



Using Objective Values to Start Multiple Objective Linear Programming Algorithms

Arbel, A. and Korhonen, P.

IIASA Interim Report
August 1998



Arbel, A. and Korhonen, P. (1998) Using Objective Values to Start Multiple Objective Linear Programming Algorithms. IIASA Interim Report. IR-98-066 Copyright © 1998 by the author(s). <http://pure.iiasa.ac.at/5584/>

Interim Report on work of the International Institute for Applied Systems Analysis receive only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute, its National Member Organizations, or other organizations supporting the work. All rights reserved. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage. All copies must bear this notice and the full citation on the first page. For other purposes, to republish, to post on servers or to redistribute to lists, permission must be sought by contacting repository@iiasa.ac.at

INTERIM REPORT IR-98-066/August

Using Objective Values to Start Multiple Objective Linear Programming Algorithms

Ami Arbel (ami@eng.tau.ac.il)
Pekka Korhonen(korhonen@iiasa.ac.at)

Approved by
Gordon MacDonald(macdon@iiasa.ac.at)
Director, IIASA

Contents

1. Introduction	1
2. Starting the Algorithm in Objective Space	2
3. Implications for MOLP algorithms	8
3.1 Starting MOLP Algorithms from a Dominated Point	8
3.2 Starting MOLP Algorithms from an Undominated Point	10
4. Numerical Examples	11
4.1 Starting the Search from the Nadir Solution	11
4.2 Starting the Search from the Ideal Solution	13
5. Concluding Remarks	15
References	15

Abstract

We introduce in this paper a new starting mechanism for multiple-objective linear programming (MOLP) algorithms. This makes it possible to start an algorithm from any solution in objective space. The original problem is first augmented in such a way that a given starting solution is feasible. The augmentation is explicitly or implicitly controlled by one parameter during the search process, which verifies the feasibility (efficiency) of the final solution. This starting mechanism can be applied either to traditional algorithms, which search the *exterior* of the constraint polytope, or to the algorithms moving through the *interior* of the constraints. We provide recommendations on the suitability of an algorithm for the various locations of a starting point in objective space. Numerical considerations illustrate these ideas.

Key Words: Multiple Objective Linear Programming, Interactive, Starting Solution, Aspiration Levels, Scalarizing Functions, Reference Direction, and Interior Point Algorithms.

About the Authors

Ami Arbel is Professor at the Department of Industrial Engineering, Tel Aviv University, Tel, Aviv, Israel. Currently, he is a Visiting Professor at the Systems Engineering and Engineering Management Department, Chinese University of Hong Kong.

Pekka Korhonen is Project Leader of the Decision Analysis and Support Project at IIASA, and also Professor of Statistics at the Department of Economics and Management Science, Helsinki School of Economics and Business Administration.

Using Objective Values to Start Multiple Objective Linear Programming Algorithms

Ami Arbel
Pekka Korhonen

1. Introduction

The field of multiple-objective linear programming (MOLP) has attracted a lot of attention since the early 70's and many approaches were developed to address these problems. In the absence of a precise statement of a decision-maker's (DM) preference for outcomes in objective space, MOLP problems do not generally have a unique solution like in single-objective linear programming. Instead, a family of "*reasonable*" (*nondominated*) solutions is identified, and the intervention of a DM is required to find the "*most preferred*" one. The resulting set of procedures developed for these types of problems is referred to as *interactive methods* for MOLP problems. A number of approaches and computer implementations were developed over the years differing from each other in the way they assess preferences from the DM and the way they derive search directions to move from the current iterate to the next. For a review of Multiple Criteria Decision-Making (MCDM) in general as well as MOLP, and interactive MOLP procedures, see Steuer [1986].

While the specifics of these procedures vary, they share one common feature: at each iteration, a single-objective linear programming problem is formulated to generate a candidate solution or a set of candidate solutions for a DM's examination. The objective of that single-objective problem reflects the DM's stated preferences and is referred to as a *scalarizing function*. Thus the solutions the DM obtains for examination reflect his/her expectations. Because the final choice of the DM depends on his/her search path, it is very important to pay attention to an initial solution. The initial solution is typically generated by a procedure using some heuristic rules or preference information, which is initially gathered from the DM in objective space. A single-objective linear programming problem, however, is always formulated in decision space. Thus the initial solution is first generated in the same space - and then mapped into objective space for the DM's examination. This approach may cause a problem, because it does not provide the flexibility to specify an initial (feasible) solution in objective space.

Our aim in this paper is to propose a mechanism, which enables MOLP-algorithms to start from any point in objective space. We may, for instance, start the algorithm from

an *ideal point* or a *nadir point*. There are several classical MOLP-algorithms (see, e.g. Benayoun et al. [1971], Zeleny [1982, 153-178], Steuer and Choo [1983]), where the ideal solution plays an important role. The ideal solution, however, is not generally feasible, and thus not a feasible starting point for an algorithm. Traditional algorithms operate with a nondominated point or a set of nondominated points. The DM can make comparisons between an ideal point and nondominated points, but does not get support for a search in the "intermediate" region. However, in some situations it may be very natural for the DM to start from the ideal point and iterate toward the nondominated frontier and stop upon reaching it. On the other hand, in some situations the best algorithm may be one based on a *win-win* strategy. Then a DM starts from the nadir solution and operates along directions, which always improve all objectives simultaneously until reaching the nondominated frontier.

For the first situation, we may use a traditional algorithm for which only a simple augmentation of the model is needed. For the latter problem, we opt for the use of the interior MOLP algorithm proposed by Arbel and Korhonen [1996a]. The same algorithmic recommendations are valid if, instead of the ideal point, the DM is using as a starting point any point not dominated by any feasible point, or when any point dominated by at least one feasible point replaces the nadir point. Our approach is general in the sense that we augment the problem in the form most convenient to us. When we use a traditional approach, the best way might be to first find an initial (feasible) solution and its optimal basis, and then augment the model. When an Interior MOLP-algorithm is used, the required augmentation is already used in the start of the algorithm.

