

$$f(X) = \{ y \in \boldsymbol{R}^m : \exists x \in \boldsymbol{R}^n\text{: } y = Fx,\ Hx \le h \}.$$

Therefore,

$$f(X) = \{ y \in \boldsymbol{R}^m : \exists x\text{: } (x,y) \in Z \}.$$

That is, the set $f\,(X)$ is a projection of the set Z onto $\boldsymbol{R}^m$. Because the set Z is specified by equations of the system (1.1)-(1.2), the task of constructing the set $f(X)$ reduces to constructing the projection of the polyhedral set Z onto $\boldsymbol{R}^m$. Recall that this projection must be constructed in the form of the polyhedral set Q defined above.

Note that if $m = n$ and the matrix F is of a complete rank, one can easily find the set $f(X)$ by expressing the vector x in terms of y

as $x = F^{-1}y$ and by inserting this expression into inequality system. Therefore,

$$f(X) = \{ y \in \boldsymbol{R}^m : HF^{-1} y \leq h\}.$$

That is, the vector x may be excluded from (1.4) by an inversion of the matrix F.

However, when $m < n$, which is most often the case in applied problems, it is no longer possible to find the set $f(X)$ in this simple way. The available equalities permit the elimination of only some of the coordinates of x from (1.4). In this case, the coordinates are expressed in terms of the remaining variables and can be inserted into the inequalities. To eliminate the remaining variables and, thus, to construct the desired projection of Z, one can use the methods proposed for the convolution of systems of linear inequalities by Fourier (1826). The convolution methods help to construct the projection M_w of a convex polyhedral set $M \subset \boldsymbol{R}^p \times \boldsymbol{R}^q$ in the form

$$M_w = \{ w \in \boldsymbol{R}^q \colon Dw \leq d \} \qquad (1.5)$$

To be precise they help to find a matrix D and a vector d. of the description of M_w. Since the convolution methods are not widely known, we find it necessary to describe them briefly.

Convolution of linear inequality systems

It is supposed that the convex polyhedral set M is specified as

$$M = \{ (v,w) \in \boldsymbol{R}^p \times \boldsymbol{R}^q : Av + Bw \leq c\}, \qquad (1.6)$$

where A, B are specified matrices, and c is a specified vector. It is needed to find a matrix D and a vector d of the description of its projection M_w in the form (1.5).

Fourier proposed the convolution method for eliminating the vector v from the finite system of linear inequalities (1.6) in a way that results in the constructing of the projection (1.5) of the solution set of (1.6). The method starts with the elimination of one coordinate of the vector v, i.e. with constructing the projection of the set M onto $\boldsymbol{R}^{p-1} \times \boldsymbol{R}^{q}$. Constructing of the projection is based on eliminating the first component of the vector v. Then the process continues until all components of the vector v are eliminated. So, in the framework of the Fourier convolution method, every next projection is found from the previous one by the elimination of one coordinate of the vector v.

Eliminating of one component of the vector v is carried out by summation of pairs of the inequalities. The Fourier method resembles to some degree the method proposed by Gauss for solving systems of linear equations. However, the Fourier method is a bit more sophisticated.

The Fourier method for p = 1

In the case $p = 1$, the system (1.6) may be recast as

$$a_i\, v + \langle b_i\, ,\, w\rangle \leq c_i\, ,\; i = 1, 2, \ldots , N, \qquad (1.7)$$

where a_i are numbers and b_i are vectors. We break up all inequalities into three groups, Π_+, Π_- and Π_0, in correspondence to the sign of the coefficient by the variable v. The system describing the projection includes

- all inequalities from (1.7), which have zero coefficients at the variable v (i.e. the inequalities that belong to Π_0),
- all possible linear combinations of pairs of inequalities with opposite signs of the coefficients, i.e. all inequalities

$$\langle a_j\, b_i - a_i\, b_j\, ,\, w\rangle \leq a_j\, c_i - a_i\, c_j\, , \text{ where } i \in \Pi_-\, , j \in \Pi_+.$$

When excluding a greater number of variables by the Fourier method, the procedure remains the same: on a current elimination step, the system, which is a result of elimination of the previous coordinate, is taken as the starting system.

Convolution of particular systems shows that number of inequalities resulting from the consecutive elimination of variables by the Fourier method grows extremely fast. For this reason, the array of their coefficients soon overfills the memory of any existing computer. Fortunately, there is an opportunity to improve the situation considerably. It is based on the fact that many of the resultant inequalities are redundant, i.e. they follow from other inequalities of the system. Therefore, their elimination from the system does not affect the solution set. The ability to identify and eliminate the redundant inequalities is crucial for successful application of the convolution methods.

To show how a part of the redundant inequalities can be eliminated by convolution methods, we use the results of the theory of linear inequalities, in particular, the Farkas lemma and the Ky Fan lemma.

Farkas lemma and Ky Fan lemma

Farkas lemma is a fundamental result of the theory of linear inequalities. Among other things, it provides the basis for the duality theory of linear programming. Farkas lemma has two formulations: a geometrical formulation (see, for example, Goldman and Tucker, 1956) and an algebraic one (Gale, 1960). First, we introduce several concepts needed to formulate Farkas lemma.

DEFINITIONS. By a *polyhedral cone* is understood the set

$$K = \{b: b = A\lambda,\ \lambda \geq 0\},$$

where A is a matrix composed of an arbitrary collection of vectors $\{a_1, a_2,..., a_M\}$. By the *conjugate cone* of a cone K is understood the set

$$K^+ = \{u: <u, b> \geq 0, b \in K\}.$$

In the case of a polyhedral cone K specified by a matrix A, its conjugate takes the form

$$K^+ = \{u: uA \geq 0\}.$$

In turn, the conjugate of the cone K^+ may be defined as

$$K^{++} = \{w: <u, w> \geq 0, u \in K^+\}.$$

Farkas lemma (the geometrical formulation). For a polyhedral cone K, it holds $K^{++}=K$.

Farkas lemma (the algebraic formulation). Let there be given a matrix A. For a vector b, there exists a vector $\lambda \geq 0$, such as $b = A\lambda$, if and only if for any solution of the system$uA \geq 0$ it holds $< u, b > \geq 0$.

An obvious corollary of Farkas lemma, convenient for an analysis of the convolution methods, is the Ky Fan lemma (Ky Fan, 1956). By the way, A.D.Aleksandrov (1950) seems to be the first to prove and use the lemma.

Ky Fan lemma. System $Av \leq b$ has a solution if and only if for any solution of the system

$$uA = 0, u \geq 0 \tag{1.8}$$

it holds

$$<u, b> \geq 0.$$

Let U designate a finite matrix whose rows are vectors, which jointly generate a polyhedral solution cone for the system (1.8). Then, the lemma may be restated as follows.

Ky Fan lemma (an equivalent formulation). Let U be a finite matrix whose rows are the vectors that generate the solution cone for the system (1.8). The system $Av \le b$ is compatible if and only if $Ub \ge 0$.

The Ky Fan lemma can be used to describe the projection M_w of the polyhedral set (1.6) in the following way. Consider the system

$$Av \le b, \text{ where } b = c - Bw.$$

By the definition of a projection and according to the Ky Fan lemma, it holds $w \in M_w$ if and only if

$$U(c - Bw) \ge 0.$$

Therefore,

$$M_w = \{\, w \in \boldsymbol{R}^q \colon UBw \le Uc \,\}. \tag{1.9}$$

In accordance with (1.8), the elements of the matrix U are nonnegative. Therefore, the elements of the matrix D and of the vector d can be derived upon multiplying by a positive number and summing up the rows of the original inequality system (1.6). This is what happens in the Fourier method.

Note that there exists an infinite number of matrices U that can generate the solution cone for (1.8). However, matrices with a minimal number of rows do exist. Such matrices, denoted by U_f, are called the fundamental matrices of solutions for (1.8). Examples show that, for $p > 1$, the Fourier method does not usually result in such system (1.9) that can be related to a fundamental matrix of solutions, that is, such that

$$D = D_f = U_f B,\ d = d_f = U_f c.$$

Important ideas that can be used for constructing the matrix U_f were developed by (Motzkin et al., 1953), however, the ideas were suggested in a rather vague form and without a proof. Burger (1956) reformulated the ideas and supplied the necessary proof. Using these ideas, Chernikov (1965a, 1965b, 1968) invented several methods for the convolution of inequality systems, which make it possible to find D_f and d_f. The gist of these methods is as follows.

Chernikov methods

The Chernikov methods for convolution of linear inequality systems are based on the result provided in (Chernikov, 1968, Theorem 5.1). Here, we describe the same ideas in a somewhat simplified form, which relies on the generally known concept of the affine dependence of vectors.

DEFINITION. The vectors a_i, $i = 1, 2, ..., r$, are called affine-dependent if there exists a non-zero vector $\lambda \in \boldsymbol{R}^r$ such that

$$\sum_{i=1}^{r} \lambda_i a_i = 0,\ \sum_{i=1}^{r} \lambda_i = 0,\ \exists \lambda_i \neq 0,\ i = 1, 2, \dots, r.$$

To apply this definition for the description of the Chernikov convolution methods, we need the notion of extreme vectors of a cone. An extreme vector of a cone is one that cannot be represented as a linear combination with positive coefficients of two non-collinear vectors from the cone. Let us restrict with the pointed cones, i.e. the cones that do not contain subspaces of nonzero dimension. It is known that the minimal subset of vectors, which generate a pointed cone, coincides with the subset of its extreme vectors. The solution cone for (1.8) is a pointed cone because it belongs to the non-negative orthant. So, it is desirable to find the subset of its extreme vectors of the solution cone for (1.8).

