JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

3/2003

Decision support for telecommunications and information society

Special issue edited by Wiesław Traczyk

Editorial Board

Editorial Advisory Board

ISSN 1509-4553

Circulation: 300 copies

Sowa - Druk na życzenie, www.sowadruk.pl, tel. 022 431-81-40

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

The Second International Conference on *Decision Support for Telecommunications and Information Society DSTIS-2002*, organised by the National Institute of Telecommunications in Warsaw (5th – 7th September 2002), assembled researchers working in several modern and important fields of telecommunications and computer science. Some papers submitted to this conference are presented at this issue.

Domains of *decision support and optimisation* are represented here by five papers. One of them is devoted to the procedure for finding the optimal decision in the case when information about the state of nature and utility functions is imprecisely defined. The next paper shows how the class of reference point methods can be reduced to the class of weight methods, so it is possible to implement both classes in the same technical framework. Another approach to reference point and vector optimisation applies evolutionary algorithms, using them in an interactive mode. The problem of allocation of limited resources among competing activities is also presented, showing how to achieve the best overall performances with fair treatment of all activities. The last paper of this group describes two ideas of modifying projection methods for the case of nonlinear optimisation.

The set of papers devoted to *network modelling and design* starts from two descriptions of new developments of a multiple objective dynamic routing method for multiexchange networks, that enables numerical calculation of global network performance parameters. Dynamic routing problems are also considered in the next work which shows that they may be interpreted as optimal control or regulation problems, and solved with the use of well known methods. Afterwards the topological network design is described as optimisation problem, determining network structure and allocation pattern that would minimise the cost of the network, using various algorithms. Another optimisation method is proposed for reconfiguration of the IP over optical network, in order to balance the load of network elements.

Tools for intelligent systems are presented in four papers. A theory is described that defines contextual probability function satisfying all axioms of probability and therefore generalising the classical probability theory. The author of rough set theory considers some relationship between Bayes theory and rough sets, using flow graphs and decision rules. The next elaboration proposes modified training algorithms for artificial neural networks, in which the initial weight configuration remains unchanged throughout the training process. Qualitative features of decision problems can be represented by classical influence diagrams but their usefulness is enlarged if time is introduced as special parameter, what is described in appropriate paper.

Data mining is the subject of the last group of conference papers. The first paper presents a method of computerised generation of hypotheses (as a kind of association rule) based on given data, represented as a matrix of objects and attributes. The second one suggests several models of data mining operations, based on the concept of information system. The last paper describes the new form of quantitative and multi-dimensional association rules that may be used for cellular network planning.

Decision support is needed in almost all domains of human activities and presented here set of papers shows some possibilities.

Wiesław Traczyk
Guest Editor

Paper

Possibilistic approach to Bayes decisions

Olgierd Hryniewicz

Abstract — The decision problems are considered when the prior probabilistic information about the state of nature and decision maker's utility function are imprecisely defined. In such a case the risks (or the expected utility) of considered decisions are also imprecisely defined. We propose two-step procedure for finding the optimal decision. First, we order possible decisions using the λ-average ranking method by Campos and Gonzalez [1]. Then we use possibilistic possibility of dominance and necessity of strict dominance indices proposed by Dubois and Prade [3] for the comparison of consequences of the most promising solutions.

Keywords — optimal decisions, imprecise information, fuzzy risks, possibility indices.

1. Introduction

In decision making we deal with uncertainties related to an unknown state of nature. The behaviour of a decision maker may be described as a kind of game between him and a fictitious player who may be called "nature" or "chance". Decisions made by a decision maker are rational if they are derived from his knowledge about nature's behaviour and the knowledge of the consequences of his decisions. Mathematical theories of decision making are known for more than fifty years and are based on probabilistic models of nature's behaviour and utility functions. Their basic ideas and main results were published in a famous book by Raiffa and Schlaifer [6] that has been recently republished by J. Wiley & Sons. In the classical models of decision making it is assumed that the decision maker knows the joint probability distribution of all possible states of the nature and all possible results of experiments which provide him with some knowledge about the actual state of the nature. Moreover, it is assumed that there exists a precisely defined utility function which assigns decision maker's utility related to all possible pairs: decision and state of the nature. These premises have been recently relaxed by assuming that some parameters of decision models may be defined only with a certain degree of precision. As a consequence of such more general assumptions we arrive at mathematical models of imprecise risks.

In this paper we present some results obtained under the assumption of the existence of imprecisely defined risks. In Section 2 we present a mathematical model of decision making in the presence of imprecisely defined probabilistic prior information about the possible states of the nature and imprecisely defined utility functions. A lack of the precision we describe in the language of the fuzzy sets theory. We propose to find the best decisions by the defuzzification of imprecisely defined expected risks. For this purpose we propose the use of the defuzzification method proposed by Campos and Gonzalez [1]. This method allows the user to take into account his attitude, i.e. his level of optimism (or pessimism). In Section 3 we propose a possibilistic method for the comparison of different decisions. By applying this method we provide the user with addition information about the real differences between the consequences of his decisions. In this comparison we take into account the impact of imprecise input information on the decision making.

2. Mathematical model and the choice of optimal decisions

There exist different methods for modelling decisions. In this paper we adopt the approach described in a general form by Raiffa and Schlaifer [6]. The model proposed by Raiffa and Schlaifer consists of two parts: one part is dedicated to the choice of the final decision, and the second part is dedicated to the choice of the experiment whose ultimate goal is to provide the decision maker with some information about the actual state of nature. According to this model the decision maker can specify the following data defining his decision problem:

1. Space of terminal decisions (acts): $A = \{a\}$.
2. State space: $\Theta = \{\theta\}$.
3. Family of experiments: $E = \{e\}$.
4. Sample space: $Z = \{z\}$.
5. Utility function: $u(\cdot,\cdot,\cdot,\cdot,)$ on $E \times Z \times A \times \Theta$.

The decision maker evaluates an utility $u(e,z,a,\theta)$ of making a particular experiment e, obtaining the result of this experiment z, taking a decision a in the case when the true state of nature is θ. In order to find appropriate (hopefully optimal) decisions the decision maker has also to specify a joint probability measure $P_{\theta,z}(\cdot,\cdot|e)$ for a Cartesian product $\Theta \times Z$. The knowledge of this probability measure means that we know the joint probability distribution of observation z in an experiment e when the *random* state of nature is described by θ. Knowing this joint probability distribution we can calculate some important marginal and conditional probability distributions. In particular, for a given experiment e we are usually interested in three distributions:

1. The marginal distribution on the state space Θ describing our *prior* information about possible states of nature. We assume that this distribution does not depend on e.

2. The conditional distribution on the sample space Z for a given state of nature θ.

3. The conditional distribution on the state space Θ for a given result of the experiment z describing our *posterior* information about possible states of nature.

Note, that we may know only these particular distributions as their knowledge is equivalent to the knowledge of the joint probability distribution on $\Theta \times Z$.

Let us consider the simplest case of the general model when there is no experiment e. In such a case the only information we need is the probability distribution $\pi(\theta)$ defined on the state space Θ. We call this distribution *the prior distribution* of the parameter (parameters) describing the unknown state of nature. If we know the utility function $u(a,\theta)$ defined on $A \times \Theta$ we may calculate *the expected utility* assigned to a particular action (decision) a from the simple formula

$$u(a) = \int_{\Theta} u(a,\theta)\,\pi(\theta)\,\mathrm{d}\theta\,. \tag{1}$$

If we use *a loss function* $L(a,\theta)$ for the description of potential consequences of taking decision a we may calculate *the expected* loss (usually called *a risk*) from an equivalent formula

$$\rho(a) = \int_{\Theta} L(a,\theta)\,\pi(\theta)\,\mathrm{d}\theta\,. \tag{2}$$

Having the expected utilities for *all* possible decisions we can find the optimal one which is related to the maximal expected utility (or the minimal risk). This procedure is in principle very simple. However, in many practical cases (when the number of possible decisions is sufficiently large) it may require the use of sophisticated optimisation methods.

When the decision maker has an additional information about the state of nature in a form of observations $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ of a random vector described by a probability distribution $f(\mathbf{z},\theta)$ we may calculate *the expected utility* assigned to a particular action (decision) a from a formula

$$u(a,\mathbf{z}) = \int_{\Theta} u(a,\theta)\,g(\theta|\mathbf{z})\,\mathrm{d}\theta\,, \tag{3}$$

where

$$g(\theta|\mathbf{z}) = \frac{f(\mathbf{z}|\theta)\,\pi(\theta)}{\int\limits_{\Theta} f(\mathbf{z}|\theta)\,\pi(\theta)\,\mathrm{d}\theta} \tag{4}$$

is the posterior distribution of the parameter θ which describes the state of nature. In such a case the expected utility attributed to each decision is calculated from

$$u(a|\mathbf{z}) = \int_{\Theta} u(a,\theta)\,g(\theta|\mathbf{z})\,\mathrm{d}\theta\,, \tag{5}$$

and the respective risk from the formula

$$\rho(a|\mathbf{z}) = \int_{\Theta} L(a,\theta)\,g(\theta|\mathbf{z})\,\mathrm{d}\theta\,. \tag{6}$$

The procedure for finding the optimal decision is exactly the same as in the case described previously.

Suppose now that the prior distribution $\pi(\theta)$ and the loss (or utility) $L(a,\theta)$ are functions of parameters ζ and ψ, respectively, and that these parameters are known only imprecisely. Let us assume that our imprecise knowledge about possible values of ζ and ψ is represented by fuzzy sets $\widetilde{\zeta}$ and $\widetilde{\psi}$, respectively. A fuzzy set $\widetilde{X}$ is defined using the membership function $\mu_{\widetilde{X}}(x)$ which in the considered context of this paper describes the grade of possibility that a fuzzy parameter, say $\widetilde{X}$, has a specified value of x. Each fuzzy set may be also represented by its α-cuts defined as ordinary sets

$$X^{\alpha} = \left\{x \in R : \mu_{\widetilde{X}}(x) \geq \alpha\right\}, \quad 0 \leq \alpha \leq 1\,. \tag{7}$$

From the representation theorem for fuzzy sets we know that each membership function may be equivalently expressed as

$$\mu_{\widetilde{X}}(x) = \sup\left\{\alpha I_{\widetilde{X}^{\alpha}}(x) : \alpha \in [0,1]\right\}. \tag{8}$$

Now let us assume that imprecisely known parameters ζ and ψ are represented by their α-cuts, and that these α-cuts are given in a form of closed intervals $\left[\zeta_L^{\alpha}, \zeta_U^{\alpha}\right]$ and $\left[\psi_L^{\alpha}, \psi_U^{\alpha}\right]$, respectively. The knowledge of these α-cuts let us calculate fuzzy equivalents of the expected utility or the expected loss (risk). To make the presentation simple we assume that decision are based exclusively on the knowledge of the prior distribution $\pi(\theta)$ and the loss function $L(a,\theta)$. As these functions are the functions of imprecise fuzzy parameters, they are also fuzzy, and may be denoted as $\widetilde{\pi}(\theta;\widetilde{\zeta})$ and $\widetilde{L}(a,\theta;\widetilde{\psi})$, respectively.

Now, let us rewrite formula (2) as

$$\widetilde{\rho}(a) = \int_{\Theta} \widetilde{L}(a,\theta;\widetilde{\zeta})\,\widetilde{\pi}(\theta;\widetilde{\psi})\mathrm{d}\theta\,. \tag{9}$$

The risk calculated from formula (9) is now an imprecisely defined *fuzzy number* whose membership function may be calculated using Zadeh's extension principle (see Klir and Yuan [5], or any other textbook on fuzzy sets for

a reference). It is easy to show that the fuzzy risk $\widetilde{\rho}(a)$ is now represented by its α-cuts $\left[\rho_L^\alpha, \rho_U^\alpha\right]$, where

$$\rho_L^\alpha = \inf_{\substack{\zeta \in \left[\zeta_L^\alpha, \zeta_U^\alpha\right] \\ \psi \in \left[\psi_L^\alpha, \psi_U^\alpha\right]}} \widetilde{\rho}(a) \tag{10}$$

and

$$\rho_U^\alpha = \sup_{\substack{\zeta \in \left[\zeta_L^\alpha, \zeta_U^\alpha\right] \\ \psi \in \left[\psi_L^\alpha, \psi_U^\alpha\right]}} \widetilde{\rho}(a). \tag{11}$$

Thus, for every possible decision a we may find a fuzzy risk $\widetilde{\rho}(a)$ or a fuzzy expected utility $\widetilde{u}(a)$ which may be calculated in the same way. Moreover, if there exists an additional information in the form of observations $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ we may use exactly the same procedure in order to fuzzify the expected utility given by formula (5) and the risk given by formula (6). Note however, that in this case the respective calculations (especially for the fuzzy posterior distribution) may be much more complicated.

In contrast to the non-fuzzy (crisp) case the univocal optimal solution of the decision problem for imprecisely defined input parameters does not exist. It stems from the fact that fuzzy sets are not naturally ordered. Thus, in general, it is not possible to indicate the decision with lowest risk (or the highest expected utility). In order to do this we must apply one of the many proposed in literature ranking methods.

There are many methods for ranking fuzzy numbers that are based on different defuzzification methods. Gil and Lopez-Diaz [4] have noticed that the λ-average ranking method proposed by Campos and Gonzalez [1] is especially useful in decision making. Let $\widetilde{X}$ be a fuzzy number (fuzzy set) described by the set of its α-cuts $\left[X_L^\alpha, X_U^\alpha\right]$, and S be an additive measure on $[0,1]$. Moreover, assume that the support of $\widetilde{X}$ is a closed interval. The λ-average value of such a fuzzy number $\widetilde{X}$ is defined by Campos and Gonzalez [1] as

$$V_S^\lambda(\widetilde{X}) = \int_0^1 \left[\lambda X_U^\alpha + (1-\lambda) X_L^\alpha\right] \mathrm{dS}(\alpha), \quad \lambda \in [0,1]. \tag{12}$$

In the case of continuous membership functions the integral in formula (12) is calculated with respect to $\mathrm{d}\alpha$. Thus, the λ-average value of $\widetilde{X}$ can be viewed as its defuzzified value.

The parameter λ in (12) is a subjective degree of decision maker's optimism (pessimism). In the case of fuzzy risks $\lambda = 0$ reflects his highest optimism as the minimal values of all α-cuts (representing the lowest possible risks) are taken into consideration. On the other hand, by taking $\lambda = 1$ the decision maker demonstrates his total pessimism, as only the maximal values of all α-cuts (representing the highest possible risks) are considered. In the case of fuzzy expected utilities the situation is reversed, i.e. $\lambda = 1$ represents decision maker's optimism, and $\lambda = 0$ reflects his total pessimism. If the decision maker takes $\lambda = 0.5$ his attitude may described as neutral. Thus, by varying the value of λ the decision maker is able to take into account the level of his optimism (pessimism) which may arise e.g. from having some additional information that has not been reflected in the prior distribution. Some interesting features of the λ-average ranking method have been discussed in Gil and Lopez-Diaz [4].

Having a simple ranking method given by formula (12) we may calculate defuzzified values of fuzzy risks (expected utilities) related to all considered decisions. The optimal decision has the lowest defuzzified risk (or the highest defuzzified expected utility). Moreover, the decision maker can order all considered decisions with respect to their risks (or expected utilities).

3. Possibilistic analysis of optimal decisions

The procedure described in the previous section allows the decision maker to find the optimal decision. It has to be noted, however, that the ranking method gives only a partial information about the differences between competitive decisions. Therefore, we claim that it is necessary to perform an additional analysis that provides the decision maker with an additional information about the considered decisions. Such an analysis is especially interesting when the consequences of different decisions are similar, and when other decision maker's preferences, not reflected in the optimisation model, exist. To analyse the consequences of different decisions we propose to use the methodology known from the theory of possibility, namely the *possibility of dominance* and *necessity of strict dominance* indices proposed by Dubois and Prade [3].

For two fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ the *possibility of dominance (PD)* index is calculated from the formula

$$PD = Poss(\widetilde{A} \geq \widetilde{B}) = \sup_{x,y:x \geq y} \min\left\{\mu_{\widetilde{A}}(x), \mu_{\widetilde{B}}(y)\right\}. \tag{13}$$

The PD index gives the measure of *possibility* that the fuzzy number $\widetilde{A}$ is not smaller than the fuzzy number $\widetilde{B}$. Positive value of this index tells the decision maker that there exists even slightly evidence that the relation $\widetilde{A} \geq \widetilde{B}$ is true.

The degree of *conviction* that the relation $\widetilde{A} > \widetilde{B}$ is true is reflected by the *necessity of strict dominance (NSD)* index defined as

$$NSD = Ness(\widetilde{A} > \widetilde{B}) =$$

$$= 1 - \sup_{x,y:x \leq y} \min\left\{\mu_{\widetilde{A}}(x), \mu_{\widetilde{B}}(y)\right\} = 1 - Poss(\widetilde{B} \geq \widetilde{A}). \tag{14}$$

The *NSD* index gives the measure of *necessity* that the fuzzy number $\widetilde{A}$ greater than the fuzzy number $\widetilde{B}$. Positive value of this index tells the decision maker that there exists rather strong evidence that the relation $\widetilde{A} > \widetilde{B}$ is true.