The paper is arranged as follows. In section 2 we provide an approach that uses stated values in objective space to generate a starting solution in decision space. We use this starting solution to generate a sequence of iterates that move to a point on the nondominated frontier. Section 3 highlights possible uses for this starting solution in the context of MOLP algorithms. Section 4 illustrates it with an example, and section 5 provides a summary and some suggestions for future research.

2. Starting the Algorithm in Objective Space

Multiple Objective Linear Programming (MOLP) problems are usually described through

$$\begin{aligned} & \text{"max"} \quad \mathbf{v} = \mathbf{C}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{S} = \{\mathbf{x} \in \mathfrak{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{b} \in \mathfrak{R}^m\}, \end{aligned} \tag{2.1}$$

where \mathbf{A} is an $m \times n$ constraints matrix, \mathbf{C} is a $q \times n$ objective matrix whose rows C_i , $i = 1, 2, \dots, q$, are formed by the q individual objectives, $v_i = C_i \mathbf{x}$ ($1 \leq i \leq q$), and where $\mathbf{v} \in \mathfrak{R}^q$ is the objective vector. Moreover, we assume that \mathbf{A} and \mathbf{C} are of full row rank. The set, \mathcal{S} , of feasible solution vectors is a subset in the so-called *decision (variable) space* (\mathfrak{R}^n). Similarly, the set $\mathcal{V} = \{\mathbf{v} \in \mathfrak{R}^q : \mathbf{v} = \mathbf{C}\mathbf{x}, \mathbf{x} \in \mathcal{S}\}$ of all objective vectors corresponding to feasible solution vectors $\mathbf{x} \in \mathcal{S}$, defines the so-called *objective space*.

Any iterative algorithm—including an MOLP algorithm—requires a starting solution vector from which to get the iterative process underway. Since MOLP problems are usually formulated in decision space, this is also where the starting solution is specified. This, however, is not where the more meaningful data – that

associated with the objectives – is gathered. The importance of the latter is due to the way the search for a solution is guided. Modern MOLP algorithms derive their intermediate steps through interaction with a DM. Such an interaction is often done in either of two ways: either values of objectives are presented to the DM for relative preference elicitation, or the DM is allowed to specify levels of aspirations for the different objectives. Regardless of the specific manner of interaction, the information received from the DM is stated in terms of values from *objective space*. We, therefore, seek ways of using preference stated in objective space to start and evolve an MOLP algorithm in decision space.

Considering starting an algorithm in objective space raises two issues: which are natural candidates for starting points in this space, and how to incorporate them into the iterative process. Considering the first question we have two candidates that readily spring to mind: the *Ideal* and the *Nadir* points. The former provides a point (usually infeasible) that summarizes the best possible values for each of the individual objectives. The latter (often also infeasible), provides a point where the worst values for all nondominated solutions are summarized. While these points address different issues, both are defined in objective space. Regardless of the point selected as a starting point, we face the problem of finding its (usually non-unique) corresponding point in decision space. These two points are illustrated in *Figure 2.1* below, which depicts— in objective space —some problem having seven constraints and two objectives. The nondominated set (corresponding to the efficient frontier in decision space) is shown shaded.

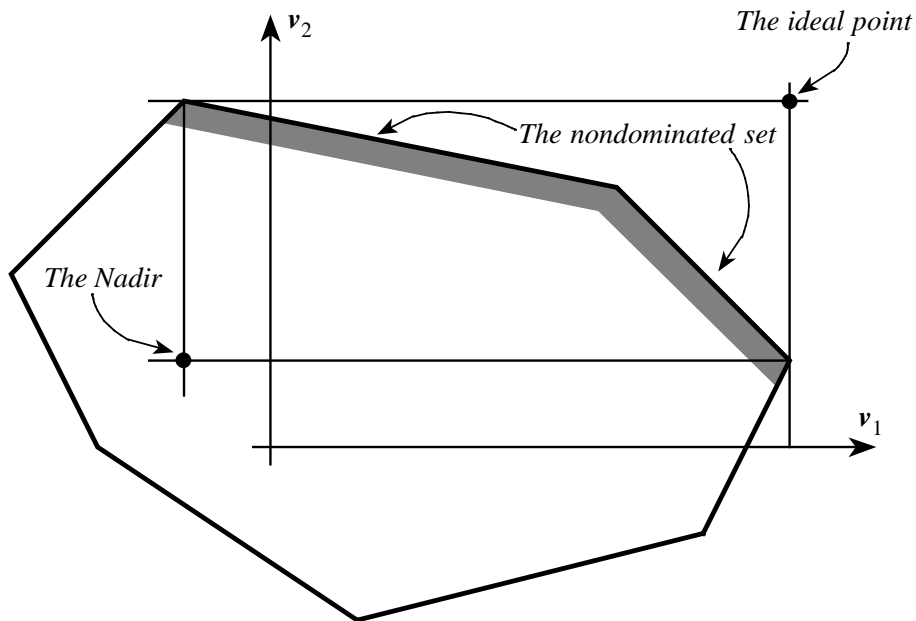


Fig. 2.1, Problem description in objective space.

Let us examine the use of the ideal and nadir points as candidates for a starting solution. If one uses an algorithm that requires statements of aspiration then, clearly, the ideal point provides a good point to start with. However, if one use an algorithm, which generates a sequence of, iterates by eliciting, say, trade-off information between the objectives then, perhaps, the nadir may offer a better choice. The reasoning goes as

follows. We know that whatever solution we end up with will be on the nondominated set. Therefore, if one starts from the point having the lowest possible value for the objectives on the nondominated set, then progress is made by improving values of all objectives. Such a sequence of iterates defines a so-called *win-win* trajectory. Moving along such a trajectory may appeal to the DM more than simply traversing the nondominated set. The reason is that rather than trading off one objective for another—as is the case when traversing the nondominated set—we move from one point to another while improving *all* objectives. Usually, the ideal and nadir points are not feasible solutions, and so is often the DM's specified aspiration point. The ideal and nadir points are not the only reasonable starting solutions. In some cases, it may be more meaningful to start the algorithm from the DM's specified aspiration values, regardless of its specific value as it represent some point of value to the DM.