The necessarily and sufficient condition for a solution of (1.8) to be an extreme vector of the solution cone is given by the following lemma formulated and proved by O.L. Chenykh.

Chernykh lemma. The vector u is an extreme vector of the solution cone for the system (1.8) if and only if the set of rows of the matrix A, corresponding to the positive coordinates of u, is affine-independent.

The proof of the lemma can be found in (Lotov et al., 1999b). The lemma shows that the extreme vectors of the solution cone for the system (1.8) can be identified by testing for affine independence those rows of the matrix $A,$ which enter into a linear combination with positive coefficients.

The problem of computing the matrix D_f and the vector d_f using the convolution methods can be solved in the following way. Note that any inequality of the system (1.6) can be associated with its number. Let us denote by I_i the set of inequalities of the original system (1.6) that have entered into a linear combination upon the formation of the i-th inequality of the system (1.5). This set of numbers is *the index* of the i-th inequality of the system (1.5).

The following rule for identifying redundant inequalities in the system (1.5) can be used: if

$$I_i \subset I_j \,, \tag{1.10}$$

then the rows of the matrix A with numbers from I_j are affine-dependent. Therefore, the j-th inequality is redundant.

The rule (1.10) can be derived from the results given in the book (Chernikov, 1968). On the other hand, it follows straight away from the Chernykh lemma. The rule (1.10) helps to derive a new method based on the Fourier convolution. The new method consists in testing the rule (1.10) at any step of elimination of a component of vector v. The new method is known as the *fundamental convolution method* (Chernikov, 1968). We do not describe the method in details. Instead, we consider another new method that is used in our software systems for constructing the projections of polyhedral sets. The method is proposed by Chernikov, too, and is named as the *reduced fundamental convolution method* (Chernikov, 1968). The method is as follows.

It is easy to show that a matrix of p columns cannot contain more than $(p+1)$ affine independent rows. Therefore, the number of elements in the index of the j-th inequality denoted by $|I_j|$ can be used for a preliminary testing of its redundancy: if

$$|\,I_j| > p+1,$$

then the j-th inequality is redundant. Hence, the condition (1.10), which requires a comparison of the indices of pairs of inequalities, should preferably be preceded by testing the number of elements in its index. This is the main idea of the reduced fundamental convolution method.

Reduced fundamental convolution method

An elimination step of the reduced fundamental convolution method can be described as follows. Suppose we have eliminated the variables v from the system (1.6) and derived a system of the type (1.5). In doing so, for each i-th inequality of the system (1.5), we retain its index I_i. Suppose that the further elimination of one more variable from the system (1.5) is required. As with the Fourier method, we break up all inequalities into three groups, Π_+, Π_- and Π_0. Now, we include all inequalities from Π_0 into the new system (index is not changed). We also include into the new system the linear combinations of pairs of inequalities, one of which is taken from Π_+, and the other, from Π_-, but, in contrast to the Fourier method, we limit ourselves to those combinations whose indices satisfy a condition defined below. Let us consider two inequalities $i \in \Pi_+$ and $j \in \Pi_-$ from the system (1.5):

$$b_i^1 w_1 + \sum_{k=2}^{q} b_i^k w_k \le c_i,$$

$$-b_j^1 w_1 + \sum_{k=2}^{q} b_j^k w_k \le c_j.$$

Here both $b^1{}_i$ and $b^1{}_j$ are positive. We include the new inequality

$$\sum_{k=2}^{q} (b_j^1 b_i^k + b_i^1 b_j^k) w_k \le b_j^1 c_i + b_i^1 c_j$$

with the index $(I_i \cup I_j)$ into the new system if and only if

1. $|\, I_i \cup I_j \,| \le p + 2$;

2. does not exist a k-th inequality of the system (1.5), such that $I_k \subseteq I_i \cup I_j$.

This completes the step of the method. Note that the first condition is given in a simplified form; actually, it is possible to make it more effective (Chernikov, 1968).

Discussion of convolution methods

Note that, although the number of resultant inequalities is much smaller in Chernikov methods than it is in the Fourier method, elimination of variables may result in an exponential growth of the number of the inequalities. This fact poses difficulties in using the convolution methods in real-life problems, even with a relatively small number of dimensions.

Let us consider a typical example. In a practical decision problem (not described here) it was necessary to eliminate seven variables from a system that contained 36 original inequalities. Two alternative variants of input data were available, which differed solely in several coefficients of the matrix *A*. With the first data set, the number of inequalities exceeded 500 after the elimination of the third variable, and the computation was terminated. With the second data set, it was possible to eliminate six variables, and the number of the inequalities did not exceed 500 until the seventh variable was eliminated. This example shows that the efficiency of the convolution methods strongly depends on elements of the matrix *A*. So it seems to be reasonable to change the elements of the matrix a bit to decrease the number of inequalities. The influence of modification of elements of the matrix *A* on the solution set of the system method has been examined in (Lotov, 1984 and 1995). The same technique was used for estimation of the influence of rounding errors on the stability of the Fourier convolution (Lotov, 1986).

The experience gained in using the convolution methods in real-life problems shows that these methods can guarantee the final result only for systems of a very small dimension (say, with 5 variables and 10 inequalities). At the same time, there are practical tasks where the convolution methods are efficient enough because the elimination of variables results in minor growth of the number of inequalities. The order in which variables are eliminated also has a strong influence on the progress of convolution. This implies that the desired result can sometimes be obtained through a change in the elimination order.

Let us stress that application of the reduced fundamental convolution method solves the problem of the redundant inequalities only partially. Even the minimal matrix U_f will usually generate redundant inequalities in the description of the set M_w. Indeed, the matrix U_f describes the edges of the solution cone for (1.8). This cone depends on the matrix *A* only. The matrix U_f has to include rows that help to describe the projection of the set *M* for any *B* and *c*. Therefore a great number of redundant constraints do arise in the process of constructing a projection for specific *B* and *c* by the method of reduced fundamental convolution.

To extend the applicability of the convolution methods, it has proved necessary to invoke methods for the identification of redundant inequalities in linear systems. A mathematical statement of the problem is as follows. In a finite system of linear inequalities

$$<a_i, x> \leq b_i,\ i = 1, 2, \dots , N,$$

where $a_i \in \boldsymbol{R}^p$ are the given coefficient vectors, and b_i are the given numbers, it is necessary to identify a subsystem that has the same solution set, but that does not contain redundant inequalities. Problem of eliminating the redundant inequalities in linear systems has been known for a long time (Karwan et al., 1983). Most methods are based either on iterations of the prime or dual problems of linear programming or on some heuristic techniques. We have also developed such methods and used them to construct sets of the type (1.3) in some practical tasks. We have found that there is no way of finding universal methods for elimination of the redundant constraints that are efficient in most problems. Some methods will identify practically all redundant inequalities, but they require an enormous amount of time to do it; others will operate quickly, but they will not even suggest how many redundant inequalities still remain. To make the convolution methods more effective, along with the redundant inequalities one can exclude the inequalities that are “quasi-redundant”. To be precise, an inequality is quasi-redundant, if its exclusion from the inequality system results in a minor change of its solution set. In more detail, these matters are considered in (Bushenkov and Lotov, 1980 and 1982b).

Speaking about possible application of the convolution methods for constructing the set *f(x)*, one can state that each problem calls for separate experimenting using various orders for the elimination of variables, as well as various methods for the elimination of redundant and quasi-redundant inequalities. Therefore application of convolution methods cannot be completely automatic, and the need arises for intervention by a human operator. For convenience, a package of programs called POTENTIAL was developed (Bushenkov and Lotov, 1984) in the beginning of 80s. It offered a way to solve several applied problems. Using the POTENTIAL package, the following record result was obtained. The model was studied that consisted of three blocks, each with 12 decision variables and 18 equalities and inequalities. The coordinating problem had four additional variables and 17 global constraints. The set *f(X)* for seven criteria was constructed, it included 180 inequalities.

A major advantage of the convolution methods is the opportunity to construct the sets *f(X)* for the systems with large number of criteria. Another advantage consists in the opportunity to construct the sets *f(X)* for non-bounded sets *X*. However, it is clear that the convolution methods have multiple disadvantages. First of all, it is obvious that these methods are inapplicable to real-life models involving hundreds of variables. The most important disadvantage of the convolution methods consists in the following: in cases where the convolution process cannot be carried out to its completion, the effort fails to yield even an approximate result.

Note that the elimination of almost redundant inequalities from a system describing a projection leads to a fundamentally new situation; the task of constructing the precise projection is replaced by the task of approximating the projection. It is logical to ask why, if one finally ends up with an approximation, one should strive for the precise projection rather than construct an approximation from the very beginning in an optimal way (e.g., by seeking an approximating polytope, which has minimal number of faces for a specified accuracy). Such an approach is used in the *Projection-Estimate Refinement* (PER) method proposed in (Bushenkov, 1981 and 1985). The PER method helped to overcome the principal drawback of the convolution methods, especially the lack of an approximate estimate of the projection in cases where it is impossible to carry the task to completion.