According to Cutell and Montero [2] we may use the *PD* and *NSD* indices to evaluate mutual relationship between two considered decisions. Let us describe the evaluation procedure for two decisions a_1 and a_2 with associated fuzzy risks $\widetilde{\rho}(a_1)$ and $\widetilde{\rho}(a_2)$, respectively. The value of $NSD = Nec\big(\widetilde{\rho}(a_1) > \widetilde{\rho}(a_2)\big)$ indicates that extend decision a_1 is inferior in comparison to decision a_2. On the other hand, $1-PD = 1-Poss\big(\widetilde{\rho}(a_1) \geq \widetilde{\rho}(a_2)\big)$ indicates that extend decision a_1 might be considered superior in comparison to decision a_2. If instead of fuzzy risks we compare fuzzy expected utilities the conclusions are reversed, i.e. the value of $NSD - Nec\left(\widetilde{u}\,(a_1) > \widetilde{u}\,(a_2)\right)$ indicates that extend decision a_1 is superior in comparison to decision a_2, etc. The value of $PD - NSD$ may be viewed upon as the measure of *indifference* between the consequences of the considered decisions.

If the decision maker has the ordered sequence of his possible decisions he should always consider a possibility of performing pairwise comparisons between the best two (or more) competitive solutions. High values of the indifference indices reveal that the consequences of considered decisions are rather insignificant due to the lack of precision of the optimisation model. In such a case the decision maker may use some additional criteria for choosing an appropriate decision. This is also the signal that it is advisable to make the optimisation model more precise.

4. Decisions with two possible outcomes – a numerical example

Let us consider the simplest situation when each action from a set of alternatives $\{a_1, \ldots, a_M\}$ leads to two possible outcomes $w^{(m)}$, $m = 1, \ldots, M$ and $v^{(m)}$, $m = 1, \ldots, M$, respectively. The outcome $w^{(m)}$ appears with probability $p^{(m)}$, $m = 1, \ldots, M$, and the outcome $v^{(m)}$ appears with probability $1-p^{(m)}$. Suppose that the expected outcome is equivalent to the expected utility. Thus the expected utility associated with the action a_m is given by

$$u^{(m)} = p^{(m)} w^{(m)} + \left(1 - p^{(m)}\right) v^{(m)}, \quad m = 1, \ldots, M. \tag{15}$$

In this way, the optimal action is a such one which maximises Eq. (15) when the outcomes are given in terms of profits or minimises Eq. (15) when outcomes are expressed in terms of losses.

Let us assume that all information about the outcomes and respective probabilities are imprecise and are given by fuzzy numbers described by a trapezoidal membership functions. In general, any trapezoidal membership function of a fuzzy number $\widetilde{X} = \widetilde{X}(x_l, x_{0,l}, x_{0,r}, x_r)$ is described by the following formula:

$$\mu_{\widetilde{X}}(x) = \begin{cases} 0 & x \leq x_l \\ \dfrac{x - x_l}{x_{0,l} - x_l} & x_l < x \leq x_{0,l} \\ 1 & x_{0,l} < x \leq x_{0,r} \\ \dfrac{x_r - x}{x_r - x_{0,r}} & x_{0,r} < x \leq x_r \\ 0 & x > x_r \end{cases}. \tag{16}$$

The α-cuts of the fuzzy number described by the membership function given by formula (16) have the following form: $\big(x_l + \alpha(x_{0,l} - x_l),\, x_r - \alpha(x_r - x_{0,r})\big)$.

Denote by $\widetilde{w}^{(m)}$, $m = 1, \ldots, M$, and $\widetilde{v}^{(m)}$, $m = 1, \ldots, M$ the fuzzy counterparts of the crisp outcomes $w^{(m)}$ and $v^{(m)}$, respectively. Moreover, let $\widetilde{p}^{(m)}$, $m = 1, \ldots, M$ be the fuzzy counterpart of the crisp probability $p^{(m)}$. Assume now, that for each α-cut we have $w_{0,l}^{(m)} > v_{0,r}^{(m)}$. It means that despite their imprecision both possible outcomes are separated. When this assumption does not hold we have either to assume that the outcomes are interactive in a special way or to assume that they are indistinguishable to some extent. In both cases, this leads to severe complication of the optimisation procedure.

Now, we can define a fuzzy expected utility as follows

$$\widetilde{u}^{(m)} = \widetilde{p}^{(m)} \widetilde{w}^{(m)} + (1 - \widetilde{p}^{(m)}) \widetilde{v}^{(m)},\ m = 1, \ldots, M. \tag{17}$$

Using the extension principle of Zadeh we can find the membership function of the fuzzy expected utility $\widetilde{u}^{(m)}$, $m = 1, \ldots, M$. In further calculations in order to simplify the notation we omit the upper index (m) that indicates the undertaken action. Denote by $\big(u_l(\alpha), u_r(\alpha)\big)$ the α-cut of $\widetilde{u}$. By simple calculations we can show that

$$\begin{aligned} u_l(\alpha) = {} & p_l w_l + (1 - p_l) v_l + \alpha\big[(p_{0,l} - p_l) w_l + \\ & + p_l (w_{0,l} - w_l) - (p_{0,l} - p_l) v_l + (1 - p_l)(v_{0,l} - v_l)\big] + \\ & + \alpha^2 \big[(p_{0,l} - p_l)(w_{0,l} - w_l) - (p_{0,l} - p_l)(v_{0,l} - v_l)\big] \end{aligned} \tag{18}$$

and

$$\begin{aligned} u_r(\alpha) = {} & p_r w_r + (1 - p_r) v_r + \alpha\big[(p_r - p_{0,r}) v_r + \\ & - (1 - p_r)(v_r - v_{r,0}) - p_r (w_r - w_{0,r}) + (p_r - p_{0,r}) w_r\big] + \\ & + \alpha^2 \big[(p_r - p_{0,r})(w_r - w_{0,r}) - (p_r - p_{0,r})(v_r - v_{0,r})\big]. \end{aligned} \tag{19}$$

The λ-average value of $\widetilde{u}$ calculated from formula (12) is now given by

$$\begin{aligned} V^{\lambda}(\widetilde{u}) = {} & \lambda \Big[\frac{1}{2}(v_r + v_{0,r}) + \frac{1}{3}\big(p_r (w_r - v_r) + \\ & + p_{0,r}(w_{0,r} - v_{0,r})\big) + \frac{1}{6}\big(p_r (w_{0,r} - v_{0,r}) + p_{0,r}(w_r - v_r)\big)\Big] + \\ & + (1 - \lambda)\Big[\frac{1}{2}(v_l + v_{0,l}) + \frac{1}{3}\big(p_l (w_l - v_l) + p_{0,l}(w_{0,l} - v_{0,l})\big) + \\ & + \frac{1}{6}\big(p_l (w_{0,l} - v_{0,l}) + p_{0,l}(w_l - v_l)\big)\Big]. \end{aligned} \tag{20}$$

Having λ-average values of the fuzzy expected utilities for all considered actions we can find the optimal one that has the maximal value of $V^{\lambda}\left(\widetilde{u}^{(m)}\right)$.

Let us assume that all considered actions are numbered in such a way that $V^{\lambda}\left(\widetilde{u}^{(1)}\right) \geq V^{\lambda}\left(\widetilde{u}^{(2)}\right) \geq \ldots V^{\lambda}\left(\widetilde{u}^{(M)}\right)$. The next step of the possibilistic analysis consists in the comparison of fuzzy expected utilities $\widetilde{u}^{(1)}$ and $\widetilde{u}^{(2)}$. The analysis of Eq. (14) shows that $NSD\left(\widetilde{u}^{(1)} > \widetilde{u}^{(2)}\right) > 0$ if the relation holds $u_r^{(2)}(1) < u_l^{(1)}(1)$. In such a case $NSD\left(\widetilde{u}^{(1)} > \widetilde{u}^{(2)}\right) = 1 - \alpha^*$, where α^* is the solution of the equation

$$u_r^{(2)}(\alpha) = u_l^{(1)}(\alpha). \tag{21}$$

Let

$$x_1 = p_l w_l + (1 - p_l) v_l, \tag{22}$$

$$x_2 = (p_{0,l} - p_l) w_l + p_l (w_{0,l} - w_l) + \\ -(p_{0,l} - p_l) v_l + (1 - p_l)(v_{0,l} - v_l), \tag{23}$$

$$x_3 = (p_{0,l} - p_l)(w_{0,l} - w_l) - (p_{0,l} - p_l)(v_{0,l} - v_l), \tag{24}$$

$$y_1 = p_r w_r + (1 - p_r) v_r, \tag{25}$$

$$y_2 = (p_r - p_{0,r}) v_r - (1 - p_r)(v_r - v_{r,0}) + \\ -p_r (w_r - w_{0,r}) + (p_r - p_{0,r}) w_r, \tag{26}$$

$$y_3 = (p_r - p_{0,r})(w_r - w_{0,r}) - (p_r - p_{0,r}), \tag{27}$$

$$A_1 = x_1 - y_1, \tag{28}$$

$$A_2 = x_2 - y_2, \tag{29}$$

$$A_3 = x_3 - y_3. \tag{30}$$

Hence, the solution of Eq. (21) is given by

$$\alpha^* = \begin{cases} \dfrac{-A_2 + \sqrt{A_2^2 - 4A_1A_3}}{2A_3} & \text{if} \quad A_3 \neq 0 \\ -A_1/A_2 & \text{if} \quad a_3 = 0 \end{cases}. \tag{31}$$

To illustrate these theoretical results let us consider a numerical example. Suppose, that there are four possible actions described by the following sets of their fuzzy parameters:

- action a_1:
  $\widetilde{p}^{(1)} = \widetilde{p}^{(1)}(0.2; 0.25; 0.3; 0.35)$,
  $\widetilde{w}^{(1)} = \widetilde{w}^{(1)}(80; 90; 100; 110)$,
  $\widetilde{v}^{(1)} = \widetilde{v}^{(1)}(20; 25; 30; 35)$;
- action a_2:
  $\widetilde{p}^{(2)} = \widetilde{p}^{(2)}(0.2; 0.25; 0.25; 0.25)$,
  $\widetilde{w}^{(2)} = \widetilde{w}^{(2)}(60; 70; 80; 90)$,
  $\widetilde{v}^{(2)} = \widetilde{v}^{(2)}(15; 20; 20; 25)$;
- action a_3:
  $\widetilde{p}^{(3)} = \widetilde{p}^{(3)}(0.2; 0.25; 0.25; 0.3)$,
  $\widetilde{w}^{(3)} = \widetilde{w}^{(3)}(60; 70; 80; 90)$,
  $\widetilde{v}^{(3)} = \widetilde{v}^{(3)}(-10; 20; 20; 25)$;
- action a_4:
  $\widetilde{p}^{(4)} = \widetilde{p}^{(4)}(0.2; 0.2; 0.2; 0.4)$,
  $\widetilde{w}^{(4)} = \widetilde{w}^{(4)}(30; 60; 60; 70)$,
  $\widetilde{v}^{(4)} = \widetilde{v}^{(4)}(-10; 0; 10; 20)$.

The expected utilities associated with each action are given as fuzzy numbers whose λ-averages calculated according to Eq. (20) are the following (for $\lambda = 0.5$, i.e. for a neutral decision maker):

$$V^{\lambda}\left(\widetilde{u}^{(1)}\right) = 46.33; \qquad V^{\lambda}\left(\widetilde{u}^{(2)}\right) = 33.17;$$
$$V^{\lambda}\left(\widetilde{u}^{(3)}\right) = 29.06; \qquad V^{\lambda}\left(\widetilde{u}^{(4)}\right) = 17.5.$$

Thus, action a_1 is visibly better than the others. However, if we compare the fuzzy utility of a_1 with the fuzzy utility of the second best action a_2 we arrive at the following results. For a_1 from Eqs. (21)–(24) we get: $x_1 = 32$, $x_2 = 9$, $x_3 = 0.25$, and for a_2 from Eqs. (25)–(27) we get: $y_1 = = 41.25$, $y_2 = -6.25$, $y_3 = 0$. Hence, from Eqs. (28)–(30) we get: $A_1 = -9.25$, $A_2 = 15.25$, $A_3 = 0.25$. Thus, from Eq. (31) we obtain $\alpha^* = 0.6$, and the necessity of strict dominance index is the following $NSD\left(\widetilde{u}^{(1)} > \widetilde{u}^{(2)}\right) = 0.4$. It means that there exists only limited necessity that a_1 is better than a_2, and – to some extent – their results are indistinguishable. This is especially true, when the parameters of the decision model come from different sources.

5. Conclusions

In the paper we present a generalisation of a classical Bayes decision model. In this generalised model we assume that all input parameters describing prior probabilities, costs, and statistical data may be expressed in an imprecise way. If we apply a fuzzy description of those vague data we arrive at fuzzy risks or fuzzy expected utilities associated with each possible action (decision). Unfortunately, a method for an unique ordering of fuzzy numbers does not exist. Therefore, we propose to use the defuzzification method of Campos and Gonzalez [1] in order to find two possibly best actions. Imprecise consequences of these decisions we compare using possibility and necessity indices. This approach

gives us a better insight in the process of decision making. We illustrate the proposed procedure with a numerical example when each action (decision) may result with two possible outcomes.

References

[1] L. M. Campos and A. Gonzalez, "A subjective approach for ranking fuzzy numbers", *Fuzzy Sets Syst.*, vol. 29, pp. 145–153, 1989.

[2] V. Cutell and J. Montero, "An extension of the axioms of utility theory based on fuzzy rationality measures" in *Preference and Decisions under Incomplete Knowledge*, J. Fodor, P. De Baets, and P. Perny, Eds. Heidelberg: Physica-Verlag, 1999, pp. 33–50.

[3] D. Dubois and H. Prade, "Ranking fuzzy numbers in the setting of possibility theory", *Inform. Sci.*, vol. 30, pp. 184–244, 1983.

[4] M. A. Gil and M. Lopez-Diaz, "Fundamentals and Bayesian analyses of decision problems with fuzzy-valued utilities", *Int. J. Approx. Reason.*, vol. 15, pp. 203–224, 1996.

[5] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. New York: Prentice-Hall, 1995.

[6] H. Raiffa and R. Schlaifer, *Applied Statistical Decision Theory.* New York: Wiley, 2000.

Olgierd Hryniewicz is a Professor at the Polish Academy of Sciences, Director of the Systems Research Institute of PAS, and Head of the Laboratory of Stochastic Methods. He received his M.Sc. degree from Warsaw University of Technology (1970). The Ph.D. degree he received from the Institute of Management and Automatic Control of PAS (1976), and the D.Sc. degree from Cracow Academy of Economics (1986). Since 1996 he is a Professor at the Polish Academy of Sciences. He published over 150 books, papers, and reports on reliability, quality control and decision support systems. He is also a Professor of Warsaw School of Information Technology and Management.

e-mail: hryniewi@ibspan.waw.pl
Systems Research Institute
Polish Academy of Sciences
Newelska st 6
01-447 Warsaw, Poland

Paper

Weight versus reference point multiple criteria decision making methods – analogies and differences

Ignacy Kaliszewski

Abstract — In this work we shall be concerned with interactive multiple criteria decision making methods. We show how on the technical level the class of reference point methods can be reduced to the class of weight methods. Though methods from these two classes represent two different interactive decision making paradigms, the equivalence observed opens a way for a joint implementation of a pair of methods each representing a different class. This would establish a firm ground for systematic comparison of both classes of methods as well as for hybrid schemes mixing decisional tools specific for each class.

Keywords — multiple criteria decision making, weight methods, reference point methods.

1. Introduction

A rough taxonomy of interactive multiple criteria decision making (MCDM) methods distinguishes three major classes, namely *weight methods, reference point methods*, and *constraint methods*. All methods of these classes amount to a partial, decision maker (DM) guided search of the set of *efficient* decisions. The trichotomy is based on which elements are manipulated to capture DM's preferences: weights, reference points, or constraints. In this presentation we shall confine ourselves to the first two classes, which are believed to capture DM's preferences in a favourable manner.

In weight methods the DM articulates his partial preferences pointing to preferred decisions in pairwise comparisons. Partial preferences are translated next into relations expressed in terms of weights. In some methods weights are provided explicitly by the DM. In reference point methods the DM articulates his preferences by pointing to reference points which can be any elements of the space of criteria.

In weight methods *the set of weights* is systematically searched and reduced according to DM articulated preferences. The volume of the set of weights is a natural measure of progress and convergence of the decision making process. Reference point methods lack such a systemic convergence indicator.

The purpose of this paper is to show that in technical terms reference point methods can be reduced to weight methods. With such an interpretation provided it is possible to implement methods of these two classes in the same technical framework. This would establish a firm ground for systematic comparison of both classes of methods as well as for hybrid schemes mixing decisional tools specific for each class. Moreover, a convergence indicator is then available for either class of methods.