Given a point in objective space—such as the ideal or the nadir—leaves us with the task of finding its corresponding point in decision space. We note that the image of the point in objective space has to satisfy the constraints defining S as in (2.1). The problem we face, therefore, starts by stating a desired starting point, \mathbf{g}_0 , in objective space and we wish to find its image in decision space. The latter should satisfy the defining equation for the objectives, namely

$$\mathbf{C}\mathbf{x}_0 = \mathbf{g}_0 \quad (2.2)$$

Our task, therefore, is to find a point, \mathbf{x}_0 , that satisfies (2.2) for any given starting objective vector, \mathbf{g}_0 , as well as satisfying $\mathbf{x}_0 \in S$. Since the $q \times n$ objective matrix, \mathbf{C} , is usually non-invertible, the required solution may not be unique. Conditions for solvability as well as an operational procedure are discussed next.

Traditionally, the algorithms do not really start from point \mathbf{g}_0 . Instead it is first projected onto the set of (weakly) nondominated solutions and the solution $\mathbf{v} \in V$ is used as a starting solution. The projection can be made e.g. by using a so-called Chebyshev-type (achievement) scalarizing function as proposed by Wierzbicki [1980]. A typical form for such a function is given below:

$$s(\mathbf{g}, \mathbf{v}, \mathbf{w}, \phi) = \max_{k \in K} \left[\frac{g_{0k} - v_k}{w_k} \right] + \phi \sum_{i=1}^q (g_i - v_i) \quad (2.3)$$

where $K = \{1, 2, \dots, q\}$, parameter $\phi > 0$, and $\mathbf{w} > \mathbf{0} \in \mathfrak{R}^q$ is a (given) vector of weights. By minimizing $s(\mathbf{g}, \mathbf{v}, \mathbf{w}, \phi)$ subject to $\mathbf{v} \in V$, we find a nondominated solution vector \mathbf{v}^* . If we let $\phi = 0$, then the solution is weakly nondominated, but may be dominated.

It is easy to see that given an MOLP problem as shown in (2.1) and any (feasible or infeasible) stated objective vector $\mathbf{g} \in \mathfrak{R}^q$, the projection of this point using the scalarizing function (2.3) onto the nondominated set is provided through the solution of the following auxiliary linear programming problem.

$$\begin{aligned}
\min \quad & \alpha + \phi \sum_{i=1}^q (g_i - v_i) \\
\text{s.t.} \quad & \\
& \mathbf{C}\mathbf{x} + \alpha\mathbf{w} \geq \mathbf{g} \\
& \mathbf{x} \in \mathcal{S}, \alpha \text{ free.}
\end{aligned} \tag{2.4}$$

Note that solving (2.4) results in an optimal solution in decision space, which corresponds to the projection of the stated objective vector in objective space. Subtracting a surplus nonnegative vector, \mathbf{z} , from the inequality constraint in (2.4) results in

$$\begin{aligned}
\min \quad & \alpha + \phi \sum_{i=1}^q (g_i - v_i) \\
\text{s.t.} \quad & \\
& \mathbf{C}\mathbf{x} + \alpha\mathbf{w} - \mathbf{z} = \mathbf{g} \\
& \mathbf{x} \in \mathcal{S}, \alpha \text{ free}, \mathbf{z} \geq \mathbf{0}.
\end{aligned} \tag{2.5}$$

Defining now an augmented solution vector through $\bar{\mathbf{x}} = [\mathbf{x} \quad \alpha \quad \mathbf{z}]^T$, the problem in (2.4) is compactly described through

$$\begin{aligned}
\min \quad & \alpha + \phi \sum_{i=1}^q (g_i - v_i) \\
\text{s.t.} \quad & \\
& \bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}} \\
& \mathbf{x} \geq \mathbf{0}, \mathbf{z} \geq \mathbf{0}, \\
& \alpha \text{ free.}
\end{aligned} \tag{2.6}$$

where

$$\bar{\mathbf{A}} = \left[\begin{array}{c|c|c} \mathbf{A} & \mathbf{0} & \mathbf{O} \\ \hline \mathbf{C} & \mathbf{w} & -\mathbf{I}_q \end{array} \right], \quad \bar{\mathbf{b}} = \left[\begin{array}{c} \mathbf{b} \\ \mathbf{g} \end{array} \right] \tag{2.7}$$

$$\mathbf{x}, \mathbf{z} \geq \mathbf{0}.$$

To improve readability of the paper, we assume henceforward that $\phi = 0$.

The original MOLP-problem is now reduced to a single-objective optimization problem. The augmented system is amenable now to the application of any single-objective linear programming algorithm.

Model (2.6) projects any given point $\mathbf{g} \in \mathfrak{R}^q$ onto the nondominated frontier, but not provides any support to start from this given point. We will modify the problem (2.6) in such a way that an interactive search can be started from any aspiration vector \mathbf{g} . We consider two kinds of starting strategies:

- we may start both the solution algorithm and an interactive search at the same time or
- we may first start the algorithm and find a nondominated solution, and then modify the problem in such way that the DM starts an interactive search from a given aspiration level point.

The first approach is a natural approach to use in the connection with interior point algorithms, and we may, therefore, directly apply the technique developed for these algorithms. The latter strategy works very well and is easy to implement in the case of a traditional algorithm. In either case, we proceed as follows.

Let us assume that one chooses any nonnegative vector, $\mathbf{x}_0 \geq \mathbf{0}$ as an initial *guess* for a solution (note that when using an interior point approach, such an initial vector has to be strictly positive). The primal feasibility error is then readily evaluated through

$$\rho_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 \quad (2.8)$$

By adding this feasibility error as an additional column to the constrains matrix, \mathbf{A} , we now arrive at an augmented systems of constraints given by

$$\left(\mathbf{A} \mid \rho_1\right) \begin{pmatrix} \mathbf{x}_0 \\ x_{n+1} \end{pmatrix} = \mathbf{b} \quad (2.9)$$

it is easily verified that the solution given by $\mathbf{x} = [\mathbf{x}_0 \ 1]^T$, is feasible for the augmented system. This takes care of the upper part of the system shown in (2.6). To address the lower part we proceed much in the same manner. We start by defining another feasibility error, this time measuring how close we come to satisfying $\mathbf{C}\mathbf{x}_0 = \mathbf{g}_0$. Denoting this error by

$$\rho_2 = \mathbf{g}_0 - \mathbf{C}\mathbf{x}_0 \quad (2.10)$$

We augment the constraint matrix in (2.5) by adding this error vector. The resulting system of constraints assumes now the form given by

$$\left(\begin{array}{c|c|c|c} \mathbf{A} & \rho_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \rho_2 & \mathbf{w} & -\mathbf{I}_q \end{array}\right) \begin{pmatrix} \mathbf{x}_0 \\ x_{n+1} \\ \alpha \\ z \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{g}_0 \end{pmatrix} \quad (2.11)$$

the lower part of (2.11) has to satisfy the condition given by

$$\mathbf{C}\mathbf{x}_0 + \rho_2 + \mathbf{w}\alpha - z = \mathbf{g}_0 \quad (2.12)$$