Projection-estimate refinement (PER) method

In this method, projection (1.5) of the polytope (1.6) is approximated by a sequence of polytopes P^0, P^1, ... , P^k, ..., such that P^k tends to $M_w \subset \boldsymbol{R}^p$ in the Hausdorff metrics. The sequence of polytopes is constructed in iterative way, beginning from P°, upon solving auxiliary convolution problems.

Prior to the *(k+1)*-th iteration of the PER method, a polytope P^k must be specified, such that

(a) $P^k \subset M_w$;

(b) vertices of P^k belong to the frontier of M_w;

(c) P^k is specified in two forms simultaneously, namely:

(c$_1$) as a convex hull of vertices $\{w^{(1)}, w^{(2)}, \dots , w^{(r)} \}$;

(c$_2$) as a solution set of the system of linear inequalities

$$P^k = \{w \in \boldsymbol{R}^q\colon <c_j,w> \le d_j, j = 1, 2, \dots , N\}, \qquad (1.11)$$

where c_j are vectors from $\boldsymbol{R}^q$, and d_j are numbers.

Transition to P^{k+1} consists of the following steps. For faces of P^k , which are specified by vectors c_j and numbers d_j, the following N linear programs are solved:

$$<c_j, w> \rightarrow max \text{ while } (v,w) \in M, j = 1, 2, ..., N. \qquad (1.12)$$

From the solutions $(v_{(j)}, w_{(j)})$ of (1.12), a solution with maximal distance of the point $w^{(j^*)}$ from P^k is selected. Then we put

$$P^{k+1} = \text{conv } \{ w^{(j^*)}, P^k \},$$

and construct the polytope P^{k+1} in the form (1.11). This completes the *(k+1)*-th iteration of the method.

The problem of constructing the polytope P^{k+1} in the form (1.11) deserves special discussion. The following method for the solution of the problem was proposed in (Bushenkov, 1981 and 1985).

Bushenkov method for constructing the convex hull of a polytope and a point

The problem consists in constructing the convex hull of a polytope given as the solution set of the linear inequality system (1.11) and of a point w^*. The convex hull must be constructed in the same form of the solution set of a linear inequality system. So, a linear inequality system must be constructed, the solution set of which coincides with the above convex hull. The method has the following steps.

(1) Inequalities of the system (1.11) violated by the point w^* are identified;

(2) Vertices $w^{(i)}$, $i = 1, 2, ... ,L$, of the polytope are identified, which belong to the faces related to the violated inequalities;

(3) The cone $K \subset \boldsymbol{R}^q$, specified by edges $w^{(i)} - w^*$, $i = 1, 2, \ldots ,L$, is constructed in the form of the solution set for the system of linear inequalities (which is equivalent to constructing the edges of the conjugate cone K^+);

(4) In the inequality system (1.11), the inequalities, which are violated by the point w^*, are eliminated; new inequalities, which are specified by the edges of K^+ and the point w^*, are included.

To construct a conjugate cone, the method uses the above methods for the convolution of systems of linear inequalities. Note that the conjugate cone is constructed in $\boldsymbol{R}^m$, but not in the original space $\boldsymbol{R}^n$. In the FGM, aimed at decision support via visualization of the set *f(X)*, the number of criteria *m* is relatively small; so, we can use the PER method for approximating the set *f(X)* for problems with a large number of decision variables. Using our experience in the convolution of systems of linear inequalities and the appropriate computer software, it is possible to use the PER method as a powerful tool for approximating the projections of polyhedral sets.

On the other hand, it is clear that the basic ideas of the PER method can be used for invention of methods for solution of a more general problem – the problem of iterative approximation of bounded convex sets by polytopes. The only requirement that exists is that one must be able to solve optimization problems with linear criteria for the approximated body. These methods could be used for approximation the set *f(X)* in the convex case, too. A class of methods for polyhedral approximation of convex bounded bodies is described in Section 5.2. These methods use the following ideas of the PER method:

(a) approximation polytopes are constructed iteratively;

(b) transition from P^k to P^{k+1} consists of two independent steps. The first step is related to optimization of (1.12) type, and the second one is concerned with the construction of conv $\{P^k, w^*\}$ in the form of (1.11) using the convolution methods.

4.2. Methods for polyhedral approximation based on evaluation of support function

In this section, the set Y is assumed to be a compact convex body, for which the support function can be evaluated. We describe here the iterative methods for approximation of convex compact bodies by sequences of polytopes that can provide approximation with any given accuracy. These methods are now the main tool for approximating the convex varieties of feasible goal vectors in the framework of FGM.

The problem of polyhedral approximation of convex compact bodies is a classic problem of applied mathematics (Gruber and Wills, 1993). Let $\boldsymbol{R}^m$ be Euclidean space with the distance d. By compact bodies we understand closed confined sets with non-empty interior. Let us consider a convex compact body C from $\boldsymbol{R}^m$. Methods for polyhedral approximation of the body C are based on evaluation of its *support function* for directions*u*, i.e. on co mputing the values of the function

$$g_C\,(u) = max\ \{<u, y>: y \in C\},$$

for directions u that belong to the unit sphere of directions

$$S = \{u \in \boldsymbol{R}^m: <u, u> = 1\}.$$

It is assumed that we are able to compute the values of the support function for any direction $u \in S$. In the framework of the FGM, we approximate the set $Y=f(X)$, for which

$$g_Y(u) = max\ \{<u, y>: y=f(x),\ x \in X\}.$$

Various optimization techniques do exist that can solve this problem for particular sets *X* and mappings *f*, especially in the case of the convex set *f(X)*. In the case of linear models, various linear programming methods can be used for computation of the values of the support function for directions *u*.

Iterative approximation of convex compact bodies by polytopes is often based on computing the values of the support function for a given finite system of directions (so called *a priori* grid)

$$\{u^1, u^2, \dots, u^L\}.$$

It is clear that the above grid neglects the actual shape of the body being approximated. For this reason, the approach based on the *a priori* grids is not the best one. G.Sonnewend (1983) proved that the methods using grids of this kind are not optimal. They require too many evaluations of the support function; polytopes constructed by them have too many vertices and faces.

In this section we describe *optimal* methods for approximating the compact convex bodies by polytopes. The optimality of the methods is related to their ability to adapt to the form of the approximated body *C*. The *adaptive* methods compute the support function for the directions identified in the approximation process. To the extent of our knowledge, the first adaptive method (NISE) was proposed by J.L.Cohon (1978). The method was used for approximating a two-dimensional convex body (actually, its non-dominated frontier).

Iterative methods of polyhedral approximation of convex bodies

By an *iterative method* for the polyhedral approximation of a body *C* we mean a method for constructing an infinite sequence of polytopes

$$P^0, P^1, \dots , P^k, \dots .$$

We restrict with the sequences of polytopes, for which the number of vertices of polytopes P^0, P^1, ... , P^k, … is increased by one on an iteration. Every next polytope is constructed on the basis of the previous one using one or several procedures of computing the support function for the set C.

We consider such sequences of polytopes P^k that approximate the body C, i.e.

$$\lim_{k\to\infty} \delta(P^k, C) = 0,$$

where $\delta(\cdot,\cdot)$ is the Hausdorff metrics, i.e.

$$\delta(C_1, C_2) = max\ \{\ sup\ \{d(x, C_2): x \in C_1\ \},\ sup\ \{d(x, C_1): x \in C_2\ \}\ \}.$$

The ability of polytope sequences to approximate convex compact bodies to any degree of accuracy is an important advantage in comparison with the approximation by a single body of a specific form such as a simplex, a parallelotope, or an ellipsoid. However, a high price has to be paid for this advantage. As both practice and theory show, the complexity of description of an approximating polytope rapidly increases if accuracy of approximation and dimension of the body increase. Nevertheless, we need to construct polytopes that approximate the body very accurate, since the shape of the frontier of the approximated body is of interest in the framework of the FGM (and not just the domain where this body would be located). Therefore, we develop methods that have an optimal complexity of approximating polytopes and are optimal in respect to the number of computing the support function. Such methods are described in this section.

We consider convex polytopes, whose vertices belong to the boundary ∂C of the approximated body C. Such class of polytopes is denoted by $\boldsymbol{P}(C)$. Another class of polytopes denoted as $\boldsymbol{Q}(C)$ is provided by convex polytopes with faces touching ∂C. Methods for iterative constructing of sequences of polytopes, which are described here, are based on the so called incremental scheme.

Incremental Scheme. Let $P^k \in \boldsymbol{P}(C)$. The $(k + 1)$-th iteration involves two steps:

Step 1. Selecting the point $y^* \in \partial C$;

Step 2. Constructing $P^k = \mathrm{conv}\ \{ y^*, P^k \}$ in the required form.

Note that the above PER method embodies the incremental scheme. Particular methods based on the incremental scheme can be characterized by algorithms of solving the problems that arise on the two above steps. Apart from the incremental schemes, other classes of iterative schemes may be used, such as the cutting schemes. We do not consider them here (see Kamenev, 1992).

Let us consider a method based on the incremental scheme. Let the initial polytope P^0 belong to $\boldsymbol{P}(C)$. Then the polytopes P^k belong to $\boldsymbol{P}(C)$ for any k. Denoting the number of vertices for a polytope $P \in \boldsymbol{P}(C)$ by $N(P)$, we obtain

$$N(P^k) = N(P^0) + k.$$

Adaptive methods

Among the methods based on the incremental schemes, one can distinguish *adaptive methods*. In the framework of the adaptive methods, the choice of the point $y^* \in \partial C$ is based on information about the shape of the polytope P^k. Actually, the choice of $y^* \in \partial C$ is adapted to the shape of the body C to the same extent as P^k approximates C. The first adaptive iterative method for approximation of multiple-dimensional compact convex bodies by polytopes, the method of Estimation Refinement (The ER method), was proposed by Bushenkov and Lotov (1982b) and modified by Chernykh in 1986 (see Chernykh, 1988). The method applies the incremental scheme and the ideas of the PER method for the polyhedral approximation of the compact convex bodies.