The plan of the paper is as follows. In Section 2 we recall all the relevant definitions and formulations. In Section 3 we recall characterizations of the set of properly efficient decisions, namely the characterization by weight manipulations and the characterization by reference point manipulations, and in Section 4 we recall how these characterizations are used in the two classes of MCDM methods considered. In Section 5 we show that under a restriction of reference point methods, weight methods and reference point methods are technically equivalent. In Section 6 we discuss practical significance of such an equivalence. Section 7 concludes.

2. Preliminaries

In the multiple criteria decision making framework a decision problem is formalized as follows:

$$\text{choose "}\textit{the most preferred}\text{" vector } f(x),\ x \in X_0 \subseteq X, \tag{1}$$

where X is the space of decisions, X_0 is the set of feasible decisions, $f : X \to \Re^k$ is the criteria map, where $f = (f_1,...,f_k)$ and $f_l : X \to \Re,\ l = 1,...,k$, are criteria functions. We assume that all criteria are of the type "better if more".

From the algorithmic point of view the above problem is ill-defined. As long as we do not know what "the most preferred" means precisely we are not in a position to propose a problem solving method. The only source of supplementary information to those already given in (1) can be the decision maker (DM). The underlying assumption of MCDM is that this information cannot be acquired from the DM at once.

A formal model for MCDM is offered by the *vector optimization problem*, namely

$$vmax\, f(x),\ x \in X_0 \subseteq X, \tag{2}$$

where *vmax* stands for the identification of all *efficient* decisions of X_0. This problem is well-defined which means

that under minor assumptions, satisfied in practical problems, the solution to (2) always exists.
Decisions are represented by their criteria values. With this in mind, from now on we shall be dealing with elements $f(x)$ of set $f(X_0)$ and for the sake of simplicity we shall use the notation

$$y = f(x) \text{ and } Z = f(X_0).$$

Elements of set Z we shall call *outcomes*. Under this convention, for given feasible decision x, $y_l = f_l(x)$ is the value of lth component of outcome $y = f(x)$. Thus, y_l is the value of lth criterion.
All properties of decisions we shall need throughout this paper can be defined in terms of outcomes. The notation x, X_0, $f(x)$, $f(X_0)$ has to be used only when one is to operationalize an implicit (i.e. in the form of constraints) feasible decision representation.
The element $\hat{y}$ representing the hypothetical decision which maximizes all objective functions, called *utopian* element, is calculated as

$$\hat{y}_l = max_{y \in Z} y_l, \; l = 1, \ldots, k.$$

Definition 1. *The outcome $\bar{y} \in Z$ is **efficient** if $y_l \geq \bar{y}_l$, $l = 1, \ldots, k$, $y \in Z$, implies $y = \bar{y}$.*

For clarity of presentation and without loss of generality, in this paper we confine ourselves exclusively to a subset of efficient outcomes, namely to *properly efficient* outcomes.

Definition 2 [3]. *The outcome $y \in Z$ is **properly efficient** if it is efficient and there exists a finite number $M > 0$ such that for each i we have*

$$\frac{y_i - \bar{y}_i}{\bar{y}_j - y_j} \leq M$$

for some j such that $y_j < \bar{y}_j$, whenever $y \in Z$ and $y_i > \bar{y}_i$.

The set of all properly efficient outcomes we shall denote by $\mathcal{P}$. The distinction between efficient and properly efficient outcomes, important in formal considerations, is of little importance in practical MCDM problems. It is enough to recall that in the case set Z is polyhedral or finite all efficient outcomes are properly efficient.

3. Pareto set characterizations

A corner stone for every interactive MCDM method is the ability to derive properly efficient outcomes. Every properly efficient outcome should be potentially derivable. The so-called characterizations of $\mathcal{P}$ are useful for this purpose. Bellow we recall two types of characterizations prized for their generality and therefore often exploited in MCDM methods, namely:

- the characterization by weight manipulations,
- the characterization by reference point manipulations.

Any of the above characterizations represents a parametric family of optimization problems.

3.1. Characterization by weight manipulations

The idea of characterizing the Pareto set by weight manipulations consists in constructing a *surrogate objective function* parameterized by k parameters – *weights*. A surrogate objective function when maximized (or minimized – depending on the surrogate objective function form) over Z yields properly efficient outcome of vector optimization problem (2) (cf. Fig. 1). By changing weights and solving the resulting optimization problems one derives different properly efficient outcomes.

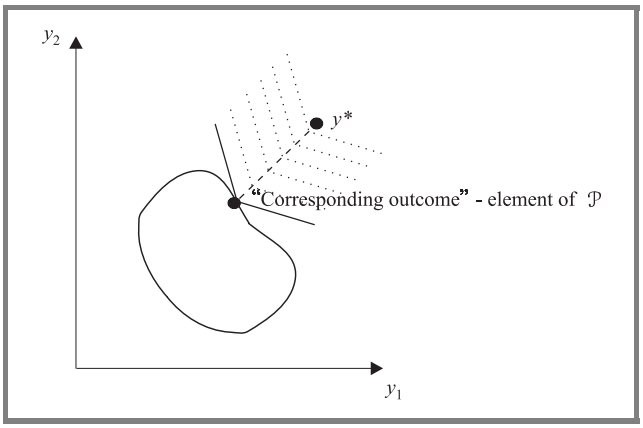


Fig. 1. Contours of a surrogate objective function and the properly efficient outcome "corresponding" to the selected vector of weights.

Below we shall make use of a selected element of the criteria space denoted y^* and defined as

$$y_l^* = \hat{y}_l + \varepsilon, \; l = 1, \ldots, k,$$

where ε is any positive number and $\hat{y}$ is the utopian element defined in the previous chapter.

Sufficient condition for proper efficiency. An outcome which solves the optimization problem

$$\min_{y \in Z} \max_l \lambda_l ((y_l^* - y_l) + \rho e^k (y^* - y)), \tag{3}$$

or the problem

$$\min_{y \in Z} \max_l \lambda_l (y_l^* - y_l) + \rho e^k (y^* - y), \tag{4}$$

where $\lambda_l > 0$, $l = 1, \ldots, k$, $\rho > 0$, and e^k is the k-dimensional row vector with all components equal to one, is **properly efficient** [1, 4, 11–13].
The surrogate functions (3) and (4) are the most general forms of functions used in weight manipulation methods.

3.2. The characterization by reference point manipulations

The idea of characterizing the Pareto set by reference point manipulations consists in constructing a surrogate objective function parameterized by an element y of $\mathcal{R}^k$. A surrogate objective function when minimized over Z yields a properly efficient outcome of vector optimization problem (2) (cf. Fig. 2). By changing reference points and solving the resulting optimization problems one derives different properly efficient outcomes.

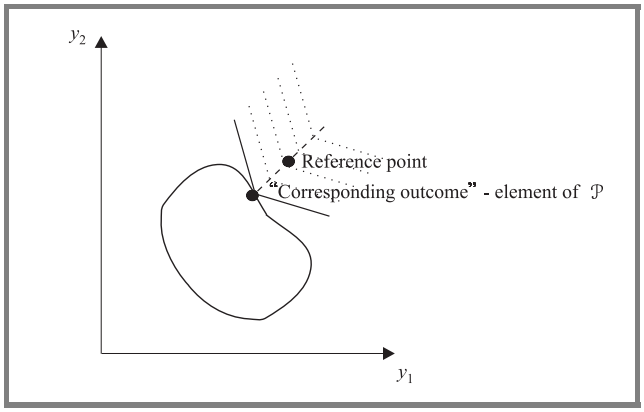


Fig. 2. Contours of a surrogate objective function and the properly efficient outcome "corresponding" to the selected reference point.

A continuous function $s_{\bar{y}}(y) : \mathcal{R}^k \to \mathcal{R}$, where $\bar{y} \in \mathcal{R}^k$, ($\bar{y}$ – a *reference point*), is called an *achievement function*. In the context of this paper it is required that an achievement function is ε-*strongly increasing* [10].

We define the following optimization problem:

$$\min_{y \in Z} s_{\bar{y}}(y). \tag{5}$$

Let outcome $\check{y}$ be a solution of problem (5), i.e.

$$\check{y} = \arg\min_{y \in Z} s_{\bar{y}}(y).$$

Sufficient condition for proper efficiency. If $s_{\bar{y}}$ is ε-strongly increasing, then outcome $\check{y}$ is **properly efficient** [10].
Functions (3) and (4) for each λ, $\lambda_l > 0$, $l = 1,\ldots,k$, are ε-strongly increasing; they both are achievement functions with $\bar{y} = y^*$. Various other forms of achievement functions exists but for the properties required achievement functions (3) and (4) posses the simplest form.

4. Methods

4.1. Weight methods

In weight methods ([2, 8, 9, 14] to name just a few, the reader is referred to e.g. [7] for a more complete list of references) the DM articulates his preferences by pointing (directly or indirectly) to a vector of weights. Then a properly efficient outcome which "corresponds" to the selected vector of weights is determined with the help of a surrogate objective function (cf. Section 3.1). The notion of correspondance is intuitively explained in Fig. 1. By manipulating weights the DM is able to determine a subset of $\mathcal{P}$ set and from this subset select the most preferred outcome.
In that manner the set of weights is systematically searched and reduced. Search can be organized in the form of *weight cuts* (the Zionts-Wallenius method and the Dell-Karwan method) or *weight zooming* (the Tchebycheff method by Steuer). Reductions of the set of weights give rise to a natural stopping rule: search is terminated if the set of weights is so small that outcomes corresponding to weights from this set differ insignificantly. Other usual stopping rules such as limit of the elapsed time or limit of iterations are of purely technical nature.

4.1.1. Weight cut methods

In weight cut methods it is assumed that the surrogate objective function used approximates locally DM's implicit utility function. With such an assumption in place a pair of outcomes subjected to DM's evaluation yields a weight cut. With the surrogate function (3) and with two outcomes y^a, y^b we have

$$\max_l \lambda_l((y^*_l - y^a_l) + \rho e^k(y^* - y^a)) < \max_l \lambda_l((y^*_l - y^b_l) + \\ + \rho e^k(y^* - y^b)) \tag{6}$$

if the DM prefers y^a to y^b, and

$$\max_l \lambda_l((y^*_l - y^a_l) + \rho e^k(y^* - y^a)) > \max_l \lambda_l((y^*_l - y^b_l) + \\ + \rho e^k(y^* - y^b)) \tag{7}$$

otherwise.

The cut (6) or (7) reduces the set of vectors λ. Vectors λ from the reduced set are selected and problem (3) is solved to derive elements of $\mathcal{P}$ for successive DM evaluations.

4.1.2. The Tchebycheff method

The so-called Tchebycheff method exploits problem (3) to determine properly efficient outcomes (in the original version of the method $\rho = 0$).
The method consists of the following operations: selecting a number of vectors λ, $\lambda_l > 0$, $l = 1,\ldots,k$, and then, iteratively:

– solving problem (3) for all selected λ to derive a number of properly efficient outcomes,

– selecting by the DM the most preferred outcome $\tilde{y}$,

– selecting a number of vectors λ, $\lambda_l > 0$, $l = 1,...,k$, in a neighborhood of $\tilde{\lambda}$ corresponding to the most preferred outcome $\tilde{y}$.

The above process has an effect of "zooming" in the set of weights in a quest for weights which yield a sequence of increasingly (or at least non decreasingly) preferred outcomes.

4.2. Reference point methods

In the simplest version of reference point methods the DM articulates his preferences by pointing to a reference point. The reference point can be an outcome, i.e. an element of Z, or any other element of $\mathcal{R}^k$. Then a properly efficient outcome which "corresponds" to the reference point and the achievement function used (cf. Section 3.2) is determined. The notion of correspondence is intuitively explained in Fig. 2. By manipulating reference points the DM is able to determine a subset of $\mathcal{P}$ set and from this subset select the most preferred outcome.

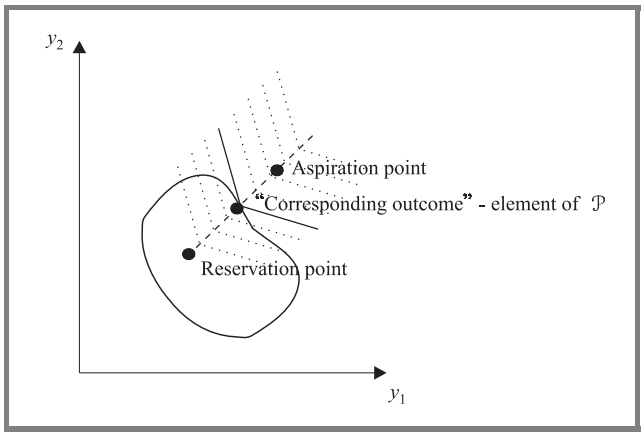


Fig. 3. Contours of a surrogate objective function and the properly efficient outcome "corresponding" to the selected pair of reservation-aspiration points.

A variant of reference point methods admits also DM pointing to a pair of reference points; a point y^{res} called a *reservation point* and a point y^{asp}, $y^{asp} \in y^{res} + int(\mathcal{R}^k_+)$, called an *aspiration point*, where $int(\cdot)$ denotes the interior of a set. It is quite natural to assume that $y^{res} \in Z$ and $y^{asp} \notin Z$ provided such points are easily identifiable. In general, the condition $y^{asp} \in y^{res} + int(\mathcal{R}^k_+)$ is sufficient. It is possible then to construct an achievement function such that an outcome y which minimizes that function over Z is an element of $\mathcal{P}$ farthest from the reservation point and at the same time closest to the aspiration point. One such an achievement function is the function (3), where

$$\lambda_l = \frac{1}{y^{asp}_l - y^{res}_l},\ l = 1,...,k.$$

This is schematically illustrated in Fig. 3.
In reference point methods no explicit evaluations (comparisons) of outcomes take place.

5. Weight versus reference point methods

5.1. Weight versus reference point methods – methodological level

On the methodological level weight methods and reference point methods represent two entirely different decision making paradigms.
In weight methods it is assumed (assumption A), often implicitly, that at each iteration of the interactive decision making process the DM is able to express his partial preferences by pointing to a preferred outcome (and hence decision) from a handful of outcomes presented to him. Then his preference is translated into relations in terms of vectors λ.
In reference point methods it is assumed (assumption B) that at each iteration of an interactive decision making process the DM is able to express his partial preferences by pointing to a reference point representing his preferred decisional pattern, or, as in the variant of the reference point methods, by pointing to a pair of reservation-aspiration points.
There is no decisive evidence which assumption is better justified. Quite evidently assumption A is better justified than assumption B when the DM posses some analytical capabilities. In turn, assumption B seems to be better justified than assumption A when the DM acts intuitively and tends to present his preferences in a holistic manner. Pointing to a reference point is a holistic form of expressing preferences.

5.2. Weight versus reference point methods – technical level

Let us observe that in weight methods selecting at each iteration a vector $\lambda > 0$ amounts in fact to selecting a half-line starting from y^* along which the apexes of the contours of the function (3) lie (cf. Fig. 1). This line has the form

$$s = y^* - t\tau,$$

where $t > 0$ and $\tau = (\tau_1,...,\tau_k)$, $\tau_l = \frac{1}{\lambda_l}$, $l = 1,...,k$. In course of iterations one gets a "fan" of half-lines all starting at y^* (Fig. 4).
In reference point methods the DM specifies at each iteration a reference point y^{ref}, what amounts in fact to selecting

(recall we have assumed that (3) is the achievement function) a half-line starting from y^{ref}, i.e.

$$s' = y^{ref} - t\tau,$$

where $t > 0$ and $\tau = (\tau_1, ..., \tau_k)$, $\tau_l = \frac{1}{\lambda_l}$, $l = 1, ..., k$. In course of iterations one gets a "forest" of parallel half-lines (the vector λ is fixed) (Fig. 5).

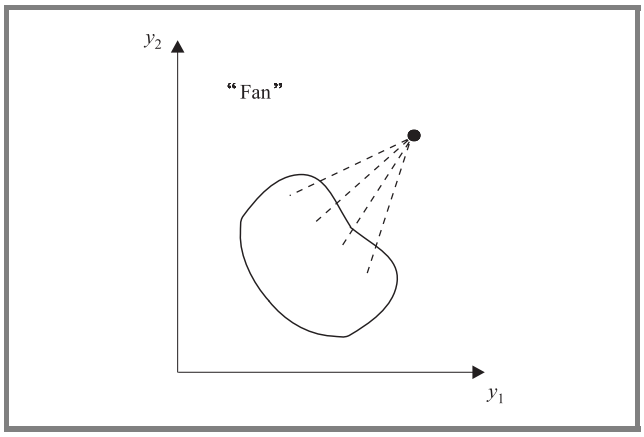


Fig. 4. A fan-type interactive decision making process.

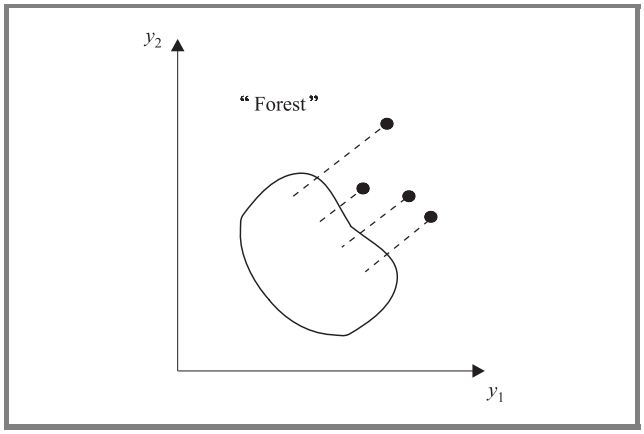


Fig. 5. A forest-type interactive decision making process.