Given our earlier choice for the primal solution, this becomes

$$\mathbf{C}\mathbf{x}_0 + (\mathbf{g}_0 - \mathbf{C}\mathbf{x}_0)(1) + \mathbf{w}\alpha - z = \mathbf{g}_0, \quad (2.13)$$

where $\rho_2 = \mathbf{g}_0 - \mathbf{C}\mathbf{x}_0$. Hence it follows that since $\mathbf{w} > \mathbf{0}$, choosing $\alpha = 1$ and the surplus vector, \mathbf{z} , through $\mathbf{z} = \mathbf{w}\alpha > \mathbf{0}$, satisfies the nonnegativity constraints for \mathbf{z} . Note that the above formulation almost satisfies the requirement imposed by interior point algorithms of maintaining all decision variables strictly positive. The only exception is the scalar α , which is formulated as a free variable rather than a nonnegative one. We, therefore, redefine this parameter through the simple transformation given by

$$\alpha = \alpha^+ - \alpha^-, \text{ where } \alpha^+, \alpha^- > 0 \quad (2.14)$$

With this definition, (2.11) now becomes

$$\left(\begin{array}{c|c|c|c|c} \mathbf{A} & \rho_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \rho_2 & \mathbf{w} & -\mathbf{w} & -\mathbf{I}_q \end{array} \right) \begin{pmatrix} \mathbf{x}_0 \\ x_{n+1} \\ \alpha^+ \\ \alpha^- \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{g} \end{pmatrix} \quad (2.15)$$

which leads to the following restriction on the choice of \mathbf{z}

$$\mathbf{C}\mathbf{x}_0 + (\mathbf{g}_0 - \mathbf{C}\mathbf{x}_0) + \mathbf{w}\alpha^+ - \mathbf{w}\alpha^- - \mathbf{z} = \mathbf{g}, \quad (2.16)$$

from which follows

$$\mathbf{z} = \mathbf{w}\alpha^+ - \mathbf{w}\alpha^- \quad (2.17)$$

and the nonnegativity for \mathbf{z} is assured whenever we choose $\alpha^+ > \alpha^-$. A choice of starting values for the system in (2.15) is now given as follows

$$\mathbf{x}_0 > \mathbf{0}, \quad x_{n+1} = 1, \quad \alpha_0^+ = 2, \quad \alpha_0^- = 1, \quad (2.18a)$$

and where

$$\rho_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \rho_2 = \mathbf{g}_0 - \mathbf{C}\mathbf{x}_0, \text{ and } \mathbf{w} > \mathbf{0}. \quad (2.18b)$$

Most traditional simplex-based MOLP algorithms (see, e.g. Steuer [1986]) operate on a nondominated frontier. A nondominated solution is an optimal solution of problem (2.5). It is found as a projection of an aspiration vector \mathbf{g} onto the nondominated frontier. The projection, in turn, depends on the particular choice made for the weight vector $\mathbf{w} > \mathbf{0}$. In case we would like to start the search from a given aspiration point \mathbf{g}_0 , we have to modify the original problem in such a way that \mathbf{g}_0 is the optimal solution of the modified problem. For necessary modification, we need to know a nondominated solution of the original problem (2.5) with $\phi = 0$. Let \mathbf{x}_0 be such kind of solution. Now we can reformulate the problem in the similar way as we did in (2.13):

$$\begin{aligned} & \min \alpha \\ & \text{s.t} \\ & x_{n+1} = 1 \\ & \mathbf{C}\mathbf{x} + (\mathbf{g}_0 - \mathbf{C}\mathbf{x}_0)x_{n+1} + \alpha\mathbf{w} - \mathbf{z} = \mathbf{g}_0 \end{aligned} \quad (2.19)$$

$$\mathbf{x} \in S, \alpha \text{ free}, \mathbf{z} \geq \mathbf{0},$$

where the controlling parameter x_{n+1} is initially set one. It is easy to see that if $\mathbf{C}\mathbf{x}_0$ is a nondominated solution of the original problem, then $\mathbf{C}\mathbf{x}_0 + (\mathbf{g}_0 - \mathbf{C}\mathbf{x}_0)x_{n+1}$ is a nondominated solution of the modified problem, and it has value \mathbf{g}_0 , when $x_{n+1} = 1$. Thus, we can simply start a traditional simplex-based MOLP algorithm from any given point by making it a nondominated solution of a modified problem. When the control parameter x_{n+1} have value 0, we have a nondominated solution of the original problem.

3. Implications for MOLP algorithms

Having considered the starting issue from two different perspectives, we turn now to the question of integrating them into an actual algorithm. To this end we consider both an interior as well as an exterior approach as is discussed here.

3.1 Starting MOLP Algorithms from a Dominated Point

Starting an MOLP interior algorithm requires both a starting point as well as a stepping mechanism that moves us away from it. The first issue was discussed in the previous section, integrating it with the second is discussed next.