Let us consider a polytope $P \in \boldsymbol{P}(C)$. Let $U(P)$ denote the finite list of outer normals to its facets. Evidently, $U(P)$ is given in the case when the polytope P is given in the form of the solution set of a linear inequality system.

Method of Estimate Refinement (ER method)

Prior to the *(k + 1)*-th iteration of the method, the polytope $P^k \in \boldsymbol{P}(C)$ has to be constructed in the form of the solution set of a linear inequality system.

Step 1. Find $u^* \in U(P^k)$ which solves

$$\max\{(g_C(u) - g_{P^k}(u)) : u \in U(P^k)\}; \qquad (2.1)$$

select a point $y^* \in \partial C$ such that

$$<u^*, y^*> = g_C(u^*).$$

*Step 2***.** Find $U(P^{k+1})$ for

$$P^{k+1} = \text{conv}\ \{y^*, P^k\}$$

upon constructing a linear inequality system, which solution set coincides with $P^{k+1} = \text{conv}\ \{y^*, P^k\}$.

Surely, the initial polytope $P^0 \in \boldsymbol{P}(C)$ must be specified to start the method. Its construction is discussed in (Chernykh, 1991 and 1992), and we do not dwell on the matter here.

So, in the framework of the ER method, as in the case of the PER method, the computationally complex problem of constructing the convex hull of the polytope P^k and the point $y^* \in \partial C$ should be solved. However, the ER method applies a different method for constructing the convex hull of a polytope and a point, which is discussed at the end of the Section.

The ER method, as the PER method, has the following important property: the solution of the problem (2.1) provides a rough estimate of the distance $\delta(P^k, C)$. Indeed, the polytope

$$\hat{P}^k = \{y \in \boldsymbol{R}^m : <u, y> \leq g_C(u),\ u \in U(P^k)\ \}$$

contains the body C. Due to it, we have the internal and external estimates for the approximated body C at any iteration

$$P^k \subset C \subset \hat{P}^k .$$

Therefore, it is possible to evaluate the distance $\delta(P^k,C)$ visually, by depicting two-dimensional slices of the polytopes P^k and $\hat{P}^k$. This property of the method can be used in real-life problems for human decision to stop the approximation procedure.

Note that the external polytope $\hat{P}^k$ plays a fairly passive role in the ER method. It only helps to evaluate the distance $\delta(P^k,C)$, but does not play any role at the iterations of the method. In contrast, external polytopes play an active role in a different iterative method for polyhedral approximating the multiple-dimensional compact convex bodies, the method of Mutually Converging Polytopes (MCP) proposed in (Kamenev, 1986).

Method of Mutually Converging Polytopes (MCP)

Prior to the *(k + 1)*-th iteration of the method, we should have two polytopes constructed: the polytope $P^k \in \boldsymbol{P}(C)$ and the polytope $Q^k \in \boldsymbol{Q}(C)$, both in the form of the solution set of linear inequality systems. Then, the iteration consists of two steps:

Step 1. Find $u^* \in U(P^k)$ which solves

$$\max\{(g_{Q^k}(u) - g_{P^k}(u)) : u \in U(P^k)\}.$$

Compute a point $y^* \in C$ such that

$$<u^*, y^*> = g_C(u^*).$$

Step 2. (a) Find $U(P^{k+1})$ for

$$P^{k+1} = \text{conv}\ \{y^*, P^k\}$$

upon constructing a linear inequality system, which solution set coincides with $P^{k+1} = \text{conv}\ \{y^*, P^k\}$;

(b) Let

$$Q^{k+1} = Q^k \cap \{y \in \boldsymbol{R}^m : < u^*, y> \leq g_C(u^*)\}.$$

The vertices of the original polytope $P^0 \in \boldsymbol{P}(C)$ are supposed to be located on the faces of the original polytope $Q^0 \in \boldsymbol{Q}(C)$. For this reason, the resultant polytopes also possess this property.

As one can see, the MCP method is based on the incremental scheme. However, due to the active role of the external polytope Q^k, the MCP method requires only one evaluating the support function for the body C per iteration. In this sense, it differs from the ER method, which is related to a fairly large number of such evaluations. The direction u^*, for which the support function of the body C is computed, is selected on the basis of the polytope Q^k. Due to this, the direction can be found faster than in the case of the ER method. However, it can result in an inefficient selecting of the new vertex. Experiments support the anxiety. To avoid the problem, a new method named the Modified MCP (MMCP) method was proposed (Bourmistrova, 2000). The modification looks fairly simple – one has to introduce a threshold , which values are between one and zero, and to check the inequality

$$(g_C(u^*) - g_{P^k}(u^*)) > \quad (g_{Q^k}(u^*) - g_{P^k}(u^*)) \qquad (2.2)$$

before *Step 2* of the method is started. The condition (2.2) means that the potential vertex $y^* \in C$ is sufficiently distant from the polytope P^k . If the condition (2.2) is satisfied, the new vertex y^* is included into the polytope P^k and the related inequality $< u^*, y> \leq g_C(u^*)$ is included into the description of the polytope $Q^{k.}$. In the opposite case, the vertex y^* is not included into the polytope P^k, but the related inequality $< u^*, y> \leq g_C(u^*)$ is still included into the description of the polytope $Q^{k.}$. Then, process returns to *Step 1*. The iteration is completed only after such a direction u^* is found that satisfies (2.2). So, in contrast to the MCP method, the support function of the body C can be evaluated several times at an iteration of the MMCP method, and several new inequalities may be included into the description of the polytope $Q^{k.}$.

Note that for $\beta = 0$ the modified MCP method coincides with the MCP method, and its polytope sequence coincides with the polytope sequence constructed by the ER method in the case of $\beta = 1$. However, the ER method does not require constructing of the sequence of the polytopes $Q^{k.}$ and searching for the best direction u^* for the polytope $Q^{k.}$. By the way, the last problem is not so simple as it looks at the first glance. Though the polytopes of $Q^{k.}$ type belong to the space of the small dimension m, they are described by linear inequality systems that usually have only a small number of zeros among their coefficients. For this reason, solution of optimization problems at *Step 1* of the modified MCP method may require computational time comparable with computational time of *Step 1* in the ER method.

The above small modification of the MCP method results in new important features, which will be discussed later, after we introduce several basic concepts of the theory of adaptive methods for polyhedral approximation of multiple-dimensional compact convex bodies.

Theory of the incremental scheme

To evaluate the quality of iterative methods, a ‘reference’ sequence of polytopes must be considered that gives the best approximation of the convex compact body C. It is known (Gruber, 1983) that among the polytopes with a given number of vertices N there always exists a polytope $P_N \in \boldsymbol{P}(C)$ with the minimum of distance $\delta(C, P_N)$. The polytope P_N is denoted as the *polytope of best approximation* (PBA). It is known that $\delta(C, P_N) \to 0$ while $N \to \infty$. In the case the body C has a sufficiently smooth boundary, there exist positive constants k_C and K_C such that

$$k_C / N^{2/(m-1)} \leq \delta(C, P_N) \leq K_C / N^{2/(m-1)} \qquad (2.3)$$

where m is the dimension of the space (Bronshtein and Ivanov, 1975, Schneider and Weacker, 1981, and Gruber and Kendrov, 1982). So, the distance of the PBA from such a body decreases with the order of convergence $2/(m\text{-}1)$. The PBA can be found for extremely simple bodies only, but their sequence can be used as the ‘reference’ sequence of approximating polytopes in a general case. However, it is important to remember that the PBA-based ‘reference’ sequence of polytopes provides an ideal that is not feasible in reality since vertices of P_N are not related to vertices of P_{N-1}, and so one can not even imagine an iterative procedure that constructs the sequence of PBA.

So, the sequences of polytopes generated by approximation methods can be compared with the sequence of PBA. Since the polytopes generated by an iterative method cannot approximate the body C better than PBA, the difference between them is estimated. In (Kamenev, 1992 and 1993) the *Hausdorff class* of methods for polyhedral approximating of compact convex bodies is introduced.

Methods from this class construct polytopes that are close to the sequence of PBA. We restrict the definition of the class to the methods based on the incremental scheme. An incremental scheme-based method is denoted as a Hausdorff method with a constant $\gamma >$ 0 for a body C, if it results in a sequence of polytopes $\{P^k\}_{k = 0, 1, ...}$ for which it holds

$$\delta(P^k, P^{k+1}) \geq \gamma\delta(P^k, C), k = 0, 1, ...$$

It was shown that, for any convex compact body C, a Hausdorff method results in a sequences of polytopes that approximates the body. Then, it was proven that, for convex compact bodies with a sufficiently smooth boundary, the distance $\delta(C, P^k)$ has the same order of convergence $2/(m\text{-}1)$ asymptotically, i.e. for sufficiently large numbers N_k of vertices of the polytope P^k it holds

$$\delta(C, P^k) \sim 1 / N_k{}^{2/(m\text{-}1)}.$$

When compared with (2.3), this estimate shows that for such bodies the sequence of polytopes generated by a Hausdorff method has the same order of convergence $2/(m\text{-}1)$ as the sequence of PBA! This statement means that the Hausdorff methods are *asymptotically optimal with respect to the order of the number of vertices*. Since the number of vertices of the polytope P^k is related to the number of iterations, the Hausdorff methods are *asymptotically optimal with respect to the order of the number of iterations*.