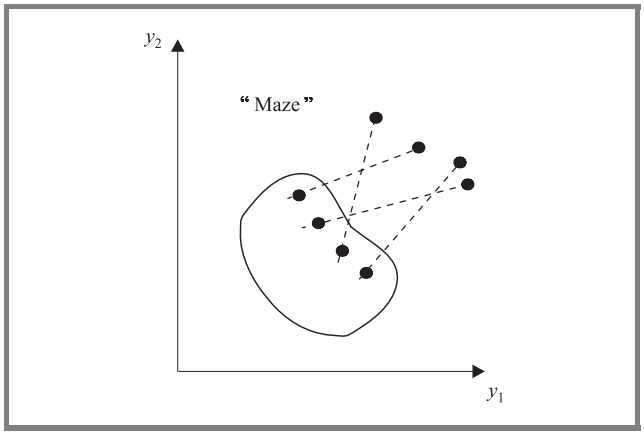


Fig. 6. A maze-type interactive decision making process.

In the variant of reference point methods the DM specifies at each iteration a reservation point y^{res} and an aspiration point y^{asp}, what amounts in fact to selecting (recall we have assumed that (3) is the achievement function) a half-line starting from y^{asp} and passing through y^{res}, i.e.

$$s'' = y^{asp} - t\tau,$$

where $t > 0$ and $\tau = (\tau_1, ..., \tau_k)$, $\tau_l = \frac{1}{y_l^{asp} - y_l^{res}}$, $l = 1, ..., k$. In course of iterations one gets a "maze" of half-lines (Fig. 6).

Table 1 summarizes the mechanics of the methods.

Table 1
Mechanics of considered interactive decision making methods

Methods	Decisional item	
	fixed	to be selected
Weight methods	y^*	τ
Reference point methods	τ	y^{ref}
Reference point methods – the variant	–	y^{res}, y^{asp}

From Table 1 we see that though weight methods and reference point methods represent totally different decision making (searching) methodologies, technically they are very similar. Indeed, in each method to proceed to the next iteration, i.e. to derive a subsequent trial outcome, a combination of two decisional items is required: either two elements of $\mathcal{R}^k$ or a direction and an element of $\mathcal{R}^k$. The methods differ in presence or absence of fixed items and in which item is an active toll to search over the set of efficient outcomes.

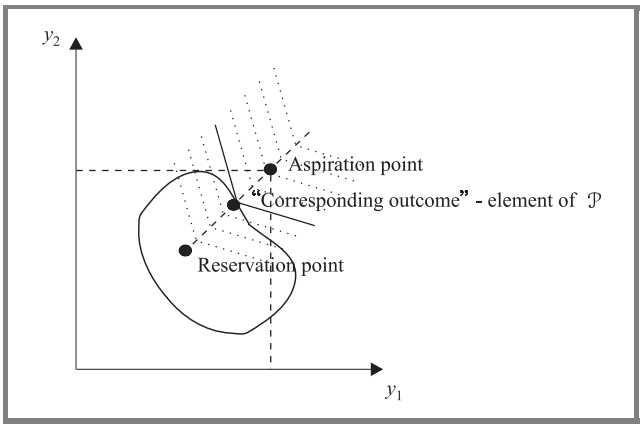


Fig. 7. Aspiration points take the role of y^* in appropriately constrained set Z.

The most flexible method is the variant of reference point methods since no decisional item is fixed a priori. Let us note, however, that a great extent of flexibility is not necessarily always plausible.

We can make weight methods and the variant of reference point methods *technically equivalent* with the following re-

striction of the latter. Let us assume $y^{asp} = y^*$, i.e. the aspiration point is fixed. Then DM changes only reservation points what amounts in fact to selecting a half-line starting from y^* and passing through y^{res}, i.e.

$$s''' = y^* - t\tau,$$

where $t > 0$, and $\tau = (\tau_1, ..., \tau_k)$, $\tau_l = \frac{1}{y_l^* - y_l^{res}}$, $l = 1, ..., k$.
If it is the case, we can say that both methods rely on a direction selection mechanism.
Let us observe that y^* plays the same role to Z as y^{asp} (y^{ref}) to the set

$$Z \cap \{y \,|\, y_l \leq y_l^{asp} - \varepsilon,\ l = 1, ..., k\}$$

$$(Z \cap \{y \,|\, y_l \leq y_l^{ref} - \varepsilon,\ l = 1, ..., k\}),$$

where $\varepsilon > 0$ is the value we used in Section 3 to define y^* (cf. Fig. 7).
To decide if the proposed restriction of the variant of reference point methods is methodologically justified a vast practical experience with applications of these methods is required and this is lacking. At this stage we can only note that the technical equivalence of weight methods and the variant of reference point methods we have just shown has some interesting practical consequences.

6. Discussion

There are two major practical consequences of the technical equivalence of weight methods and the variant of reference point methods.
The first consequence is that with the equivalence shown two methods, one representing the class of weight methods and other representing the class of reference point methods, can be implemented jointly with the same computing (optimizing) software and an interface admitting the DM to select which of these two methods he would like to work. This would establish a firm ground for systematic comparison of these methods in the same technical environment. This also would open a way for some hybrid type decision processes mixing elements of the two methods.
The second consequence is as follows. In weight methods the principle of weight set reduction gives rise to a natural convergence measure. Namely, convergence can be controlled (and a stopping rule invoked) basing on the "volume" of sets of weights resulting from subsequent reductions. In general, reference point methods do not incur a similar natural convergence measure.
Only by the simple modification proposed above the variant of reference point methods acquires this property. Indeed, any two outcomes from two subsequent iterations give rise to a cut (6) or (7) and in consequence to a reduction of the set of weights. Though the DM would have to answer questions "which of two outcomes do you prefer?" those are kind of technical questions of no influence on the course of the decision process, which relies in, we recall, selecting a reservation point with the aspiration point fixed at y^*.

7. Concluding remarks

The fact that weight methods and the variant of reference point methods can be realized in the same technical framework has an important practical consequence. Namely, as shown in companion papers of the author [5, 6], with weight methods it is possible to calculate bounds, lower and upper, on values of criteria (outcome components) prior to explicitly decision determining. This possibility is of utmost practical importance for applications of MCDM methods because using bounds instead of exact values one can avoid determining decisions explicitly and hence solving optimization problems. Since, as shown above, weight and the variant of reference point methods can be reduced to (and implemented in) the same technical framework realizing a "fan" type decisional process, reference point methods also enjoy this property. This aspect will be a topic for further research.

References

[1] E. U. Choo and D. R. Atkins, "Proper efficiency in nonconvex programming", *Math. Oper. Res.*, vol. 8, pp. 467–470, 1983.

[2] R. F. Dell and M. H. Karwan, "An interactive MCDM weight space reduction method utilizing a Tchebycheff value function", *Nav. Res. Log.*, vol. 37, pp. 263–277, 1990.

[3] A. M. Geoffrion, "Proper efficiency and the theory of vector maximization", *J. Math. Anal. Appl.*, vol. 22, pp. 618–630, 1968.

[4] I. Kaliszewski, *Quantitative Pareto Analysis by Cone Separation Technique*. Boston: Kluwer, 1994.

[5] I. Kaliszewski, "Multiple criteria decision making without optimization", *Syst. Sci.*, vol. 27, no. 2, pp. 5–13, 2002.

[6] I. Kaliszewski, "Dynamic parametric bounds of efficient outcomes in interactive multiple criteria decision making problems", *Eur. J. Oper. Res.*, vol. 147, pp. 94–107, 2003.

[7] K. M. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer, 1999.

[8] A. Roy and J. Wallenius, "Nonlinear and unconstrained multiple-objective optimization: algorithm, computation, and application", *Nav. Res. Log.*, vol. 38, pp. 623–635, 1991.

[9] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*. New York: Wiley, 1986.

[10] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization", in *Multiple Criteria Decision Making; Theory and Applications*, G. Fandel and T. Gal, Eds., *Lecture Notes in Economics and Mathematical Systems*. Berlin: Springer, 1980, vol. 177, pp. 468–486.

[11] A. P. Wierzbicki, "On the completeness and constructiveness of parametric characterizations to vector optimization problems", *OR Spectrum*, vol. 8, pp. 73–87, 1986.

[12] A. P. Wierzbicki, "Multiple criteria solutions in noncooperative game theory", Part III: "Theoretical foundations". Discussion Paper 288, Kyoto: Kyoto Institute of Economic Research, Kyoto University, 1990.

[13] A. P. Wierzbicki, "Reference point approaches", in *Multicriteria Decision Making – Advances in MCDM: Models, Algorithms, Theory and Applications*, T. Gal, Th. Stewart, and Th. Hanne, Eds. Kluwer, 1999.

[14] S. Zionts and J. Wallenius, "An interactive multiple objective linear programming method for a class of underlying nonlinear value functions", *Manag. Sci.*, vol. 29, pp. 519–529, 1983.

Ignacy Kaliszewski holds Assistant Professor position in the Systems Research Institute of the Polish Academy of Sciences in Warsaw, from which he got his Ph.D. and D.Sc. degrees, and also Professor position in College for Management and Finance in Wrocław. His interests are in single and multiple criteria programming, aided decision making, financial engineering and competitive market organization. He published over 60 research papers, a research monograph (Quantitative Pareto Analysis by Cone Separation Technique by Kluwer), and over 30 popular papers on risk issues pertaining to trading in financial instruments and commodities. He has also a business experience with a bank, a futures commodity exchange, and a clearing house. Ignacy Kaliszewski is a founding member and currently the Secretary of the Poland Chapter of the Institute for Operations Research and the Management Sciences (INFORMS), member of the Polish Mathematical Society, and the Polish Society for Operational and Systems Research.
e-mail: ignacy.kaliszewski@ibspan.waw.pl
Systems Research Institute
Polish Academy of Sciences
Newelska st 6
01-447 Warsaw, Poland

Paper

Application of multiple criteria evolutionary algorithms to vector optimisation, decision support and reference point approaches

Marcin Szczepański and Andrzej P. Wierzbicki

Abstract — Multiple criteria evolutionary algorithms, being essentially parallel in their character, are a natural instrument of finding a representation of entire Pareto set (set of solutions and outcomes non-dominated in criteria space) for vector optimisation problems. However, it is well known that Pareto sets for problems with more than two criteria might become complicated and their representation very time-consuming. Thus, the application of such algorithms is essentially limited to bi-criteria problems or to vector optimisation problems with more criteria but of simple structure. Even in such cases, there are problems related to various important aspects of vector optimisation, such as the uniformity of representation of Pareto set, stopping tests or the accuracy of representing Pareto set, that are not fully covered by the broad literature on evolutionary algorithms in vector optimisation. These problems and related computational tests and experience are discussed in the paper. In order to apply evolutionary algorithms for decision support, it would be helpful to use them in an interactive mode. However, evolutionary algorithms are in their essence global and of batch type. Nevertheless, it is possible to introduce interactive aspects to evolutionary algorithms by focusing them on a part of Pareto set. The results of experimental tests of such modifications of evolutionary algorithms for vector optimisation are presented in the paper. Another issue related to vector optimisation problems with more than two criteria is the computational difficulty of estimating nadir points of Pareto set. The paper describes the use of diverse variants of evolutionary algorithms to the estimation of nadir points, together with experimental evidence.

Keywords — evolutionary algorithms, vector optimisation, nadir point estimation, reference point techniques.

1. Evolutionary algorithms in vector optimisation: general comments

There are many excellent reviews of evolutionary algorithms used in vector optimisation [3–5, 10, 12]. Most of them, however, treat evolutionary or genetic algorithms as goals in themselves, as given tools that should be further developed and put into use. In this paper, we concentrate rather on the use of such algorithms for solving various tasks of vector optimisation or multiple criteria analysis for decision support.

First, let's recall the traditional distinction between genetic and evolutionary algorithms: genetic algorithms rely on binary representation of individuals, while evolutionary algorithms admit real-valued (computational) representations. For vector-valued representations, evolutionary algorithms are more appropriate. On the other hand, special methods developed for genetic algorithms can be also usefully translated into evolutionary algorithms.

Next, we observe that evolutionary algorithms are applied to vector optimisation in order to obtain accurate representation of the Pareto set (or any modified concept of a non-dominated set). Being inherently parallel, evolutionary algorithms are a natural approach to the problem of representing a complicated set. However, research on truly parallel or distributed implementations of evolutionary algorithms is scarce. Thus, the application of such algorithms is essentially limited to bi-criteria problems or very simple vector optimisation problems with more criteria. Accurate representation of more complicated Pareto sets using evolutionary algorithm still requires huge computation efforts.

On the other hand, practical applications of vector optimisation to decision support require interactive multiple criteria analysis [11], where instead of computing a single Pareto set, various characteristics of selected variants or parts of Pareto sets are needed for subsequent formulations of the problem being analysed. Such cases include utopia points, nadir points, neutral compromise points of Pareto sets and, finally and most importantly for interactive applications – representations of selected segments of Pareto sets. While evolutionary algorithms might be useful for obtaining such characteristics, little attention was given to such applications. Generally speaking, the same fact can be stated as follows: *since evolutionary algorithms are global and non-interactive in their nature, the challenge in their applications for multiple criteria analysis is to make them more local and interactive*. While this paper does not resolve all problems related to this challenge, it tries to move in this direction – by treating evolutionary algorithms not as main goal in itself, but as a way of addressing various tasks of multiple criteria analysis.

2. Modifications of evolutionary algorithms in vector optimisation

2.1. Representation of individual

By *individual* in genetic or evolutionary algorithms, we consider a current solution point together with additional parameters, typically characterising its mutation potential by specifying the dispersion σ. In vector optimisation or multiple criteria analysis, current solution is typically represented by a vector of decision variables $x \in R^n$ and vector of decision outcomes or criteria $q \in R^k$. Dispersion parameters are related to decision variables and can be represented by a vector of the same dimension. Thus, an individual is represented by:

$$ind = (x, \sigma, q) \in R^{2n+k}. \tag{1}$$

2.2. Constraints

Constraints on decision variables (either in equation or inequality form) define the permissible set of decisions:

$$X_0 = \left\{ x \in R^n : \begin{array}{l} g_i(x) \geq 0, i \in I \\ h_i(x) = 0, i \in E \end{array} \right\}. \tag{2}$$

In genetic algorithms, if x is not in X_0, the individual is simply discarded. This may lead, however, to quite long computations if the set X_0 has a complicated structure. Therefore, we shall use a method typically adopted in evolutionary algorithms to represent constraints – applying penalty functions. There are many types of penalty functions (internal, external, exact, shifted, etc. – see e.g. [11]). With evolutionary algorithms that do not need derivatives of optimised functions, it is best to use exact non-differentiable external penalty functions of the type $|h_i(x)|$ and $|g_i(x)_-| = \min\left(g_i(x), 0\right)|$ (with sufficiently large penalty coefficients), which are added to each criterion value – if it is minimised or subtracted – if maximised.

2.3. Cross-breeding

Cross-breeding is a typical evolutionary operation. In vector optimisation, cross-breeding applies to two parent individuals represented by decision variable vectors x_1 and x_2; their successor x' may be determined as follows:

$$x' = ax_1 + (1-a)x_2, \tag{3}$$

where the parameter a is a random variable from the interval $a \in [0, 1]$. This is called *basic arithmetic cross-breeding*, while *extended arithmetic cross-breeding* applies to each component x'_i of the vector x' with separately generated random coefficients a_i. There are several other variants of cross-breeding, such as *heuristic cross-breeding*, not discussed here.

2.4. Mutation

In vector optimisation, mutation is applied to every component x_i of the decision variable vector x (usually, mutation is additionally applied to a successor of cross-breeding) by selecting a random variable with a normal distribution and modifying the component x_i by this variable with a corresponding dispersion coefficient:

$$\begin{aligned} \xi_i^x &\in N(0, 1), \\ x'_i &= x_i + \sigma'_i \xi_i^x. \end{aligned} \tag{4}$$

Additionally, the dispersion coefficient is modified randomly, but usually slowly decreased after (or before) each mutation. This decreasing modification of dispersion parameters slows down mutations when approaching solutions. In vector optimisation, it results in coming closer to the Pareto set.

2.5. Selection

Selection is responsible for convergence of a genetic algorithm towards optimal solutions and applies to selection of parent individuals (selection in reproduction); there are numerous methods of such selection, not discussed here. In evolutionary algorithms, succession may substitute for selection. This means choosing the μ as best individuals from population $\mu + \lambda$ (so-called $\mu + \lambda$ succession strategy; μ denotes here a population from parent individuals, λ the corresponding population of successors) in some way. Another strategy consists of simply substituting parent population μ by successor population λ (the so-called μ, λ strategy). For vector optimisation purposes, succession is superior to selection.