Interior-point linear programming algorithms address problems described in standard form through

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{3.1}$$

When using an interior point algorithm one starts from an initial point, generates a step direction vector, and proceeds along it to the next iterate. This sequence of operations continues until some termination conditions are met. The initial iterate can be either a feasible or an infeasible point in decision space. Each case leads to different algorithmic implementation. Assuming, for example, a starting feasible and interior solution, \mathbf{x}_0 , is available (that is $\mathbf{A}\mathbf{x}_0 = \mathbf{b}$, and $\mathbf{x}_0 > \mathbf{0}$) the algorithm generates a step direction vector $\Delta\mathbf{x}$ that moves from the current iterate in an ascent direction while preserving feasibility. Denoting the new iterate by \mathbf{x}^{new} , we have

$$\mathbf{x}^{new} = \mathbf{x}_0 + \Delta\mathbf{x}. \tag{3.2}$$

The step direction vector $\Delta\mathbf{x}$ is algorithm-specific and varies from one case to another depending on the requirements imposed on the sequence of iterates. For example, to maintain feasibility we need $\mathbf{A}\mathbf{x}^{new} = \mathbf{b}$, and to move in an ascent direction we need $\mathbf{c}^T \mathbf{x}^{new} \geq \mathbf{c}^T \mathbf{x}_0$. If these are the only two requirements, then the so-called *affine-scaling primal* algorithm provides an appropriate mechanism for generating a sequence of interior iterates that satisfies these two requirements. If, in addition to feasibility and ascent, one requires centering the sequence of iterates, the so-called *path-following primal-dual* algorithm is the algorithm of choice. If a starting feasible and interior solution is not available, one has to build some sort of an *infeasible start* mechanism

into the selected algorithm. These constructs are available for both the primal as well as for the primal-dual algorithms.

The problems formulated in the previous section lend themselves readily to a direct application of an interior-point linear programming algorithm. Specifically, the problem we address is described through

$$\begin{aligned} \min \quad & Mx_{n+1} + \underbrace{(\alpha^+ - \alpha^-)}_{\alpha} \\ \text{s. t.} \quad & \end{aligned} \tag{3.3}$$

$$\left(\begin{array}{c|c|c|c|c} \mathbf{A} & \rho_1 & \mathbf{0} & \mathbf{0} & \mathbf{O} \\ \mathbf{C} & \rho_2 & \mathbf{w} & -\mathbf{w} & -\mathbf{I}_q \end{array} \right) \begin{pmatrix} x_0 \\ x_{n+1} \\ \alpha^+ \\ \alpha^- \\ z \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{g} \end{pmatrix}$$

where

$$\rho_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \rho_2 = \mathbf{g}_0 - \mathbf{C}\mathbf{x}_0, \tag{3.4}$$

and where $\mathbf{x}_0 > \mathbf{0}$, and $\mathbf{w} > \mathbf{0}$, are any positive vectors. As shown earlier, a choice of starting values for the system in (3.3) is given through

$$x_{n+1} = 1, \quad \alpha_0^+ = 2, \quad \alpha_0^- = 1, \tag{3.5}$$

The problem in (3.3) and its starting vector are described compactly through

$$\begin{aligned} \min \quad & \bar{\mathbf{c}}^T \bar{\mathbf{x}} \\ \text{s. t.} \quad & \bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}} \\ & \bar{\mathbf{x}} \geq \mathbf{0}, \end{aligned} \tag{3.6}$$

where the augmented objective vector, $\bar{\mathbf{c}}$, is defined by $\bar{\mathbf{c}} = [\mathbf{0} \quad M \quad 1 \quad -1 \quad 0]^T$, the initial augmented vector, $\bar{\mathbf{x}}_0$, is defined by $\bar{\mathbf{x}}_0 = [x_0 \quad x_{n+1}^0 \quad \alpha_0^+ \quad \alpha_0^- \quad z_0]^T$, where its components satisfy $\mathbf{x}_0 \in \mathfrak{R}^n$, $z \in \mathfrak{R}^q$, $\mathbf{x}_0 > \mathbf{0}$, $x_{n+1}^0 = 1$, $\alpha_0^+ = 2$, $\alpha_0^- = 1$ and where, in addition, we have

$$\bar{\mathbf{A}} = \left(\begin{array}{c|c|c|c|c} \mathbf{A} & \rho_1 & \mathbf{0} & \mathbf{0} & \mathbf{O} \\ \mathbf{C} & \rho_2 & \mathbf{w} & -\mathbf{w} & -\mathbf{I}_q \end{array} \right), \quad \mathbf{w} \in \mathfrak{R}^q, \quad \rho_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \rho_2 = \mathbf{g}_0 - \mathbf{C}\mathbf{x}_0. \tag{3.7}$$

Since we have a starting feasible and interior solution vector available for the problem in (3.3), any algorithm based on a feasible start is appropriate. Of these, both the affine-scaling primal (primal, for short) as well as the path-following primal-dual algorithms are proper candidates. Since the former is easier to describe, we use it in our subsequent discussion.

Using the primal algorithm, we define a diagonal matrix $\bar{\mathbf{D}}$, having the starting vector, $\bar{\mathbf{x}}_0$, along its diagonal. The step direction vector, $\Delta\bar{\mathbf{x}}$, is then derived through

$$\Delta\bar{x} = \overline{\mathbf{D}}^2(\bar{c} - \overline{\mathbf{A}}^T \mathbf{y}) \quad (3.8)$$

where the vector, \mathbf{y} , is derived by solving

$$(\overline{\mathbf{A}}\overline{\mathbf{D}}^2\overline{\mathbf{A}}^T)\mathbf{y} = \overline{\mathbf{A}}\overline{\mathbf{D}}^2\bar{c} \quad (3.9)$$

The next iterate is derived through

$$\bar{x} = \bar{x}_0 + \rho\alpha\Delta\bar{x}, \quad (3.10)$$

where α is a *step size* that maintains the nonnegativity constraints and where $0 < \rho < 1$, is a *step size factor* that keeps the new iterate interior.

Note that the step generating mechanism of (3.8)-(3.10) applied to the system in (3.6)-(3.7) will trace a sequence of iterates from any given starting solution to a termination point on the boundary of the constraints polytope. The specific purpose of this paper is to illustrate the use of special starting points in objective space in tracing this sequence of iterates. This is demonstrated through a numerical example in section 4.1 below.

3.2 Starting MOLP Algorithms from an Undominated Point

Traditional MOLP-algorithms provide a DM with a possibility of searching the exterior of the constraint polytope, i.e. traversing the set of nondominated points. The philosophy behind this approach is the assumption that dominated solutions are not interested to a rational DM. For this reason it is natural to use a traditional MCDM-algorithm in cases when the DM prefers to start the search from a pre-specified aspiration point - not dominated by any feasible point.¹ Implementing such an approach only requires the availability of a nondominated solution.