Since the Hausdorff methods proved to be optimal with respect to the order of the number of vertices, it is interesting to know about the ratio of distances $\delta(C, P^k)$ and $\delta(C, P_{N_k})$. Results in this field were obtained in (Kamenev, 1993). We describe them in short here. Let us consider a sequence of polytopes $F = \{P^k\}_{k = 0, 1, ...}$, which approximates a compact convex body C. The value

$$\eta(F) = \liminf_{k\to\infty} \frac{\delta(C, P_{N_k})}{\delta(C, P^k)}$$

is denoted as the *asymptotic efficiency* of the method that was used for generating the sequence F. Evidently, that the value $\eta(F) = 1$ can be achieved by the sequence of PBA. For the sequence of polytopes that is not optimal, it holds $\eta(F) = 0$. For the optimal sequence of polytopes, it holds

$$0 < \eta(F) < 1.$$

In (Kamenev and Efremov, 2002) it is shown that for the sequence of polytopes, which are produced by a Hausdorff method for convex compact bodies with a sufficiently smooth boundary, it holds

$$\eta(F) \geq \frac{\left(1 - \sqrt{1-\gamma}\right)^2}{4}.$$

Other properties of the ER and MCP methods

The first example of a Hausdorff method is provided by the ER method. In (Kamenev, 1994) is proved that, for any compact convex body with a sufficiently smooth boundary, the ER method is a Hausdorff method with some constant γ and that the value of γ is asymptotically close to 1. Due to this, we can assert that, for bodies with a sufficiently smooth boundary, the ER method has the optimal convergence rate and that for its asymptotic efficiency it holds

$$\eta \geq 1/4.$$

Though the MCP method itself does not belong to the Hausdorff class, it has optimal convergence order for the bodies with a sufficiently smooth boundary (Kamenev, 1996). It means that the MCP method is *asymptotically optimal with respect to the order of the number of vertices*. Because of this fact and the fact that only one evaluation of the support function is needed on an iteration of the MCP method, it has the *asymptotically optimal number of evaluations of the support function.*

The modified MCP method, however, belongs to the Hausdorff class (Bourmistrova, 2000). For this reason, it is *asymptotically optimal with respect to the order of the number of vertices*. Though the MMCP method, in contrast to the MCP method, requires more, than one evaluation of the support function on iteration, it was proved (Bourmistrova, 2000) that it has the *asymptotically optimal number of evaluations of the support function*, too.

Computational experiments and practical use of the methods show that they provide an effective tool for approximation of compact convex bodies for $m< 8$, if the approximated ellipsoid is not too flat (the asphericity is less than 50). Approximation of two- to six- dimensional ellipsoids was explored (Dzholdybaeva and Kamenev, 1992, and Bourmistrova, 1999). Ellipsoids provide the bodies, polyhedral approximation of which is a pretty complicated task. Therefore study of the properties their approximation is very educative. In particular, the class of ellipsoids includes the sphere that is the most challenging object for approximation in Hausdorff metrics. Experimental estimates for the asymptotic efficiency of the method were computed and compared with the theoretical estimates. The results of theoretical analysis were confirmed, and the convergence constants were estimated. In particular, it turned out that the experimental asymptotic efficiency of the ER method is independent of the shape of the body being approximated and is greater than 0.5 for $m > 2$. Therefore, the above estimate of the asymptotic efficiency hopefully can be improved for $m > 2$.

Experimental comparison of the modified MCP method with the original MCP method was carried out, too. It was clear from the very beginning that the ER method is able to construct the polytopes that are better than the polytopes constructed by the modified MCP method. However, it turned that the asymptotic efficiency of all the methods is fairly the same. It turned that number of support function evaluation per iteration in the modified MCP method is only 3.5 times greater than for the original MCP method for $\beta < 0.7$. This number is much greater in the ER method. In general, the analysis of the experimental results helped to recommend the value = 0.4 for approximation of bodies with a sufficiently smooth boundary.

As we have already said, the problem that is solved at *Step 1* of the modified MCP method may require much time. The comparison of the MMCP method with the ER method was carried out on the basis of problems studied in the framework of the DSS described in Section 3.3. It turned that at least for some of the multiple criteria problems the MMCP method required less time than the ER method. For example, in the five-criterion problem that was used to describe implementation of the DSS, approximating the EPH with the precision of 1% required about four times less time in the case of MMCP than in the case of the ER method.

Important results were obtained for approximation of the bodies with non-smooth boundary, but these results are beyond the scope of book (see Kamenev 1992, 1994, 1999, 2000, as well as Bourmistrova, 2000). We can add that the adaptive methods for the polyhedral approximation of convex compact bodies offer a way to estimate several characteristics of convex bodies, such as volume, surface volume, and other Minkowski measures (Leichtweiss, 1980). However, these matters lie outside the scope of the book, too.

Constructing of convex hull of a polytope and a point

We now consider the second step of the incremental scheme; that is, constructing the convex hull of a polytope and a point. The first scheme for constructing of conv $\{p^*, P^k\}$ was introduced by Bushenkov (1981) in the framework of the PER method (it is described in Section 4.1). The scheme was based on the ideas that are now called the beneath-beyond method (Preparata and Shamos, 1985). It is based on the theorem by McMullen and Shephard (1971). We describe the idea the McMullen-Shephard theorem on the basis of a three-dimensional example.

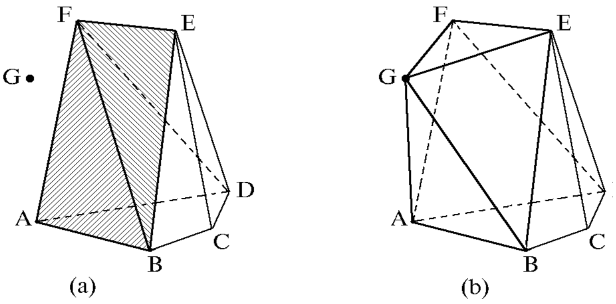

Figure 4.2.1. The beneath-beyond method

Suppose that one needs to construct the faces of the convex hull of polytope ***ABCDEF*** and point ***G*** (Figure 4.2.1 a). It is clear that the convex hull will include all faces of the original polytope invisible from point ***G*** (that is, ***ABCD, BCE, CDE, DEF*** and ***ADF***). However, it will not include any of the faces visible from point ***G*** (that is, ***ABF*** and ***BEF***, which are shown shaded in Figure). Instead, the convex hull will acquire new faces (Figure 4.2.1 b). These new faces are the faces of a minimal cone whose apex is located at ***G*** and contains the polytope ***ABCDEF***. Each such face lies in a plane passing through the edge of the polytope lying at the boundary between the visible and invisible parts of the polytope's surface. It is clear that the boundary between the visible and invisible parts consists of intersections of pairs of adjacent faces of the polytope, one being visible and another being invisible. For example, the boundary in Figure 4.2.1 consists of the edges ***AB, BE, EF,*** and ***FA***. Say, the edge ***BE*** is the intersection of the visible face ***BEF*** and of the invisible face ***BCE***. The new face ***BEG*** passes through the edge ***BE***. This scheme can be easily generalized to a space of an arbitrary dimension *m* if we recall that the boundary between the parts of the polytope that are visible and invisible from a point in $\boldsymbol{R}^m$ consists of$m$- 2 dimensional faces.

Particular methods that implement the beneath-beyond scheme differ in the way they solve the three following problems:

- How to determine whether a facet is visible from a point;
- How to determine whether two facets are adjacent or not; and
- How to transform the representation of the polytope into the convex hull.

The manner in which the above problems can be solved depends on the representation of the polytope. Within the framework of the FGM, we need a polytope to be described as the solution set of an inequality system. Such representation of polytopes was used by Bushenkov in the PER method, which included constructing the multidimensional convex hulls. The method due to Bushenkov seems to be the first published method based on the beneath-beyond scheme and implemented in the form of working software.

A bit later than Bushenkov, but independently of him, alternative methods based on the beneath-beyond scheme were proposed in unpublished papers of Kallay and Seidel who solve the above three problems in a different way. A description of the methods proposed in (Kallay, 1981) and (Seidel, 1981) can be found in (Preparata and Shamos, 1985). To represent a polytope, Kallay stores the coordinates of its vertices and a complete combinatorial structure of the polytope as an incidence graph showing the affiliation of polytope faces of all dimensions. To say whether a face is visible or not, it is necessary to determine the orientation of the half-space whose boundary passes through the vertices of the face relative to the point being attached. The adjacency of facets can be readily determined from the incidence graph. A rather sophisticated combinatorial procedure is proposed for transforming the representation of the polytope. The Seidel method is the dual of the Kallay method; it is optimal with respect to the number of operations in spaces of even dimensions. We know nothing about how successful the two methods have been in practical applications.