2.6. Pareto ranking

Succession process includes multiple stages to uniformly approximate Pareto set by an evolutionary algorithm. First, we use *Pareto ranking of a population*, then apply special *niched methods* for preserving uniformity of representation, and finally use special *succession methods*. We will describe all of them below.

Pareto ranking consists of attaching a rank value (the lower the better) to each individual. Goldberg [2] has proposed to give rank 1 to each non-dominated individual in population. Next, we delete the non-dominated individuals and determine non-dominated individuals in remaining part of population, giving them rank 2. We continue the process with increasing rank values until each individual has a rank value. Then we can either select successor population of given number of individuals according to lowest rank values, or – as proposed by Goldberg – determine the probability of reproduction depending on rank value (which is actually a selection, not a succession mechanism).

Another ranking method proposed by Fonseca and Fleming [4] involves assigning each individual a rank value of 1 plus the number of other individuals dominating this individual. This method provides for more differentiation of a population than Goldberg method.

Having a rank value, it is easy to determine a fitness indicator $fit(x)$ – for example, by defining it as inverse of the rank value.

2.7. Niched methods

Having a fitness value or fitness function for Pareto ranking, it is easy to apply the basic principle of evolutionary algorithms – the *survival of the fittest* individuals.

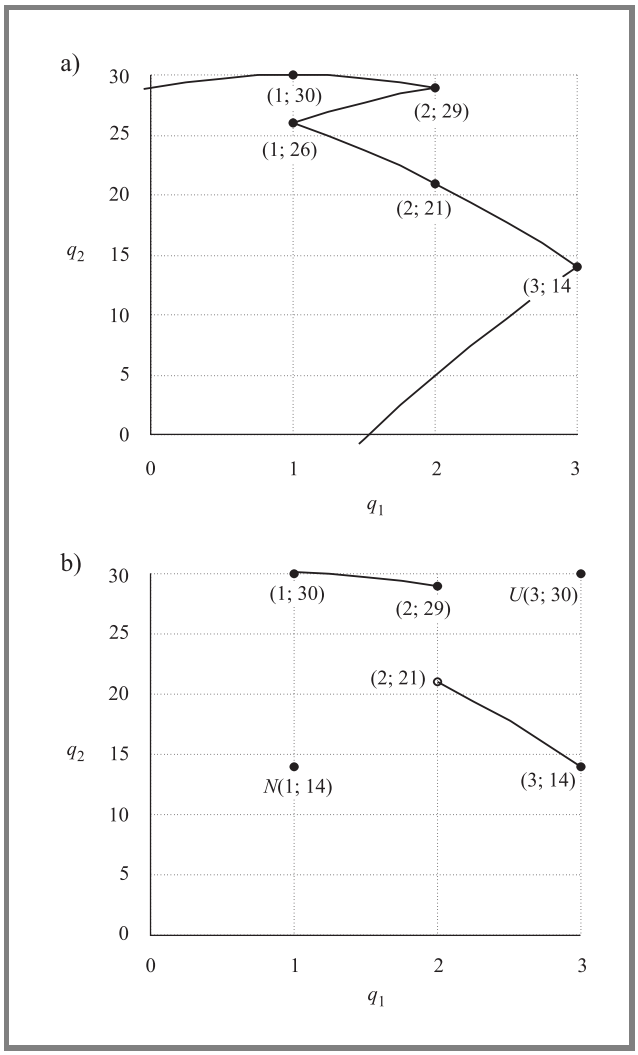


Fig. 1. The set of attainable criteria values (a) and the Pareto set (b) for the nonlinear example.

However, such a method does not result in a uniform representation of the Pareto set. The fittest individuals can form an elite close to each other, representing only an "easy" part of Pareto set. Such degeneration of the *survival of the fittest* principle can be illustrated by a relatively simple, but nonlinear example (Fig. 1). We maximize two criteria functions (with $-0.5 \le x \le 6$):

$$\begin{aligned} \max: q_1(x) &= \left\{ \begin{array}{ll} x+2 & x \le 1 \\ -x+4 & 1 < x \le 3 \\ x-2 & 3 < x \le 4 \\ -x+6 & x > 4 \end{array} \right\}, \\ \max: q_2(x) &= -x^2 + 10x + 5. \end{aligned} \tag{5}$$

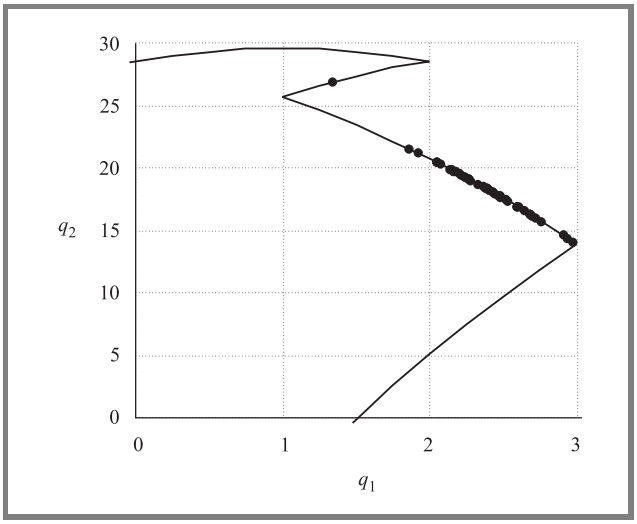


Fig. 2. Non-uniform representation of Pareto set with a simple *survival of the fittest* evolutionary algorithm (population size: 50, 200 generations).

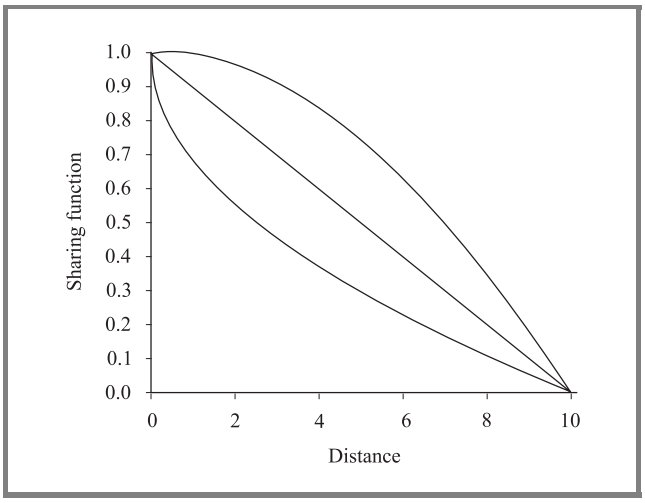


Fig. 3. Examples of sharing functions.

Application of a simple *survival of the fittest* algorithm here results in a degenerated representation of the Pareto set, concentrating on the "easy part" of the set (Fig. 2).

In order to overcome this difficulty, we must penalise the fitness function for individuals being too close to each other.

With this aim, we define a *sharing function* depending on a distance of two individuals, say x and x_0. This sharing function sh must have the following properties:

$$\begin{gathered} 0 \le sh(x-x_0) \le 1, \text{ for any distance } |x-x_0| \\ sh(0) = 1 \\ \lim sh(x-x_0) = 0, |x-x_0| \to \infty . \end{gathered} \tag{6}$$

Sharing functions shown in Fig. 3 belong to the family:

$$sh(x-x_0) = \begin{cases} 1 - \left(\frac{|x-x_0|}{D} \right)^p & |x-x_0| < D \\ 0 & |x-x_0| \ge D \end{cases}, \tag{7}$$

where D is a diameter of a *niche*.
The so-called *niched methods* consist of modifying fitness values $fit(x)$ for a given individual x, reciprocal to the sharing function:

$$fit'(x) = \frac{fit(x)}{1+m(x)}, \tag{8}$$

where $m(x)$ is a sum of sharing functions over other non-dominated individuals y in given population:

$$m(x) = \sum_y sh\big(d(x,y)\big). \tag{9}$$

Figure 4 illustrates effectiveness of such niched methods in preventing degeneration through cross-breeding of too close individuals.

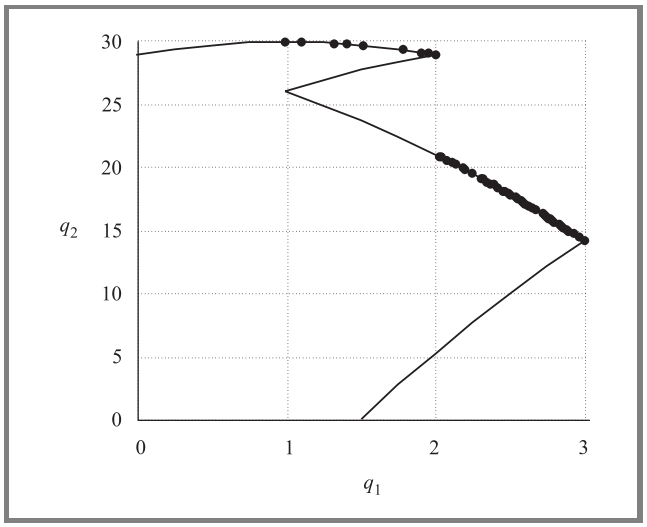


Fig. 4. Effectiveness of a *niched method* with $D = 0.1$ (population size: 50, 200 generations).

We see it is necessary to use niched methods in evolutionary algorithms of vector optimisation not only in order to obtain a uniform representation of Pareto set, but also to prevent degenerate populations resulting from naive direct application of the "survival of the fittest" principle.

2.8. Stopping tests

Before discussing succession methods, stopping tests for entire algorithm should be discussed. Stopping test for evolutionary and genetic optimisation algorithms are much less developed than for analytical optimisation methods. If the optimal value of an optimised function is known (which happens only in very special cases) then the distance from this optimal value can be used for a stopping test. Otherwise, one must limit the number of iterations in the algorithm (number of generations in a genetic or evolutionary algorithm) and hope for a good accuracy. Another stopping test is based on the speed of change of an approximation of the solution: work stops when changes fall below certain level.
For vector optimisation, the issue of stopping tests is more complicated. We can rely on a given number of iterations or generations, but cannot easily use the speed of change, because we approximate or represent an entire Pareto set and the uniformity of this representation is also a goal. A substitute for the speed of change might be a comparison of two subsequent generations and checking how many individuals in the next generation dominate some individuals in the former generation. Figure 5 shows example of such computation.

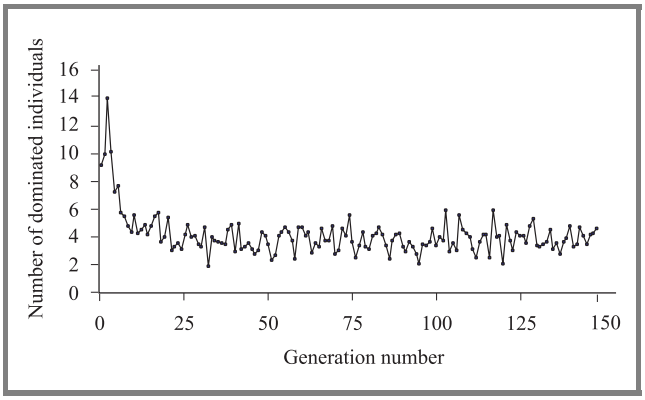


Fig. 5. Average numbers of dominated individuals between generations for a typical evolutionary algorithm.

We see such a stopping test cannot be very reliable. Other tests, however, might be related to special features of vector optimisation. One relates to the uniformity of Pareto set representation, which can be represented by average value of sharing function $m(x)$ as defined by Eq. (9). Another relates to the concept of utopia and nadir points for a Pareto set. For an approximation of Pareto set obtained in a subsequent generation numbered here by i, it is relatively easy (see also point **4**) to compute utopia points q_i^U ("lowest" points dominating entire Pareto set) as well as nadir points g_i^N ("highest" points dominated by the entire Pareto set). If the approximation of a Pareto set converges to the actual Pareto set, the distance between the approximations of utopia and nadir points:

$$un(i) = |q_i^U - q_i^N| \tag{10}$$

increases and converges to the value characterising the actual Pareto set. We illustrate both of these concepts on an example (we use here an example defined later by Eq. (22)) – see Figs. 6 and 7.

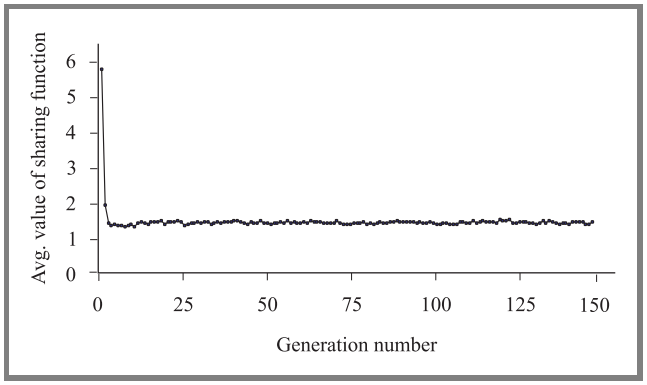


Fig. 6. Average values of sharing function in subsequent generations for Eq. (22).

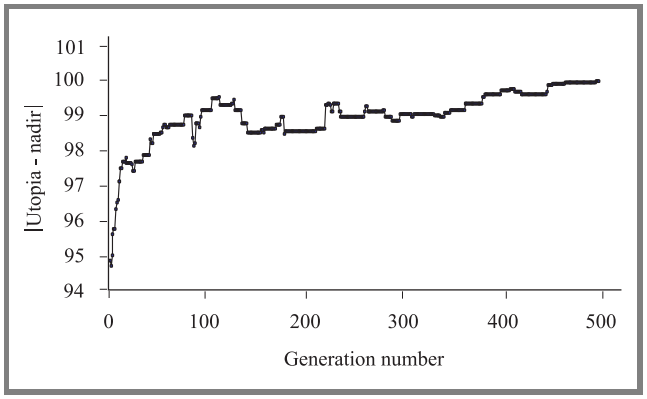


Fig. 7. Utopia-nadir distances in subsequent generations for Eq. (22).

We observe that after a small number of iterations most of the analysed measures oscillate around a constant value and thus are not particularly useful for stopping tests. An exception is the distance of utopia and nadir approximations, which converges to a constant value after a relatively large, but reasonable number of iterations. Thus, *the relative change of the distance of utopia and nadir approximations is the best stopping test for estimating a Pareto set by evolutionary algorithms*.

2.9. Succession methods

Application of niched methods results in decreasing fitness of an individual in densely represented parts of a Pareto set. However, this might lead to concentration on the boundaries of the Pareto set, demonstrated by the following example. Analysing how to choose successors in order to get a uniform representation of a Pareto set, we investigated a simple case: let the Pareto set in three-dimensional space belong to the plane $z = 0$ and be a square $x, y \in < 0; 9 >$. The simplest niched method with the niche diameter of 2 gives the following values of fitness function (100 points arranged in square table) shown in Fig. 8.

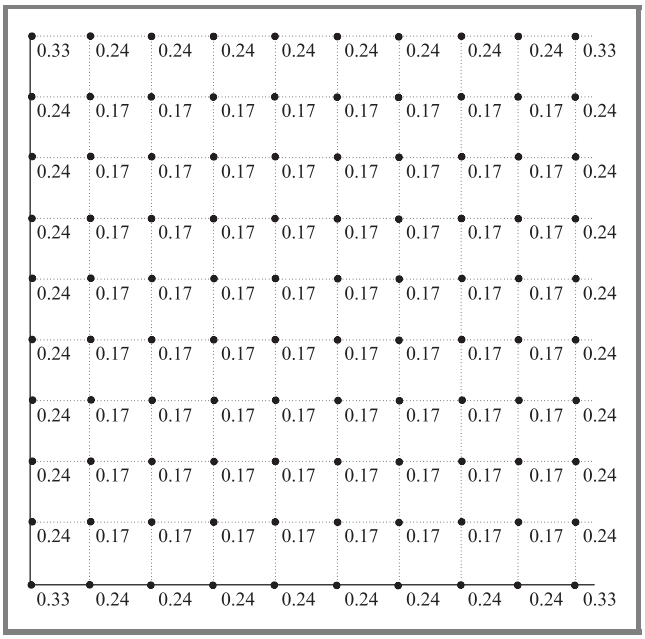


Fig. 8. Values of a fitness function for the simple case considered.

By applying the simplest succession method based on a simple ranking of the individuals to this case, we promote individuals located on the boundary of Pareto set (Fig. 9).

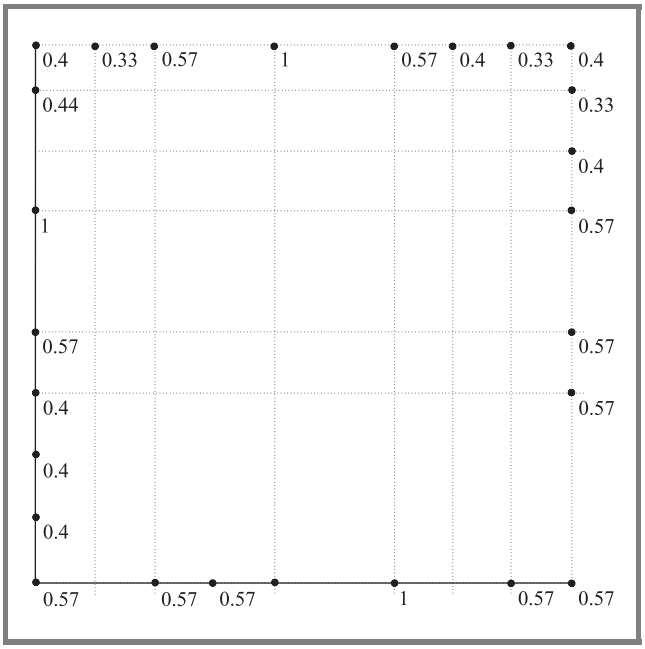


Fig. 9. Successors in the simple case with basic ranking succession rule ($\mu = 0.25$).