To illustrate the idea, we use as an example—perhaps somewhat biased—our own *Reference Direction Approach* proposed by Korhonen and Laakso [1986]. This method extends the *Reference Point Approach* by projecting a *direction* instead of a *point* onto the efficient frontier. Technically, it is accomplished by parametrizing the scalarizing function. Korhonen and Wallenius [1988] further developed this idea by introducing a dynamic specification of the reference direction. The resulting method is termed *Pareto Race*. The reference direction method and its dynamic extension enable the DM to perform a continuous (exterior) search on the nondominated frontier.

The mathematical model used in the reference direction approach is a simple modification of (2.19):

$$\begin{aligned} & \min \alpha \\ & \text{s.t} \\ & \mathbf{C}\mathbf{x} + \alpha\mathbf{w} - \mathbf{z} = \mathbf{g} + \mathbf{tr} \\ & \mathbf{x} \in S, \alpha \text{ free}, \mathbf{z} \geq \mathbf{0}. \end{aligned} \quad (3.11)$$

¹ To refer to this kind of point, we use the term 'undominated', because the word 'nondominated' is used to refer to a feasible point not dominated by any other feasible point.

where $t : 0 \rightarrow \infty$ and $r \in \mathfrak{R}^k$ is a reference direction.

The reference direction is the parameter which controls the algorithm. In the original approach, a reference direction was specified as a vector starting from the current solution and leading towards the aspiration levels. The DM is asked to specify aspiration levels for the criteria. The solutions provided to the DM's evaluation are all nondominated.

Let's assume now that we have an efficient solution x_0 and the DM would like to start the search from point g_0 . Applying the reference direction approach to our problem, we modify the problem formulation (2.19) as follows:

$$\begin{aligned}
 & \min \alpha \\
 & \text{s.t} \\
 & \quad x_{n+1} = 1 - \Delta \\
 & \quad \mathbf{C}x + (\mathbf{g}_0 - \mathbf{C}x_0)x_{n+1} + \alpha w - z = \mathbf{g}_0 + tr \\
 & \quad x \in S, \alpha \text{ free}, z \geq \mathbf{0}, \delta \geq 0.
 \end{aligned} \tag{3.12}$$

where initially $\Delta = t = 0$. By controlling the parameter Δ we may vary the value of x_{n+1} , and controlling t and the parameter vector r , we may freely search the efficient frontier corresponding to the current value of parameter x_{n+1} . Finally, we may end up with a nondominated solution, when $x_{n+1} = 0$.

4. Numerical Examples

In this section we illustrate the two starting approaches described earlier through a numerical example.

4.1 Starting the Search from the Nadir Solution

Consider the MOLP problem described through

$$\begin{aligned}
 & \text{Max} \quad x_1 \\
 & \text{Max} \quad x_2 \\
 & \text{s.t.} \quad -x_1 + 2x_2 \leq 14 \\
 & \quad \quad x_1 + 3x_2 \leq 26 \\
 & \quad \quad 2x_1 + x_2 \leq 17 \\
 & \quad \quad x_1 - x_2 \leq 4 \\
 & \quad \quad x_1 + x_2 \geq 9 \\
 & \quad \quad x_1, x_2 \geq 0
 \end{aligned}$$

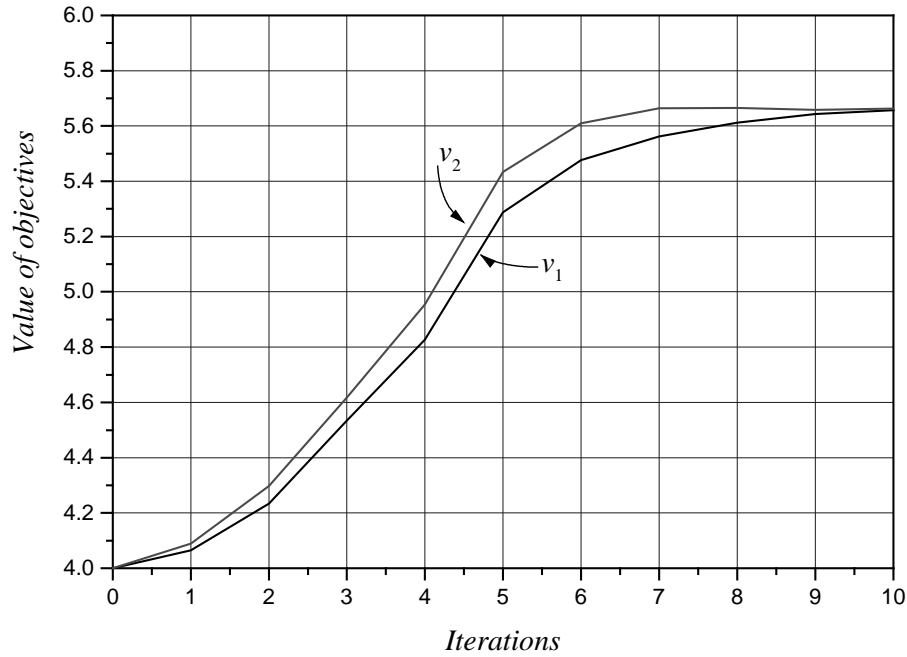
It is easy to verify that for this problem the ideal and the nadir points are (7, 8) and (3, 2), respectively.

Our first task is to find a feasible starting solution for this problem. Using the primal algorithm, adds the additional requirement of having a solution vector that is interior. Converting the problem shown above to standard form we need to specify a starting solution, $\mathbf{x}_0 \in \mathfrak{R}^7$. Letting \mathbf{x}_0 be the vector of all ones, introduces an initial feasibility error evaluated through $\rho_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. Next, we want to start from a given point in objective space, \mathbf{g}_0 . The choice of \mathbf{x}_0 , introduces another feasibility error in objective space, evaluated through $\rho_2 = \mathbf{g}_0 - \mathbf{C}\mathbf{x}_0$. Choosing the remaining starting components as shown in (3.6)-(3.7) we are ready to generate a sequence of iterates as described in section 3.1. Using an aspiration vector of (10,10), and a step size factor of $\rho = 0.6$, the following results are derived.