A new method that continues the development of the method due to Bushenkov is used now in the framework of the FGM as a part of the ER method, MCP and MMCP methods. The new method was proposed by Chernykh (1988). Though the method due to Chernykh can be applied in the case of a sequence of vertices arriving one after another, we start its description by considering the construction of a convex hull for a specified collection of s points $\{v^1, v^2, \dots , v^s\} \subset \boldsymbol{R}^m$. Once again, the convex hull must be constructed in the form of the solution set of a linear inequality system. By definition, the point $y \in \boldsymbol{R}^m$ belongs to a convex hull of the above points if there exist values $\lambda_1, \lambda_2, \dots, \lambda_s$ such that

$$y = \sum_{i=1}^{s} \lambda_i v^i , \quad \sum_{i=1}^{s} \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = 1, 2, \dots, s. \qquad (2.4)$$

Consider the space $\boldsymbol{R}^{s+m}$ of variables λ_1, λ_2,..., λ_s and y. Then, the system (2.4) specifies a polyhedral set in this space. According to the definition of the projection, the convex hull of points $\{v^1, v^2, ..., v^s\}$ is the projection of the set into the space $\boldsymbol{R}^m$ of variables y. Therefore, in order to construct the desired convex hull, one can eliminate the variables λ from the system (2.4).

We eliminate the variables λ from the system (2.4) in the same order as they are numbered. The first $m+l$ variables λ_1, λ_2, ... , λ_{m+1} can be eliminated from the system by expressing them in terms of the other variables and using the equalities that the system (2.4) contains. The remaining variables can be eliminated by the method of reduced fundamental convolution described in Section 4.1.

Let us consider the resulting system obtained after the variables λ_1, λ_2, ... , λ_q, where $q \geq m+1$, have been eliminated. The system provides, in effect, the description of the convex hull of the points v^1, v^2, ... , v^q– it is only needed to equate the variables λ_{q+1},..., λ_s to zero in it (see Chernykh, 1988, for details). The excluding the next variable λ_{q+1} from the resulting system is equivalent to attaching a point v^{q+1} to the convex hull of the points v^1, v^2, ... , v^q. So, when constructing a convex hull, the information about the points that have yet to be attached is not needed. Therefore, it is not necessary to know the future points themselves or even their number. Thus, this method can be used for constructing the polytope sequentially.

Application of the reduced fundamental convolution method (see Section 4.1) for the excluding the variables λ_1, λ_2,..., λ_s transforms the described method into the method based on the beneath-beyond scheme. The inequality index storage used in the reduced fundamental convolution method is equivalent to a partial storage of combinatorial structure of the intermediate polytope. Therefore, a polytope is, in effect, stored as a system of inequalities, each of which corresponds to a face of the polytope. Also, each inequality is stored along with information that gives the numbers of the vertices, which belong to particular faces. This information helps to solve three problems listed above. For example, the question as to whether or not a given face is visible from a new point is answered by inserting the point being attached in the linear inequality corresponding to that face.

The crucial point of the method is now the choice of adjacent faces in the convex hull of points v^1, v^2, ... , v^q with a view to constructing a new face passing through their intersection and the point v^{q+1}. If the variables λ were eliminated from system (2.4) by the Fourier method, this would produce combinations of all pairs of inequalities corresponding to one visible and one invisible face (irrespective of their adjacency). Naturally, the number of incidental inequalities would then increase catastrophically, and the method would no longer be of the beneath-beyond type. When the reduced fundamental convolution is used, the pairs that are combined correspond to adjacent faces; that is, they are combined in compliance with the beneath-beyond scheme. For this reason, the use of the reduced fundamental convolution in this method cannot produce redundant inequalities. The representation of a polytope is a fairly simple task. The transformation of indices poses no difficulty at all, and the coefficients of the inequality corresponding to a new face are calculated as a linear combination with positive coefficients of the two inequalities representing two adjacent faces.

The Chernykh method has a higher theoretical order of time complexity than the purely combinatorial method due to Kallay, because all possible pairs of visible and invisible faces have to be tested in order to establish the adjacency. Within the FGM, however, the time required for operation of the Chernykh method is quite satisfactory because the matter of adjacency is resolved by a small number of fast operations. Since *m* is small in the FGM (not larger than seven or eight), the method permits convex hulls for sets consisting of several hundred points to be constructed even on a personal computer.

With the Bushenkov method, in order to attach a point for the construction of new faces, one has to select the vertices $V = \{ v^1, v^2, \ldots , v^q \}$, which belong to the visible faces of the polytope. In the original version of the method described in section 4.1, the storing of the combinatorial structure of the polytope was not used. Therefore, the collection of vertices was found by a direct inserting of all vertices of the polytope in violated inequalities of the system. Then, the faces of the cone were computed upon the excluding the auxiliary variables $\lambda_i, i = 1,2,..,r,$ from the system of equalities and inequalities defining the cone,

$$y - y^* = \sum_{i=1}^{r} \lambda_i (v^i - y^*), \quad \lambda_i \geq 0, \;\; i = 1, 2, ..., r, \qquad (2.5)$$

where $v_i, i = 1,2,...,r,$ are the vertices that belong to the violated inequalities. The reduced fundamental convolution method was used for the excluding the variables $\lambda_i, i = 1,2,..,r$.

Let us compare the methods due to Bushenkov and Chernykh. The Chernykh method has several advantages. First, it has a simpler structure and requires less time and memory. In contrast, the attachment of one vertex to the hull in the Bushenkov method makes it necessary to eliminate from the auxiliary system (2.5) the entire set of variables λ_i whose number is equal to the number of vertices belonging to the visible faces. Secondly, the Chernykh method is more reliable because a technique was developed to control the inaccuracy due to the rounded-off errors (Chernykh, 1988). In studying this matter, it has proved convenient to use the classical concepts of combinatorial topology and to consider the collection of faces of a polytope as a cycle of an abstract simplicial complex. Owing to this device it has been possible to readily prove the validity of the simplest *a posteriori* estimate of the error in the construction of the convex hull produced by inserting all of its vertices in all inequalities of the resultant system approximately describing the hull. Note that for an accuracy check, it is necessary to store the coordinates of the vertices of the convex hull in memory (in contrast to the Bushenkov method, this is not required for operation of the main part of the Chernykh method).

However, the Bushenkov method has several advantages. It does not require implementing the adjacency test for all pairs composed of visible and invisible faces. In effect, when one more vertex is attached to the hull, only the visible part and not the entire polytope needs to be considered in the Bushenkov method. Because of this, the task of constructing the hull of a system of points is decomposed into a sequence of local subtasks. In solving each of these subtasks, there is no need to consider the current polytope as a whole. Although this property of the Bushenkov method can be important when the number of faces is great, this decomposition of the task makes it difficult to maintain the correct combinatorial structure of the polytope, which is essential to controlling the accuracy when rounded-off errors are present.

It is worth to note that fast methods for on-line display of decision maps have also been developed (Chernykh and Kamenev, 1993).

4.3. Feasible Goals Method for non-linear models

Here we describe the mathematical basis of the FGM in the non-linear case. Like in the previous section, let X be the set of feasible decisions that belongs to a decision space $\boldsymbol{R^n}$. However, now the set X is described by non-linear restrictions and may be non-convex. The mapping f: $\boldsymbol{R^n}$ $\quad$ $\boldsymbol{R^m}$ that relates decisions to m criterion values may be non-linear, too. Therefore, the set $f(X)$ is usually non-convex, and this evidence is the crucial difficulty in its approximation and visualization in the non-linear case. Say, approximation by a single convex polytope cannot be used. Moreover, because of possible sophisticated nature of the output of the non-linear models, the direct computing of the output is often the only feasible way of the model exploration. Optimization techniques based on application of analytically calculated gradients cannot be applied in a general non-linear case.

The method described herein was introduced in the paper by Kamenev and Kondrat'ev (1992) and described in the book (Lotov et al., 1999b). It is based on simulation of random decisions, i.e. on computing m-dimensional outputs of random points from the set X. In contrast to traditional approaches (see, for example, Statnikov and Matusov, 1995), which are based on computing the outputs of random feasible decisions, we do not display a list of feasible inputs and related outputs to user. Instead, the outputs provide the basis for evaluation and display of the set $f(X)$.

Concept of the method

The method is based on the global sampling of the set X, i.e. on generation of uniformly distributed random points from the set X and on computing their outputs. Selecting a small part of output points and covering the set $f(X)$ by a system of balls or boxes with centers in the selected points are used. The output points are selected in an adaptive way. The radii of balls (edges of boxes) located in the selected output points are specified in such a way that the resulting system of balls (boxes) approximates the set $f(X)$ with a desired degree of precision. Collections of two-criterion slices of such approximation can be displayed on-line by personal computers and workstations reasonably fast.

It is assumed that $\boldsymbol{R}^n$ and $\boldsymbol{R}^m$ are linear metric spaces. Let $d(v, w)$ be the distance between the points v and w in $\boldsymbol{R}^m$. It can be a usual Euclidian distance or, say, the Tchebycheff metrics

$$d(v, w) = max \{ \lambda_i |v_i - w_i|: i = 1, 2, \dots , m \},$$

where λ_i are positive values. Let > 0. We denote by (Q) the -neighborhood of the set Q; that is, the set of points distant from Q less than . The -neighborhood is a ball with the radius in the case of the Euclidian distance. In the case of the Tchebycheff metrics, it is a box (parallelotop) with edges that are parallel to the coordinate axes.