Table 1
Comparison of various succession methods

Parameter	Succession method				
	ranking	roulette	tournament	modified fitness	deterministic
	$\mu = 10$				
Computing time [ms]	36.59	4.84	7.40	28.56	1744
Average fitness	0.807	0.807	0.779	0.820	1
	$\mu = 25$				
Computing time [ms]	39.13	8.89	14.30	33.30	1665
Average fitness	0.520	0.579	0.548	0.601	1
	$\mu = 50$				
Computing time [ms]	46.92	19.14	29.70	43.63	1466
Average fitness	0.374	0.367	0.348	0.381	0.469

We can also imagine a deterministic (actually – non-evolutionary) succession rule in which we eliminate in a deterministic loop subsequent individuals, while increasing the fitness of its neighbours. The process is repeated until the population drops to a given number of individuals, as illustrated by Fig. 10.

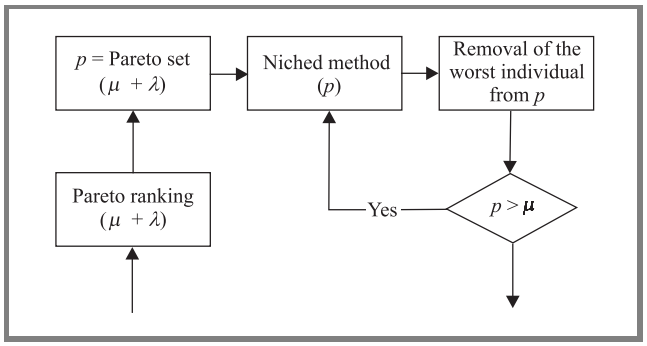


Fig. 10. Block-diagram of a deterministic succession rule.

Another succession rule is obtained by modifying definition of coefficient $m(x)$, needed to determine fitness. Instead of summing it up over all non-dominated individuals as in Eq. (9), it can be summed up only for individuals with lower index numbers on the list:

$$m(x) = \sum_{x=1}^{y-1} sh\big(d(x,y)\big)\,. \tag{11}$$

That way, the individuals considered first on the list obtain greater fitness indicators (Fig. 11).

Yet another methods of succession for evolutionary vector optimisation can be obtained by modifying roulette and tournament approaches to general evolutionary algorithms. Recall that a roulette approach determines successors (or selects individuals for cross-breeding) randomly, with probability increasing with the fitness indicator. Tournament approach determines successors by selecting randomly k individuals for a tournament and then selecting the tournament winner as the individual with highest fitness indicator (or randomly selects one of them, if there is a tie). Both approaches give similar results in our case (Fig. 12).

The above mentioned methods were compared in terms of their accuracy (defined by uniform coverage of the Pareto set, measured by average value of fitness indicator, that should be highest for a uniform coverage) and computational effort needed to solve this simple case. Table 1 gives results obtained by using a PC with 700 MHz Pentium III processor, after a large number of generations (10 000).

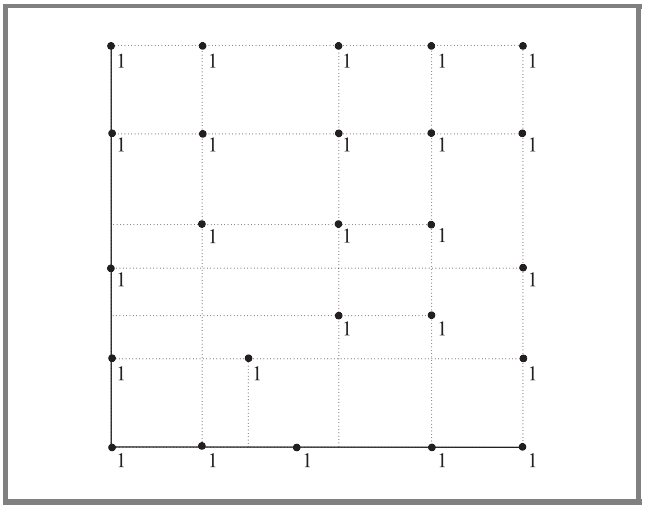


Fig. 11. Successors in the simple case with deterministic succession rule.

The most uniform representation of the Pareto set is obtained by deterministic method, though the required computing time is rather large. Among other methods, simple ranking method gives the least uniform representation – as can be expected since it favours individuals on the edge of Pareto set. For further experiments, either the roulette method (giving shortest computing time) or the determin-

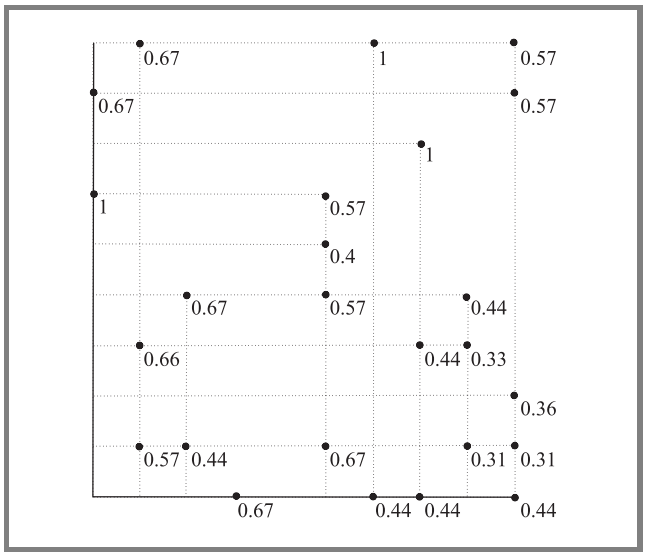


Fig. 12. Successors obtained by a roulette method in the simple case – the tournament method gives similar results.

istic method (ensuring uniform representation), were typically used. We will show later that performance of ranking succession method can be considerably improved if a more sophisticated ranking method is used.

2.10. Accuracy of representing Pareto set

When analysing more complicated Pareto sets than the simplest example presented before, it was observed that evolutionary algorithms do not converge precisely to the actual Pareto set. In a sense, this phenomenon is obvious: due to mutation necessary for evolutionary behaviour, only a few individuals come precisely to the Pareto set; most of them are oscillating just "below" the Pareto set. Even if obvious, this aspect was not sufficiently stressed and analysed in the literature. We give here results of investigating – in some cases for quite a long time with up to 30 000 generations – a simple example with known Pareto set, obtained by linear vector optimisation:

$$\begin{aligned} &\max x_j,\ j = 0,\ldots,i,\\ &\sum_{j=0}^{i} x_j \le 1,\\ &x_j \ge 0,\ j = 0,\ldots,i. \end{aligned} \tag{12}$$

We see that for the investigated example with $i = 2$, the average distance form the Pareto set oscillates about $4 \cdot 10^{-3}$ (actually, $3.76 \cdot 10^{-3}$) after only 200 generations (Fig. 13). Naturally, this value depends on the limit values for decreasing the dispersion parameter σ. This is because the oscillation of the distance from the Pareto set results from recombination and (predominantly) the mutation operation. Even if the original population were situated precisely on the Pareto set, mutation would put successors "below" this set, as illustrated by the following simple example (Fig. 14).

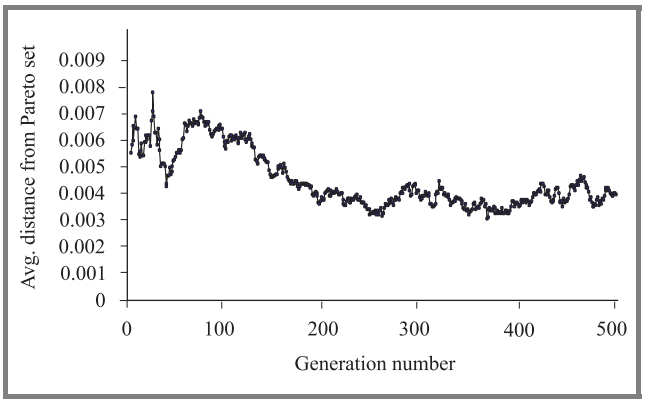


Fig. 13. Average distance from Pareto set for the example defined by Eq. (12) ($i = 2$, $\mu = 100$, $\lambda = 100$, $r = 0.01$).

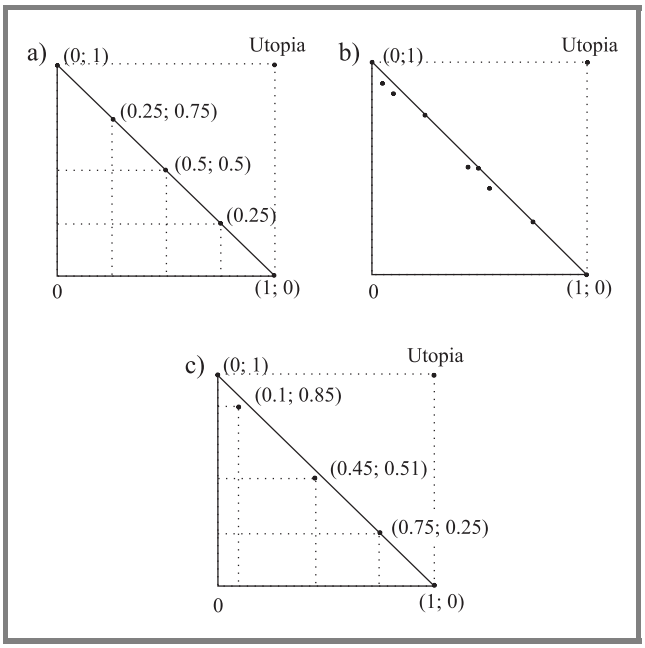


Fig. 14. Population on Pareto set (a), successor population after recombination and mutation (b) the same after succession (c).

We could of course force the algorithm to converge to the precise Pareto set, if we decided to decrease the mutation effect through decreasing dispersion parameter σ to zero. This would result, however, in losing exploratory powers of the evolutionary algorithm, considered a degeneration of the algorithm. Precise dependence of accuracy of approximating Pareto sets on the limit values of dispersion parameters requires further detailed study.

3. Use of reference points and achievement functions in evolutionary algorithms

A powerful and practical way of making vector optimisation algorithms interactive is to combine them with the

concepts of reference points and to use order-preserving achievement functions [11]. We will investigate here, how to combine these concepts with evolutionary algorithms in order to either make them more interactive or to eliminate other deficiencies.

3.1. Segments of Pareto sets dominating a reference point

In interactive analysis of Pareto sets, it might be interesting to approximate a part of Pareto set "above" a given reservation point q^{res} – see the example shown in Fig. 15. We have to add constraints:

$$f(x) \geq g_i^{res},\ i = 1, \ldots, k_1 \text{ (for maximised criteria)},$$
$$f(x) \leq g_i^{res},\ i = k_1 + 1, \ldots, k \text{ (for minimised ones)}. \quad (13)$$

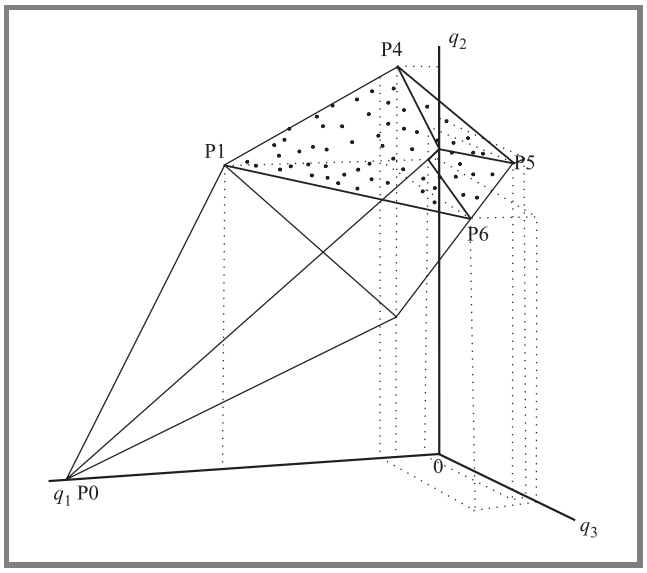


Fig. 15. A part of Pareto set above a given reservation point (0, 24, 0).

Provided that the resulting problem is feasible (the reservation point is not "above" Pareto set), specifying such additional requirement does not complicate the evolutionary algorithm. Additional constraints are simple and can be taken into account as selection conditions. We can also achieve a better approximation accuracy if the reservation point lies close to Pareto set. For the relatively simple examples of Pareto sets considered here, the necessary computational effort does not diminish, however: approximating a part of Pareto set is as expensive as approximating the entire set. On the other hand, the necessary computational effort is reasonable for simple examples. Interactive investigation by approximating first entire Pareto set, and approximating selected parts of it more precisely later is possible.

3.2. Using achievement functions for better ranking and for improving the accuracy of representing Pareto set

Ranking Pareto in evolutionary algorithms can be modified by using an order-consistent achievement function (see also [11]), e.g.:

$$\sigma(q, \overline{q}) = \min_{1 \leq i \leq m} \sigma_i(q_i, \overline{q}_i) + \varepsilon \sum_{i=1}^{m} \sigma_i(q_i, \overline{q_i}), \quad (14)$$

where $\overline{q}$ is a reference point in criteria space. The partial achievement functions can be defined for a simple case as follows:

$$\sigma_i(q_i, \overline{q}_i) = \frac{q_i - \overline{q}_i}{q_i^U - q_i^N} \text{ (for maximised criteria)},$$
$$\sigma_i(q_i, \overline{q}_i) = \frac{\overline{q}_i - q_i}{q_i^N - q_i^U} \text{ (for minimised ones)}, \quad (15)$$

where $\overline{q}^U$ and $\overline{q}^N$ are utopia and nadir point vectors or their approximations, respectively. Modification of Pareto ranking is based on the following property of the achievement function:

$$\overline{q} \in Q_0 \Rightarrow \left\{ \begin{array}{c} \max_{q \in Q_0} \sigma(q, \overline{q}) \geq 0 \\ \widehat{q} = \arg\max_{q \in Q_0} \sigma(q, \overline{q}) \geq \overline{q} \end{array} \right\}. \quad (16)$$

Thus, the value $\sigma(q, \overline{q})$ greater than 0 indicates (approximately), that point q dominates the reference point $\overline{q}$. The value 0 of the achievement functions indicates that point q is either equal or (approximately) equivalent to $\overline{q}$. Because of these properties, the Pareto rank of an individual can be determined by:

$$rank_j^{(t)} = 1 + \sum_{k=1}^{S_j} \sigma(q_k, q_j), \quad (17)$$

where q_k are individuals dominating q_j, thus $\sigma(q_k, q_J) \geq 0$, and S_j is the number of individuals dominating q_j. This way of ranking takes into account both distance of a given point from Pareto frontier and number of points dominating given point. The disadvantage is that estimation of utopia and nadir points must be available to construct the achievement function, hence this ranking method cannot be used when approximating Pareto set for the first time. It is applicable only to further, interactive analysis of selected parts of Pareto set.

Despite such drawback, the ranking method based on achievement function values has several advantages. It is more sensitive than the classical Golberg ranking method and the Fonseca and Fleming method, which can be illustrated by the simple example (Fig. 16).

Another, more practical advantage of Pareto ranking using achievement function values is that it might improve the accuracy of the entire evolutionary algorithm. We have seen

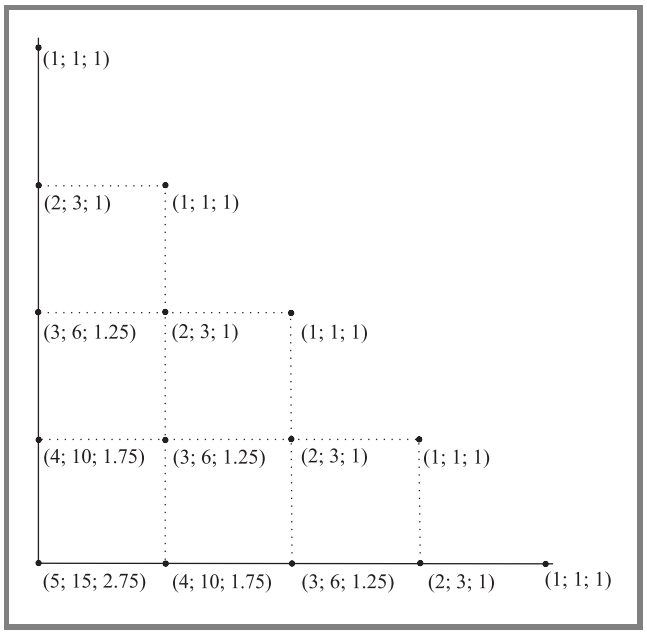


Fig. 16. Example of ranking values obtained by (1st value) Goldberg method; (2nd value) Fonseca and Flemming method; (3th value) by using achievement function values.

before that ranking methods did not behave well as succession mechanisms. A ranking method using achievement function values can perform much better: we can increase the accuracy of the entire evolutionary algorithm by increasing the value of the parameter ε, as suggested by the computation results shown in Table 2.