Table 4.1: Iteration summary ($\rho = 0.6$)

k	$x(1)$	$x(2)$	α	$v_1 = c_1^T x$	$v_2 = c_2^T x$
0	1.0000	1.0000	1.0000	4.0000	4.0000
1	1.1413	1.1660	0.9746	4.0650	4.0897
2	1.5162	1.5797	0.9060	4.2342	4.2977
3	2.2323	2.3160	0.7673	4.5343	4.6180
4	3.1064	3.2337	0.5735	4.8270	4.9543
5	4.5991	4.7458	0.2294	5.2873	5.4341
6	5.2018	5.3339	0.0918	5.4770	5.6092
7	5.4527	5.5538	0.0367	5.5629	5.6640
8	5.5681	5.6215	0.0147	5.6122	5.6656
9	5.6250	5.6411	0.0059	5.6428	5.6589
10	5.6497	5.6563	0.0024	5.6568	5.6635

The value of the objectives as the iterative process evolves is depicted below.



Note that the starting point for both trajectories is from the nadir point. From there it moves toward the projected aspiration point.

4.2 Starting the Search from the Ideal Solution

Next, using our Pareto Race approach, we demonstrate how to start from an ideal solution.

The ideal solution for the problem is (7, 8). An arbitrary nondominated solution in decision space is (5, 7). Thus, a possible formulation for starting the search from the ideal solution is as follows:

$$\begin{array}{rcll}
 \text{Max} & x_1 & + 2u & \geq 7 \\
 \text{Max} & & x_2 + u & \geq 8 \\
 \text{s.t.} & & u & = 1 \\
 & -x_1 + 2x_2 & & \leq 14 \\
 & x_1 + 3x_2 & & \leq 26 \\
 & 2x_1 + x_2 & & \leq 17 \\
 & x_1 - x_2 & & \leq 4 \\
 & x_1 + x_2 & & \geq 9 \\
 & x_1, x_2, u & & \geq 0
 \end{array}$$

Starting from the ideal solution ($u = 1$) (see, Table 4.1, col. "I"), the DM is free to search the nondominated frontier of the modified problem, which now consists of the ideal solution as well. In Figure 4.1 we illustrate one possible search path. The corresponding numerical values are given in Table 4.2. Assume that the DM first tries to reach values 6.5 for both objectives by varying the reference direction and the controlling parameter u .

After arriving at this solution, he/she may be willing to try to keep the first objective at this level and is ready to accept the drop in the value of the second objective down to 5.5. He will recognize that this point is not feasible ($u > 0$). After reconsidering the current situation, he/she accepts the decrease in the value of the first objective, but remains the value of the second objective at the current level. Now he can reach a feasible nondominated point (5.75, 5.50) ($u = 0$).

Fig. 4.1, Illustration of the Search Path Starting from the Ideal Solution

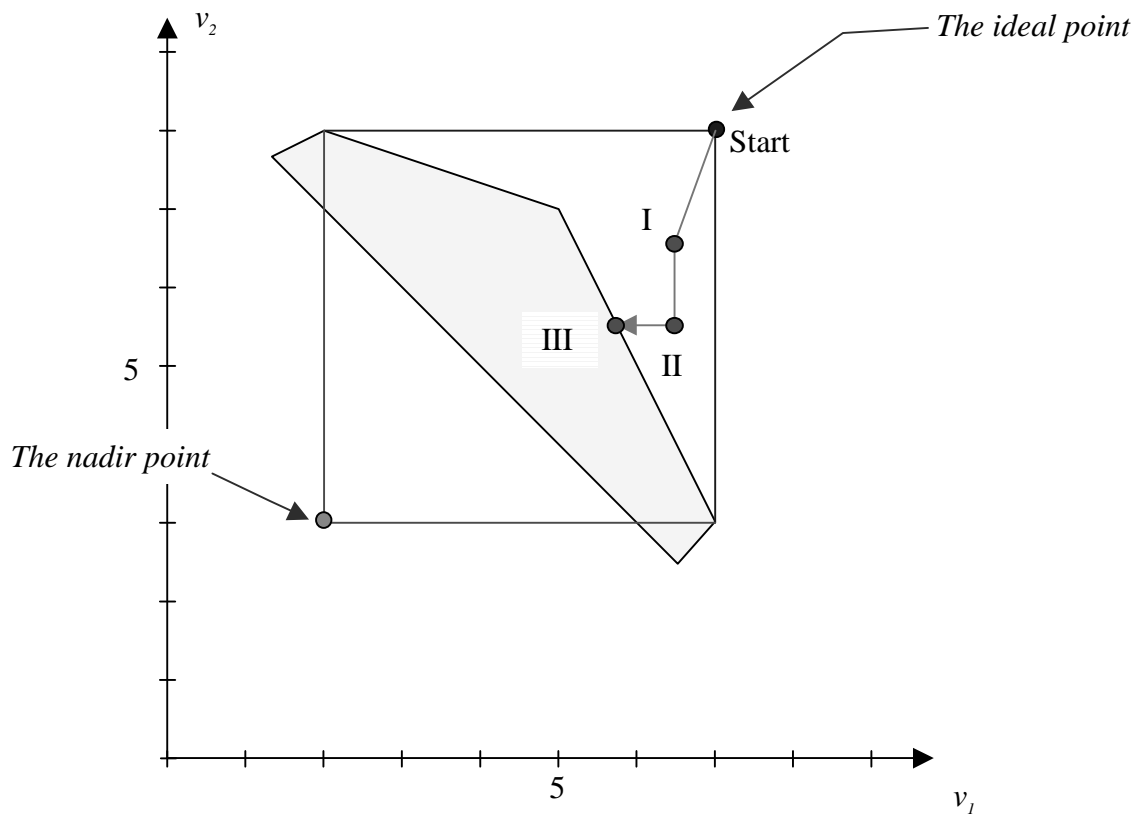


Table 4.2: Numerical Values of a Possible Search Path

		Solutions		
	Start	I	II	III
Obj 1 (x_1)	7	6.50	6.50	5.75
Obj 2 (x_2)	8	6.50	5.50	5.50
Row ₁	9	6.50	4.50	5.25
Row ₂	26	23.50	21.50	22.25
Row ₃	17	17.00	17.00	17.00
Row ₄	-2	-0.50	0.70	0.25
Row ₅	12	11.50	11.10	11.25
x_1	5	5.50	5.90	5.75
x_2	7	6.00	5.20	5.50
u	1	0.50	0.30	0.00

Parameter u can also be defined as an objective to be minimized. This works when all coefficients of parameters u on the objective rows are not negative, and at least one is strictly positive.