The set $X \in \boldsymbol{R}^n$ is supposed to be compact and measurable, i.e. the notion of the volume of the set X has a sense. Moreover, we assume that the set X is bodily, i.e. its volume is not zero. The mapping f: $\boldsymbol{R}^n \to \boldsymbol{R}^m$ is assumed to be continuous on X. Then, the set $f(X)$ is compact, too. Note that one may use a finite number of such sets X to describe important situations with non-continuous mappings f: $\boldsymbol{R}^n \to \boldsymbol{R}^m$, but this topic is beyond the scope of this book.

Let us consider the uniform measure μ on the set X, i.e. such a measure that the measure of any measurable subset of X coincides with its volume. For certainty, we require that $\mu (X) = 1$. We use the uniform measure, since the generated points are assumed to be uniformly distributed over the set X. However, more general measures can be applied, too.

As it was said earlier, a particular case of approximation of the set $Y = f(X)$ is considered here – its covering by a finite variety of neighborhoods of its points. The finite variety of centers of the neighborhoods is named the *covering base* and is denoted by T. The set $(T)_s$ is then the *covering* of the set Y by neighborhoods of the radius s located in the points of the covering base. For a given radius s, the notion of *completeness* $\eta_T(s)$ of a covering $(T)_s$ is defined as

$$\eta_T(s) = \mu(f^{-1}((T)_s \cap Y)), \tag{3.1}$$

i.e. $\eta_T(s)$ is the measure of the subset of points from X which outputs belong to $(T)_s$. Since $\mu(X) = 1$, we have that $\eta_T(s)$ is less or equal to one. The dependence of $\eta_T(s)$ on the radius s is said to be the *function of completeness* of the given covering $(T)_s$. It is clear that the completeness function depends on the radius s in a monotonic way – the function is not decreasing while the radius is increasing. Moreover, since the set Y is confined, the function $\eta_T(s)$ achieves its maximal value (one) at some radius s.

Let and η be some positive numbers, $\eta < 1$. We consider here the problem of constructing such covering base T, which satisfies the requirement

$$\eta_T(\varepsilon) \geq \eta. \tag{3.2}$$

The requirement (3.2) implies that the set (T) contains the output of the η-th proportion of the set X. It means that the set (T) is not very poor in comparison with Y for values of η that are close to one. Simultaneously, since the points of the covering base belong to Y, the distance of any point $y \in (T)$ from the set Y is less than ε. Therefore, the set (T) is not very fuzzy for small values of ε.

Several conflicting goals must be taken into account in the process of constructing a covering of the set Y. First of all, as it was said above, the covering will be used in the process of on-line visualization of the set Y using two-criterion slices of the covering. That is why one needs a covering by a relatively small number of neighborhoods for a fast display of such two-criterion slices. On the other hand, it is desirable to construct a covering base *T*, for which the requirement (3.2) is satisfied for sufficiently small values of ε and 1-η . Surely, these two goals are in conflict.

The problem of constructing a covering base *T*, which satisfies (3.2) for small values of ε and 1-η, can be solved in a precise way only for extremely simple models. For this reason, empirical methods must be applied for both constructing the covering base and testing the condition (3.2).

Constructing a covering base

We use the following adaptive algorithm for constructing a covering base *T*. At an iteration of the algorithm, *N* random independent points from *X* are generated and associated output points are computed. The output point most distanced from the current covering base is included into the base. Before the algorithm starts, user has to specify the number *N* of random points from *X* that are generated at an iteration and the maximum number *M* of points in the covering base *T*.

Algorithm for constructing a covering base

Specify the number of generated points *N* and the number of iterations *M*.

Initial stage. Generate an initial point from the set *X*. The initial covering base *T* is provided by its output.

Main stage consists of the following identical *M* iterations:

- generate *N* random uniformly distributed points from *X* and compute their output;

– add to the covering base the output, which is most distant from the current covering base *T*.

It is important to stress that the algorithm has an adaptive nature: the most distanced point is added to the covering base. The same strategy is used in methods for approximating the convex bodies (Lotov et al., 1999b). It seems that the value of *N* can be specified in a reasonable way on the basis of the desired precision ε, completeness η and, perhaps, dimension of criterion space *m.* However, in contrast to the convex case, a theory for specification of the value of *N* has not been developed yet. For this reason, an empirical testing of the completeness of a current covering base is performed at any iteration of the algorithm. The same *N* random points from *X* are used for it. Due to the empirical testing of the completeness at any iteration, the value of M plays minor role – the process can be stopped after a covering with the desired properties has been found.

Empirical testing a covering base

Empirical testing of the condition (3.2) is based on its transformation into another form that is convenient for empirical consideration. It is clear that an empirical testing can provide results with certain reliability χ, which is less than one. So, instead of the condition (3.2), the following condition of a sufficient quality of the covering base could be tested

$$\boldsymbol{P}\{\ \eta_T(\varepsilon) > \eta\ \} \geq \chi\ , \tag{3.3}$$

where $0 < \chi < 1$. Moreover, another modification of the test is reasonable – it turned to be convenient to estimate the completeness function $\eta_T(s)$ for all positive values of the radius *s*.

Indeed, the form of the set $Y = f(X)$ is not known in advance, and so the desired values of ε and η can be specified in advance only approximately. For this reason, the display of the completeness function $\eta_T(s)$ helps user to assess the convergence speed and to make a decision on whether to complete the approximation process at some iteration. It may happen that a combination of radius and completeness displayed by the graph of the completeness function is already satisfactory. Therefore, instead of automatic testing the condition (3.3), another problem is solved: it is required to construct a function $\bar{\eta}_T(s)$ that satisfies

$$\boldsymbol{P}\,\{\eta_T(s) > \bar{\eta}_T(s)\} \geq \chi \qquad (3.4)$$

for any $s > 0$.

Let us consider a method for constructing such function $\bar{\eta}_T(s)$. Note that in accordance to the definition of the completeness (3.1), it holds for a given covering base T that

$$\eta_T(s) = \boldsymbol{P}\,\{\, d\,(f(x),\, T) < s\,\},$$

i.e. the completeness function $\eta_T(s)$ is the distribution function of the distance $d\,(f(x),\, T)$ of the output $f(x)$ of a random point $x \in X$ from the covering base T. Therefore, empirical estimates of $\eta_T(s)$ can be obtained by statistical methods based on generation of independent random points from X.

Let us denote a sample of N random points from the set X by $H_N = \{x_1, \ldots, x_N\}$. We denote

$$\eta_T^{(N)}(s) = n(s)/N$$

where $n(s)$ is the number of points from H_N, for which it holds $d\,(f(x),\, T) < s$. For a given s, the value $\eta_T^{(N)}(s)$ is the sample (experimental) value of the completeness $\eta_T(s)$, and so it can be used for estimation of the completeness $\eta_T(s)$. We provide the related theorem without a proof.

Theorem. For any $s > 0$ it holds

$$\boldsymbol{P}\,\{\eta_T(s) > \eta_T^{(N)}(s) - \Delta\} \geq \chi \qquad (3.5)$$

where Δ is given by

$$\Delta(\chi,N) = (-ln\ (1-\chi)\ /\ 2N)^{1/2}. \qquad (3.6)$$

Comparing expression (3.5) with expression (3.4), we note that the difference $\eta_T^{(N)}(s) - \Delta(\chi,N)$ can be used to estimate the value of $\overline{\eta}(s)$.

Note that the inaccuracy Δ does not depend neither on the covering base T, nor on the covering radius s. Due to this, expression (3.6) can be used for evaluation of the number N of random points required to estimate the completeness with a given accuracy and reliability. Say, one can choose the minimal number N that satisfies

$$N \geq -ln(1-\chi)^{-1}\ /\ (2\Delta^2). \qquad (3.7)$$

The next table contains the estimations of N depending on the values of Δ and χ.

Table.* Estimations of *N (Δ, χ)

$\chi \setminus \Delta$	**0.10**	**0.05**	**0.01**
0.90	116	461	11513
0.95	150	600	14979
0.99	231	922	23026

The formulae (3.5)-(3.7) can be used in an algorithm for estimating the completeness function for a given covering base T. It is supposed that the values of the reliability χ and the inaccuracy Δ were specified.

Algorithm for estimating the completeness function

Formula (3.7) is used for evaluation of the number N needed to estimate the completeness function for the covering base T with the given values of χ and Δ .

1. N random independent points from X are generated; their outputs are computed;
2. The empirical completeness function $\eta_T^{(N)}(s)$ is constructed for the computed variety of the outputs;
3. Graphs of the empirical completeness function $\eta_T^{(N)}(s)$ and the estimate of precise completeness function $\eta_T(s)$ are displayed.

The algorithms for adaptive constructing a covering base and for estimating the completeness function of a constructed covering base are two independent algorithms. However, the estimation algorithm can be used for specification of the number of points N that must be generated on each iteration. Information provided by the completeness estimation algorithm can be displayed to user. Due to this, user will be able to monitor the process of approximation the set $f(X)$ and to intervene into the process if needed. In this case, the original values of process parameters M, ε and η will play a minor role in this process. However, it is possible to carry out the covering process without user. In this case the values of the process parameters can be used as termination conditions.

It can be shown that for any compact measurable bodily set X and any piecewise continuous mapping f, the algorithm with a properly specified number $N(1-\eta,\chi)$ and infinite M would construct the covering base T satisfying (3.3) in a finite number of iterations for any values of $\varepsilon > 0$, $0 < \chi < 1$ and $0 < \eta < 1$, if user does not stop the algorithm earlier. Other algorithms, which require a smaller number of simulation experiments, may be proposed. They are, however, outside the scope of this book (see Kamenev and Kondrat'ev, 1992).