Table 2
Average distance from Pareto set after 30 000 generations depending on the parameter ε ($i = 2$, $\mu = 100$, $r = 0.01$)

ε	0	0.01	0.1	1	10	100
Average distance from Pareto set $[\cdot 10^{-3}]$	3.82	3.88	3.88	1.18	0.19	0.00012

We see that, using evolutionary algorithm interactively for more precise investigation of a part of Pareto set, we could actually obtain much better accuracy or use much shorter computation times for a ranking method based on achievement function values. On the other hand, very large values of ε (say, changing it from 10 to 100) mean only increasing the absolute value of achievement function, not its character that is dominated then by its linear part. This suggests that similar results would be obtained when using a slightly different form of the ranking formula:

$$rank_j^{(t)} = 1 + \beta \sum_{k=1}^{S_J} \sigma(q_k, q_j)\,, \tag{18}$$

while increasing the parameter β over its initial value 1.

Thus, *use of ranking values based on achievement functions not only increases flexibility of ranking, but also results in much better accuracy of approximating Pareto set.*

3.3. Neutral compromise points and their neighbourhoods

Given a reservation point q_{res} and an aspiration point q_{asp} in criteria value space, we can define a *relative neutral compromise point* as a point in Pareto set in criteria space being closest to the line joining points q_{res} and q_{asp} (Fig. 17).

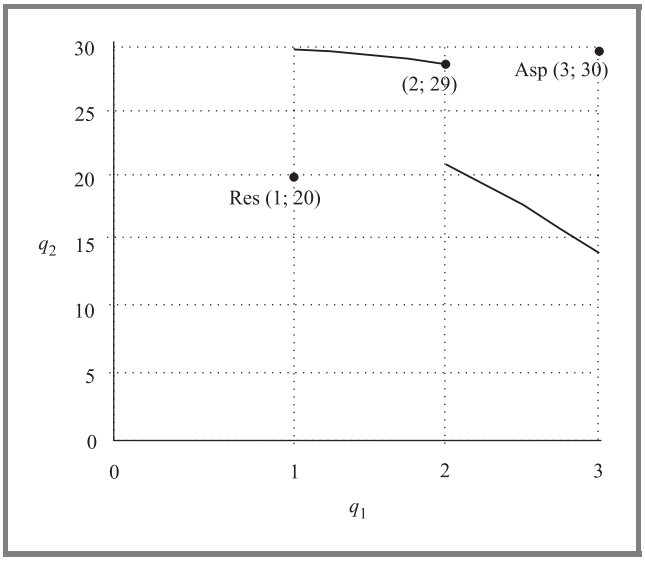


Fig. 17. Example of Pareto set with a reservation, aspiration and a relative neutral compromise points shown.

This point can be obtained by optimising the achievement function $\sigma(q, \overline{q})$ of the form (14) with partial achievement functions defined e.g. as follows:

$$\sigma_i(q_i, \overline{q}_i) = \frac{q_i - q_{asp.i}}{q_{asp.i} - q_{res.i}} \quad \text{(for maximised criteria)},$$
$$\sigma_i(q_i, \overline{q}_i) = \frac{q_{asp.i} - q_i}{q_{res.i} - q_{asp.i}} \quad \text{(for minimised ones)}. \tag{19}$$

For more sophisticated forms of partial achievement functions see e.g. [11]. In evolutionary algorithms, we can use the achievement function $\sigma(q, \overline{q})$ as a fitness measure and thus optimise it.

This results in interactive modification of evolutionary algorithms for vector optimisation: the user defines the aspiration and reservation points, the algorithms responds with the relative neutral compromise point or its approximation by a population of points (Table 3). This idea is illustrated by the following example. For the vector optimisation problem defined by Eq. (5), we define reservation and aspiration points as in Fig. 17. The line joining points q_{res} and q_{asp}

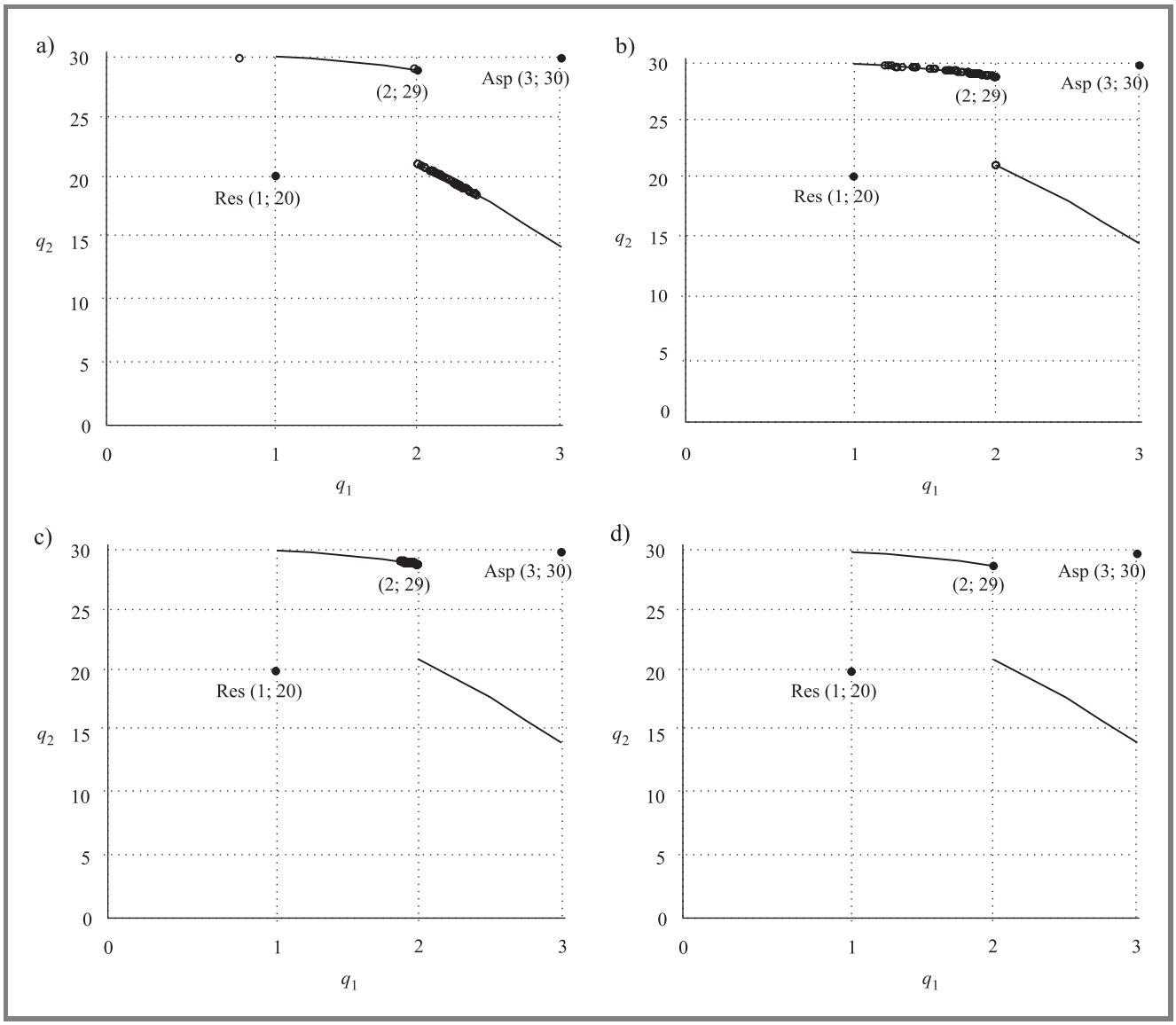


Fig. 18. The approximation of the relative neutral compromise point for the example from Fig. 17. Approximation cloud depending on generation number (a) $n = 5$; (b) $n = 10$; (c) $n = 15$; (d) $n = 30$ ($\mu = 50$, $\lambda = 50$).

Table 3
Diameter of approximation cloud depending on generation number ($\mu = 50, \lambda = 50$)

Generation number	5	10	20	40	60
$\Delta_{q_1}[\cdot 10^{-3}]$	1020	850	45.95	1.82	0.07
$\Delta_{q_2}[\cdot 10^{-3}]$	10640	9230	91.35	3.63	0.16

does not intersect Pareto set, but this makes the example more interesting. An evolutionary algorithm with achievement function used as a fitness measure produces a population approximating the relative neutral compromise point (2, 29) in the criteria space at first, and soon converges to this point (Fig. 18).

3.4. Parameterisation of representing Pareto set or its segment

The approach discussed above can be further parameterised combining a niched method with ranking based on achievement function. The niched method was originally used to provide a uniform representation of Pareto set in a global approach; here we use it to parameterise a local approach. Size of the niche can be related to e.g. the distance between aspiration and reservation points. Use of the niched method results in broadening the dispersion of a population around a neutral compromise point, as illustrated in Fig. 19.

We conclude that the evolutionary algorithms of vector optimisation, though traditionally understood as global and having non-interactive, batch character, can nevertheless be localised and used as local tools of interactive multiple criteria analysis.

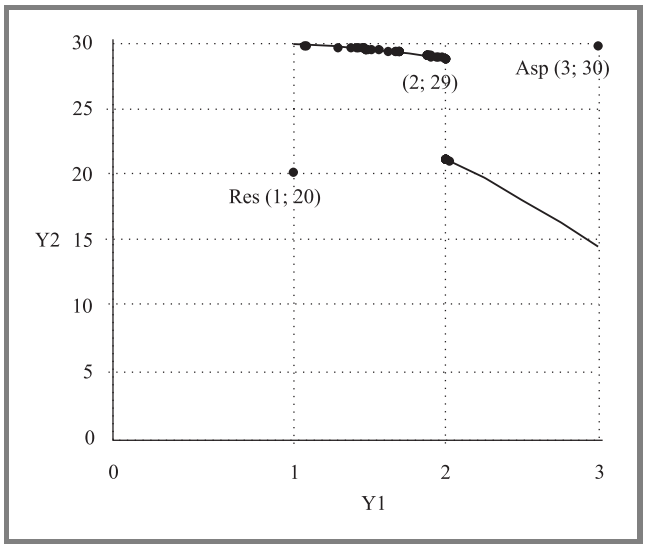


Fig. 19. Dispersion around the relative neutral compromise point for the example from Fig. 17, resulting from a niched approach with niche size equal to 10% of the range between aspiration and reservation points (population size 50, 200 generations).

4. Estimation of utopia and nadir points in evolutionary algorithms

4.1. Definitions and classical computations of utopia and nadir points

We recall that the *utopia point* q^U is defined as the "lowest" point dominating entire outcome set Q_0 (and thus entire Pareto set $\widehat{Q}_0$) in criteria value space. In other words, if some criteria are maximised and other minimised, we define:

$$q_i^U = \max_{x\in X_0} f_i(x),\ i = 1,\ldots,k_1 \quad \text{(for maximised criteria)},$$
$$q_i^U = \min_{x\in X_0} f_i(x),\ i = k_1+1,\ldots,k \quad \text{(for minimised criteria)}, \tag{20}$$

where $f_i(x)$ are criteria functions, k is the number of them (while k_1 is the number of maximised criteria), X_0 is the set of admissible decisions and $Q_0 = f(X_0)$ is the outcome set of attainable criteria vectors.

The *nadir point* is defined as the "highest" point in criteria value space dominated by the entire Pareto set $\widehat{Q}_0$ – and not necessarily the entire outcome set Q_0. This difference explains the difficulty (see e.g. [9]) of precisely calculating the nadir point, since we must perform necessary computations not over entire X_0 or Q_0, but over their efficient subsets $\widehat{X}_0$ or $\widehat{Q}_0$. Thus, if some criteria are maximised and other minimised, we define:

$$q_i^N = \min_{x\in\widehat{X}_0} f_i(x) = \min_{q\in\widehat{Q}_0} q_i,\ i = 1,\ldots,k_1 \quad \text{(for maximised criteria)},$$
$$q_i^N = \max_{x\in\widehat{X}_0} f_i(x) = \max_{q\in\widehat{Q}_0} q_i,\ i = k_1+1,\ldots,k \quad \text{(for minimised criteria)}. \tag{21}$$

We cannot replace $\widehat{Q}_0$ with Q_0 in the equation above, because this might lead to nadir estimation much lower than actual values. On the other hand, computation of precise value of the nadir point is very difficult when using classical methods. There are many methods that approximate nadir point components; the simplest of them is based on using only results of computations related to determining utopia components as in (20) and selecting the worst criteria values encountered during these computations:

$$q_i^U = \max_{x\in X_0} f_i(x),\ \widehat{q}_i = \arg\max f_i(x),\ i = 1,\ldots,k_1 \quad \text{(for maximised criteria)},$$
$$q_i^U = \min_{x\in X_0} f_i(x),\ \widehat{q}_i = \arg\min f_i(x),\ i = k_1+1,\ldots,k \quad \text{(for minimised criteria)},$$
$$q_i^N = \min_{1\le j\le k} \widehat{q}_i^j,\ i = 1,\ldots,k_1 \text{(for maximised criteria)},$$
$$q_i^N = \max_{1\le j\le k} \widehat{q}_i^j,\ i = k_1+1,\ldots,k \text{ (for minimised criteria)}, \tag{22}$$

where q_j denotes the jth component of vector q. This method is accurate if $k = 2$, for bi-criteria problems. However, in other cases it usually gives too optimistic estimations of the nadir value.

Matthias Ehrgott and Dagmar Tenfelde-Podehl [9] have proposed an algorithm computing the nadir point for three (or more) criteria by determining the Pareto sets for (each possible pair of) two criteria. For these bi-criteria Pareto sets, the values of the third missing criterion are attached, the resultant three-dimensional vectors are collected in one set, dominated results deleted, and the nadir values are directly computed from the resulting approximation of Pareto set.

4.2. Evolutionary algorithms and utopia and nadir points

Although the literature on evolutionary and genetic algorithms for vector optimisation is rather rich, it is focused more on the algorithms details than on their use for analysing Pareto set. Thus, an obvious fact was practically overlooked: since we approximate entire Pareto set by an evolutionary algorithm, the computations of utopia and nadir points should be much more easy than when using classical vector optimisation algorithms and should be actually by-products of the evolutionary algorithm applied. The questions that should be investigated are "only" how to provide for necessary accuracy of estimating these points –

especially the nadir point – while limiting the computational effort necessary for this estimation. We shall show on an example that these questions are by no means trivial.

We consider a slightly modified example from [3]:

$$\begin{aligned} \text{maximise}: f_1(x) &= 100 - 7x_1 - 20x_2 - 9x_3 \\ \text{maximise}: f_2(x) &= 4x_1 + 5x_2 + 3x_3 \\ \text{maximise}: f_3(x) &= x_3 \\ 1\tfrac{1}{2}x_1 + x_2 + 1\tfrac{3}{5}x_3 &\le 9 \\ x_1 + 2x_2 + x_3 &\le 10 \\ x_i \ge 0,\ i &= 1,2,3. \end{aligned} \tag{23}$$

The set of admissible decisions X_0 is illustrated by Fig. 20.

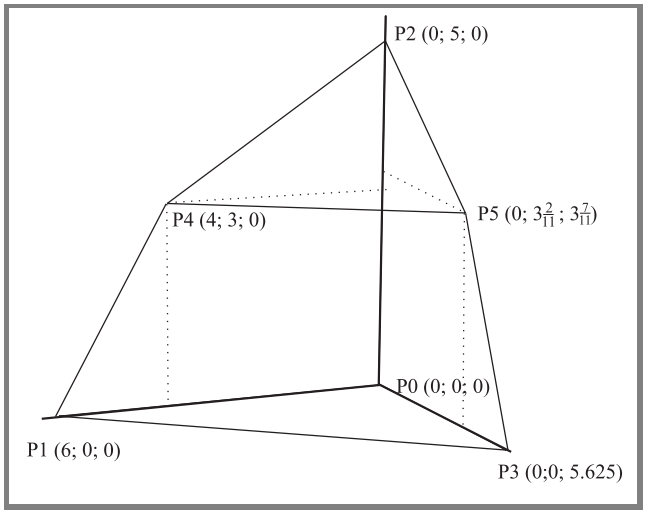


Fig. 20. Set of admissible decisions X_0 for the example defined by Eq. (23).

The set of admissible decisions X_0 is determined by its corner points:

$$\begin{gathered} \Big\{ \mathrm{P0} = (0;0;0),\ \mathrm{P1} = (6;0;0), \\ \mathrm{P2} = (0;5;0),\ \mathrm{P3} = \big(0;0;5\tfrac{5}{8}\big), \\ \mathrm{P4} = (4;3;0),\ \mathrm{P5} = \big(0;3\tfrac{2}{11};3\tfrac{7}{11}\big) \Big\}. \end{gathered}$$

Following the transformation $q = f(x)$ determined by Eq. (23), we can define also the corresponding corner points of the set of attainable criteria values Q_0 (Fig. 21). By direct examination, we can eliminate some of them as not belonging to Pareto set.