5. Concluding Remarks

We develop in this paper a method, which makes it possible to start from any point in criterion/objective space. The starting solution can be a feasible or an infeasible solution in decision space. The basic idea is simple. We first choose any solution in criterion space and find a corresponding point in a decision space. Usually, the point does not satisfy the original constraints, but augmenting the problem in a way explained in the paper, we derive an initial feasible solution. This artificial feasibility is controlled by one nonnegative parameter to be minimized. The parameter can be handled explicitly like any other criterion or it can be embedded into the formulation as a hidden parameter, which is finally is forced to zero. When it is used as one of the objectives, it is the responsibility of the user to ensure that the final value of the parameter is zero. Numerical examples demonstrate the use of the proposed starting approaches.

References

1. A. Arbel, *Exploring Interior Point Linear Programming: Algorithms and Software*, MIT Press, Cambridge, MA, 1993.
2. A. Arbel, "An Interior Multiobjective Linear Programming Algorithm," *Computers and Operations Research*, Vol. 20, No. 7, 1993, pp. 723-735.

3. A. Arbel, and S.S. Oren, "Generating Interior Search Directions for Multiobjective Linear Programming," *Journal of Multicriteria Decision Analysis*, Vol. 2, No. 2, August 1993, pp. 73-86.
4. A. Arbel, "Anchoring Points and Cones of Opportunities in Interior Multiobjective Linear Programming Problems," *Journal of the Operational Research Society*, Vol. 45, No. 1, January 1994, pp. 83-96.
5. A. Arbel, "A Multiobjective Interior Primal-Dual Linear Programming Algorithm," *Computers and Operation Research*, Vol. 21, No. 4, 1994, pp. 433-445.
6. A. Arbel, "Using Sequential Generation of Anchoring Points in an Interior Multiobjective Primal-Dual Linear Programming Algorithm," *Archives of Control Sciences*, Vol. 2, No. 1-2, 1993, pp. 5-21.
7. A. Arbel, "An Interior Multiple Objective Primal-Dual Linear Programming Algorithm using Efficient Anchoring Points," *Journal of the Operational Research Society*, Vol. 46, 1995, pp. 1121-1132..
8. A. Arbel, and P. Korhonen, "Generating Interior Search Directions in Multiple Objective Linear Programming Problems Using Aspiration Levels", in: *Multicriteria Analysis*, (J. Climaco - ed.), Springer Verlag, 1995, pp.146-156.
9. A. Arbel, and P. Korhonen, "Using Aspiration Levels in an Interactive Interior Multiobjective Linear Programming Algorithm," *European Journal of Operational Research*, Vol. 89, 1996, pp. 193-201.
10. A. Arbel, and P. Korhonen, "Using Aspiration Levels in an Interior Primal-Dual Multiobjective Linear Programming Algorithm," *Journal of Multicriteria Decision Analysis*, Vol. 5, 1996, pp. 61-71.
11. E.R. Barnes, "A variation on Karmarkar algorithm for solving linear programming problems," *Mathematical Programming*, Vol. 36, 1986, pp. 174-182.
12. V. Chankong, and Y.Y. Haimes, (1983), *Multiobjective Decision Making: Theory and Methodology*, North-Holland.
13. A.M. Geoffrion, J.S. Dyer, and A. Feinberg, "An interactive Approach for Multi-Criterion Optimization, with an Application to the Operations of an Academic Department," *Management Science*, Vol. 19, No. 4, 1972, pp. 357-368.
14. M. Halme, and P. Korhonen, "Nondominated Tradeoffs and Termination in Interactive Multiple Objective Linear Programming", in Lockett, A.G. and Islei, G. (Eds.): *Improving Decision Making in Organizations*, Springer-Verlag, 1989, pp. 410-423.
15. N.K. Karmarkar, "A new polynomial time algorithm for linear programming," *Combinatorica*, Vol. 4, 1984, pp. 373-395
16. P. Korhonen, and J. Wallenius, "A Pareto Race," *Naval Research Logistics*, Vol. 35, 1988, pp. 615-623.
17. P. Korhonen, and J. Laakso, "A Visual Interactive Method for Solving the Multiple Criteria Problem," *European Journal of Operational Research*, Vol. 24, 1986, pp. 277-287.
18. P. Korhonen, and J. Wallenius, "A Multiple Objective Linear Programming Decision Support System," *Decision Support Systems*, Vol. 6, 1990, pp. 243 - 251.

19. W.S. Shin, A. Ravindran, "Interactive Multiobjective Optimization Survey I - Continuous Case," *Computers and Operations Research*, Vol. 18, 1991, 97-114.
20. R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley, New York, 1986.
21. R.J. Vanderbei, M.S. Meketon, and B.A. Freedman, "A modification of Karmarkar's linear programming algorithm," *Algorithmica*, Vol. 1, 1986, 395-407.
22. R. J. Vanderbei, *Linear Programming*, Kluwer, 1996.
23. A. Wierzbicki, "The Use of Reference Objectives in Multiobjective Optimization," in *Multiple Criteria Decision Making, Theory and Application*, G. Fandel and T. Gal, (Eds.), Springer-Verlag, Berlin, 1980, 468-486.
24. A. Wierzbicki, "On the Completeness and Constructiveness of Parametric Characterizations to Vector Optimization Problems," *OR-Spektrum* **8**, 73-87 (1986).
25. A. Wierzbicki, "The Use of Reference Objectives in Multiobjective Optimization," in *Multiple Criteria Decision Making, Theory and Application*, G. Fandel and T. Gal, (Eds.), Springer-Verlag, Berlin, 468-486 (1980)22.