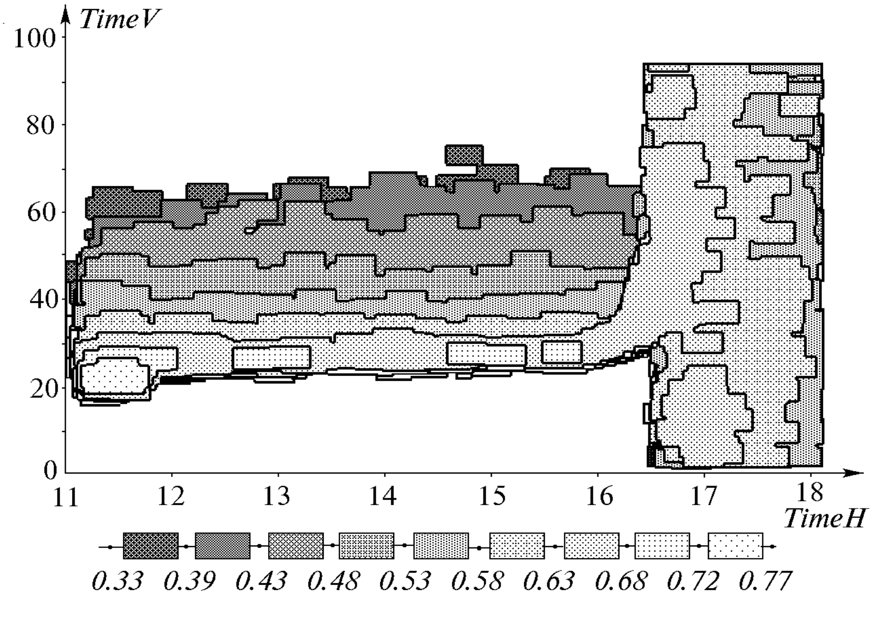

Figure 4.3.1. Black and white copy of a color display of a collection of slices for three criteria. Two criteria (to be minimized) are given on axes. The intervals of the value of the third criterion are given by shadings (by colors on the display).

Visualization, goal identification and decision computing

In the visualization process, it is convenient to use the Tchebycheff metrics. As it was said above, in this case a covering neighborhood is a box with edges parallel to the coordinate axes. Due to this, two-dimensional slices of the covering set $(T)_s$ can be computed and displayed fairly fast. Collections of slices are usually displayed. The value of the third criterion, which is associated with color, change from slice to slice. The correspondence between the value of the third criterion and color is provided in the palette located under the picture. In our book, we do not provide examples of exploration of the systems given by non-linear models (such examples are provided in Lotov et al., 1997 and 1999). For this reason, we illustrate the application of the method by an example. In Figure 4.3.1 provides a black and white copy of display with a collection of slices for a non-linear model.

User can explore different pictures of this kind displayed on-line and identify a preferable combination of criterion values (feasible goal vector) in a picture. Then, a point from the covering base T can be found, which contains to the identified goal vector in its s-neighborhood. It is possible to find a related point of the original set X: for the points of the covering base T, original points x from X are usually saved, too.

The large number of simulations needed to construct a reasonably precise covering base can be performed using the so-called *meta-computing platforms*. A meta-computing platform is a collection of computers (and possibly other resources such as visualization and storage devices) that are geographically distributed, but networked in various ways. Meta-computing platforms are inexpensive since they utilize the idle time of the collection of workstations, which is essentially free. Despite their low cost, meta-computing platforms are potentially very powerful.

However, they have features that make them much more difficult to program than traditional parallel computers. In meta-computing platforms, the number of processors available may vary over time, processors may disappear without any notice, communication latency between any given pair of processors may be high, variable and unpredictable and, finally, processors may differ in their CPU speed or amount of memory.

Usually, in order to implement a method on a meta-computing platform, one has to transform it into a special form. It is important to note that the approximation method described in this Section has this kind of a form. Due to the randomization of the simulation input, particular simulation runs play no role: the statistical analysis depends on the number of computed random outputs collected at the coordinating computer, but not on particular computers the outputs were computed at. Therefore, the dynamic availability of processors, their unreliability, heterogeneity and possible poor communication play no role for the FGM.

4.4. Approximating the Edgeworth-Pareto Hull

Methods for constructing or approximating the EPH are outlined in this Section. Once again, we assume that user is interested in decreasing the criterion values, i.e. a criterion point y' is better than a criterion point y, if and only if $y' \leq y$ and $y' \neq y$. In this case the non-dominated frontier *P(Y)* of the set *Y* is of interest and the EPH of *Y* defined as

$$Y^* = Y + R_+^m,$$

where R_+^m is the non-negative cone of $\boldsymbol{R^m}$, can be used instead of *Y*. Exploration of the EPH instead of *Y* is based on the important fact that the non-dominated frontiers of the sets *Y* and *Y** coincide, but the dominated frontiers disappear in Y^*. Note that the set Y^* can be represented in the equivalent form

$$Y^* = \{y \in \boldsymbol{R}^m: y \leq f(x), \text{ where } x \in X\}.$$

Constructing the EPH for linear models

We start with constructing the EPH for the models studied in Section 5.1. It means that the set X is polyhedral and the mapping f is linear. Let the linear inequality system that specifies the set X be

$$X = \{x \in \boldsymbol{R}^n: Hx \leq h\},$$

where H is a given matrix and h is a given vector. Let the mapping f be specified by a matrix F, such that

$$f(x) = Fx.$$

We need to construct the set

$$Y^* = \{ y \in \boldsymbol{R}^m : y \leq Fx,\ Hx \leq h \} \tag{4.1}$$

in the form

$$Q^* = \{ y \in \boldsymbol{R}^m : D^*y \leq d^*\}, \tag{4.2}$$

i.e. we need to construct the matrix D^* and the vector d^*. One can see that the only difference of the above formula from the system (1.1)-(1.2) considered in Section 4.1 consists in the presence of inequaliy instead of the equality between the vector y and the matrix F (4.1). Therefore, as in Section 4.1, we can consider the graph

$$Z = \{(x,y) \in \boldsymbol{R}^n \times \boldsymbol{R}^m: y \leq Fx,\ Hx \leq h \}$$

and construct its projection onto $\boldsymbol{R}^m$, which coincides with the set Y^*. Therefore, the set Y^* can be found by the convolution methods. Such methods have been described already in Section 4.1. All advantages and disadvantages of them have been discussed. They are valid here, too. Such approach to the EPH constructing for linear models was proposed in (Lotov, 1983). However, it has not found a broad application because of development of the methods for approximation of the EPH on the basis of polyhedral approximation of the convex compact bodies. The visible advantages of the convolution methods, which consist in the opportunity to consider a large number of criteria and consider the non-confined polyhedral sets, turned out to be not so important in real-life problems. Let us consider the methods based on polyhedral approximation of the convex compact bodies.

Polyhedral approximation of convex EPH

The methods described in this sub-section are based on approximation of Y^* by the sum of a polytope and of the non-negative cone R_+^m. It is supposed that the set Y is bounded, and so the set Y^* can be approximated by the sum of a bounded body and cone R_+^m. We do not require convexity of the set Y, but the set Y^* is supposed to be convex. In this case, the set Y^* can be approximated by the sum of the cone R_+^m and a polytope. In addition, we assume that it is possible to evaluate the support function of the set Y^* for a finite number of directions u that belong to the unit sphere of directions

$$S = \{u \in \boldsymbol{R}^m : <u, u> = 1\}.$$

As in Section 4.2, adaptive iterative methods are used that are based on the incremental scheme. Some point on the frontier of the set Y* is selected at an iteration. The selecting of the point can be based on the ideas of ER, MCM or MMCM methods, i.e. on the evaluation of the support function of the set Y^* for a finite number of directions. Then the convex hull of the selected point and of the previous approximation by the sum of a polytope and the cone R_+^m is constructed in the form of the linear inequality system.

Note that the sum of a polytope and the cone R_+^m can be considered as the EPH of a system of points. Therefore the second step of an iteration can be considered as constructing the EPH for the previous EPH and a point outside it. This idea was proposed in the paper by Chernykh (1995). It is based on the development of the method for constructing the convex hull of points, which has been described already in Section 4.2.

Let us consider a finite set of points $V = \{v^1, v^2, \dots , v^s\} \in \boldsymbol{R}^m$. The EPH for these points is defined as

$$(\operatorname{conv} V)_p = \{y \in \boldsymbol{R}^m : y \le \sum_{i=1}^{s} \lambda_i v^i , \sum_{i=1}^{s} \lambda_i = 1, \lambda_i \ge 0,\ i = 1, 2, ..., s\}.$$

Note that this definition differs from the definition of convex hull only by inequality at the variable *y*. Therefore, the polyhedral description (4.2) of this set can be found by convolution methods, too. Eliminating of the variables   in Chernikov convolution methods corresponds to iterative constructing of the EPH of a point and of the previous EPH. Additional details of the methods can be found in (Chernykh, 1995).

Non-linear case

Approximating the EPH in the non-linear case can be based on a simple substitution of the neighborhoods by the non-negative cones R_+^m (or other cones depending on user's preferences) with apexes located in the covering base points. Special algorithms that are modifications of the above algorithms were developed for approximating the EPH. This topic is, however, outside of the scope of this book.