We can show in this way that the Pareto set is composed of surfaces determined by the following points in criteria space:

$$\begin{gathered} \Big\{ \mathrm{P0} = (100;0;0),\ \mathrm{P1} = (58;24;0), \\ \mathrm{P3} = \big(49\tfrac{3}{8};16\tfrac{7}{8};5\tfrac{5}{8}\big) \Big\} \quad \text{and} \\ \Big\{ \mathrm{P1} = \big(58;24;0\big),\ \mathrm{P3} = \big(49\tfrac{3}{8};16\tfrac{7}{8};5\tfrac{5}{8}\big), \\ \mathrm{P4} = (12;31;0),\ \mathrm{P5} = \big(3\tfrac{7}{11};26\tfrac{9}{11};3\tfrac{7}{11}\big) \Big\}. \end{gathered}$$

By direct examination, we can find for these points the utopia point $q^U = \big(100;31;5\tfrac{5}{8}\big)$ and the nadir point $q^N = \big(3\tfrac{7}{11};0;0\big)$. Now we shall show the results of computing these points via three variants of evolutionary algorithms.

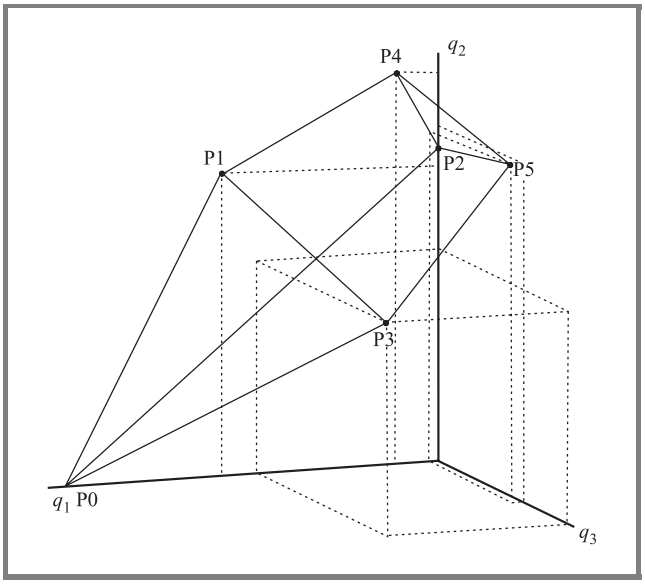


Fig. 21. The set of attainable criteria values Q_0 for the example defined by Eq. (23).

I. Evolutionary computations of utopia point with utopia based nadir approximations

The first variant uses direct determination – see Eq. (19) – of utopia point for an evolutionary approximation of a Pareto set and an indirect – see Eq. (21) approximation of the nadir point based on the data obtained in utopia point determination. An evolutionary algorithm with $(\mu, \lambda) = (200, 100)$ and 200 generations gave the following results:

$$q_1^U = (100;\ 0;\ 0)$$

$$q_2^U = (11.9998;\ 30.9999;\ 0) \Rightarrow q^U = (100;\ 30.9999;\ 5.625)$$

$$q_3^U = (49.375;\ 16.875;\ 5.625)$$

with the corresponding quite inaccurate nadir approximation $q^N = (11.9998;\ 0;\ 0)$. By increasing the computing effort (measured below as the number of new computations of criteria values, because this, rather than organisation of the algorithm determines the computational effort) we can increase the accuracy of utopia approximations, but accuracy of nadir approximations remains inadequate, as shown in Table 4. Thus, we conclude that this method of nadir approximations is not worth using with evolutionary algorithms.

Table 4
Results of utopia and nadir point approximation by method I

The number of new computations of criteria values	Nadir point approximation	Utopia point approximation
30 000	(12.06; 0; 0)	(100.0; 30.79; 5.617)
60 000	(11.97; 0; 0)	(100.0; 30.95; 5.624)
120 000	(12.00; 0; 0)	(100.0; 31.00; 5.625)

II. Evolutionary computations of utopia point and nadir point

Since an evolutionary algorithm approximates entire Pareto set, we can also simply determine utopia and nadir points directly, according to their definitions, for the subsequent evolutionary approximations of Pareto set (Fig. 22). This simple method needs not, however, be the best, since a uniform approximation of Pareto set does not necessarily cover well the remote corners of this set, which are responsible for utopia and nadir points.

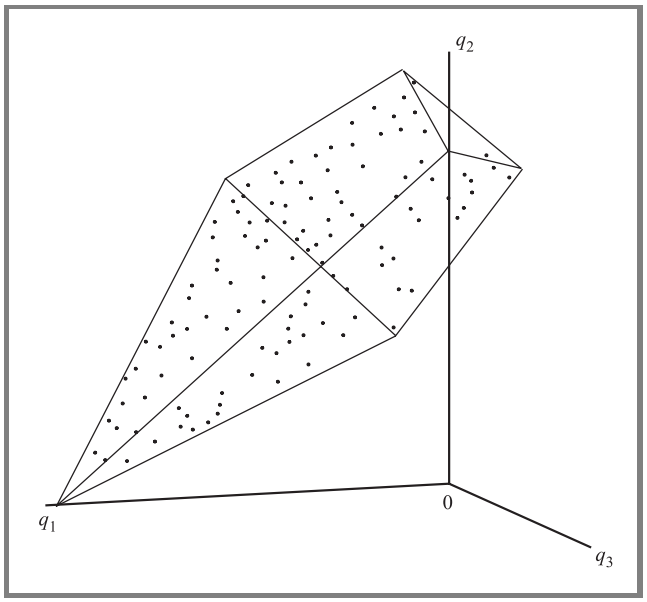


Fig. 22. Approximation of the Pareto set for the example defined by Eq. (23).

Thus, an evolutionary algorithm for vector optimisation must be modified in order to provide for a good approximation of utopia point and particularly the nadir point. It is necessary to increase fitness indicators for individuals with extreme values of criteria components.

Theoretically, such a method should give good approximations of Pareto set together with its utopia and nadir points. However, practical applications show that good approximations of the nadir point remain difficult to obtain. This is illustrated by results (Table 5) of an evolutionary algorithm with direct determination of nadir point for Pareto set approximations in subsequent iterations, with a modification of fitness indicators for individuals with extreme values of criteria vectors components. We observe that accuracy of the nadir point approximation, although much better than in method I, still remains inadequate even after very long computations.

Table 5
Results of nadir point approximation by method II

The number of new computations of criteria values	Nadir point approximation
Arbitrary starting population	
30 000	(6.78; 0; 0)
60 000	(5.90; 0; 0)
120 000	(5.91; 0; 0)
Starting population containing individuals responsible for utopia point	
30 000	(5.38; 0; 0)
60 000	(5.24; 0; 0)
120 000	(5.06; 0; 0)

III. Evolutionary approximations of Pareto sets for smaller number of criteria

The method proposed by Ehrgott and Tenfelde-Podehl [9] was not developed as an evolutionary algorithm, but can be easily combined with evolutionary approaches, involving the following steps:

- for each pair of criteria, Pareto sets are be approximated by using an evolutionary algorithm;
- for each individual in these approximations, the corresponding values of other criteria are computed;
- results obtained this way are combined and dominated points deleted, resulting in an approximation of Pareto set for the original problem;
- utopia and nadir points are computed according to their definitions for this approximation of Pareto set.

The advantage of this method over method II is that approximation of Pareto sets for bi-criteria problems in a natural way provides for more attention paid to extreme values of criteria components.

We illustrate the working of this method by showing the results of such approximations obtained by using an evo-

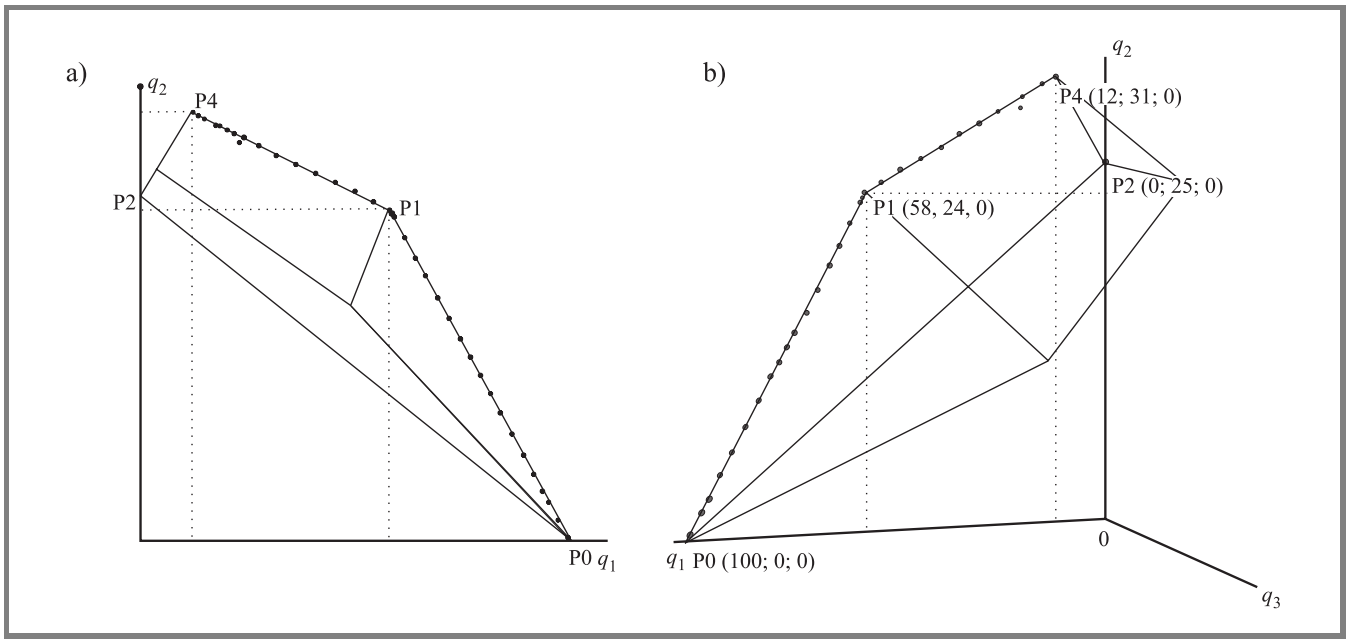


Fig. 23. Approximation of Pareto set for criteria 1 and 2 (a) with three-dimensional presentation (b).

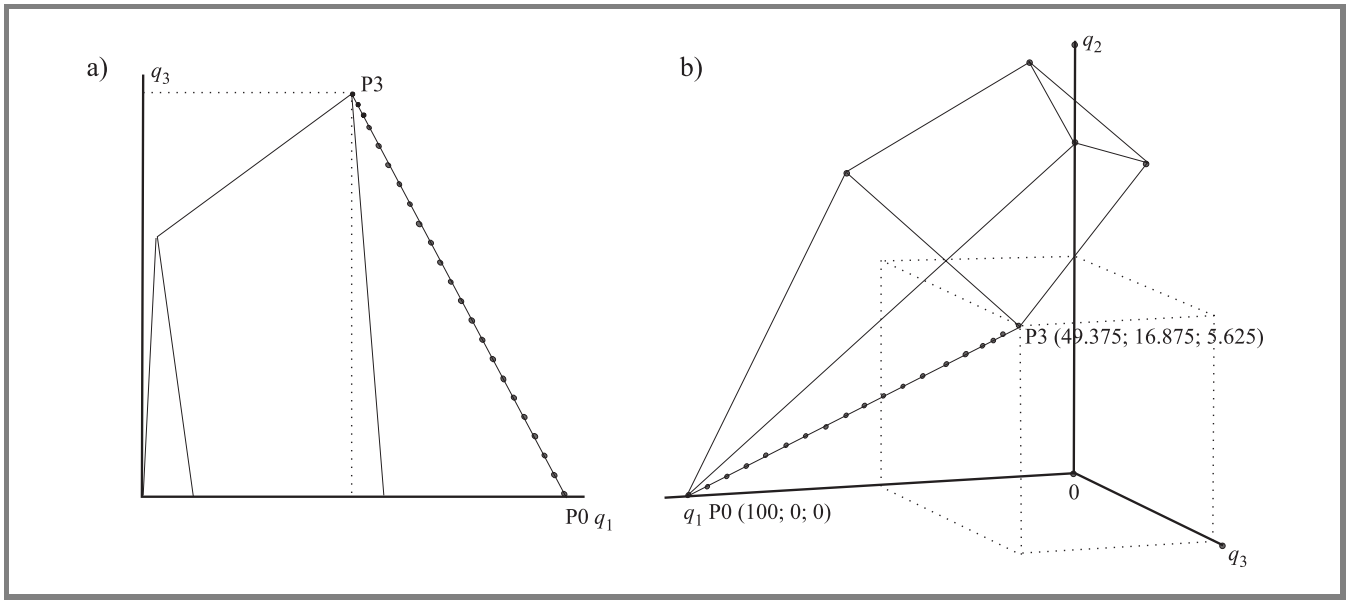


Fig. 24. Approximation of Pareto set for criteria 1 and 3 (a) with three-dimensional presentation (b).

lutionary algorithm with $(\mu, \lambda) = (200, 100)$, 200 generations and alternative niche diameters 4,75; 1,55; 0,28) – see Figs. 23–25.

This way, after a large number (360 000) of computations of new criteria vectors, the following approximations were obtained: utopia point $q^U = (100; 30.999; 5.625)$ and nadir point $q^N = (4.36; 0; 0)$. We see that nadir point approximation, though much better than in other methods, still remains inadequate. Moreover, method III requires more computations (three times in this case) than methods II and I, and a fair way of comparing them is to compare nadir approximations after the same number of computations of new criteria vectors. Such a comparison is presented in Table 6.

When we compare the results of these three methods, we see that method III is most promising. The example defined by Eq. (23) might be especially difficult for nadir point approximation, hence we tried another variant

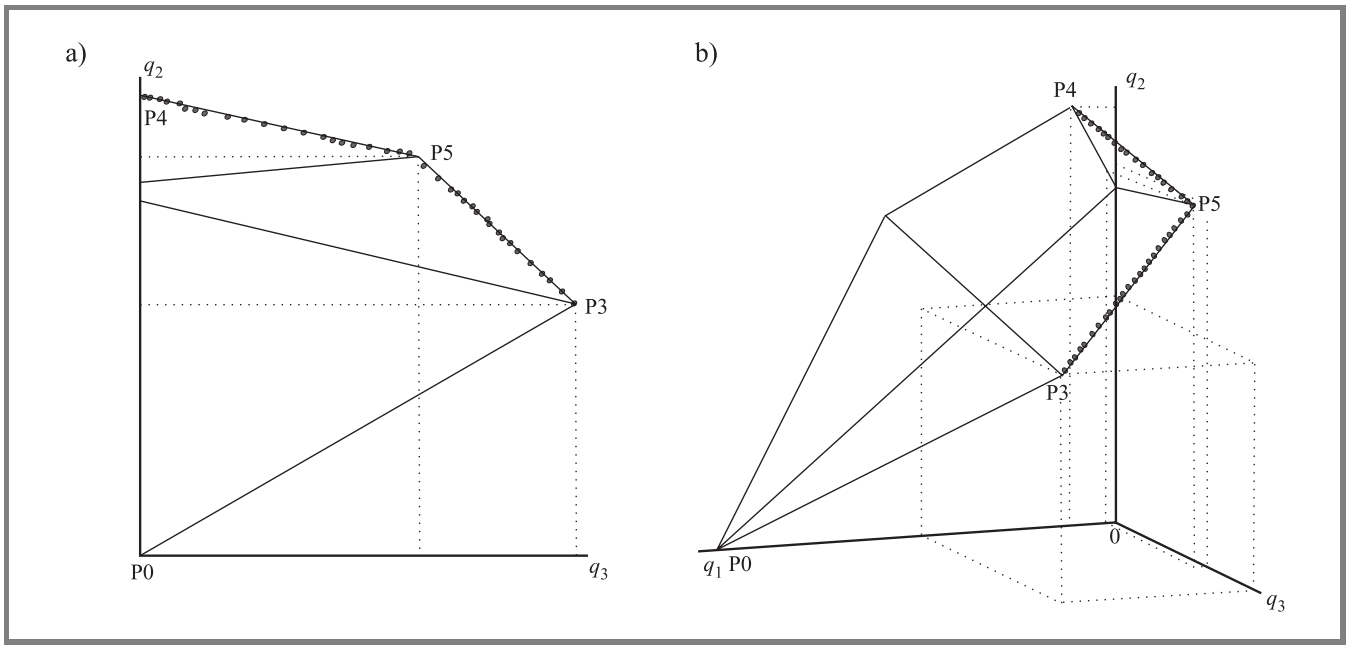


Fig. 25. Approximation of Pareto set for criteria 2 and 3 (a) with three-dimensional presentation (b).

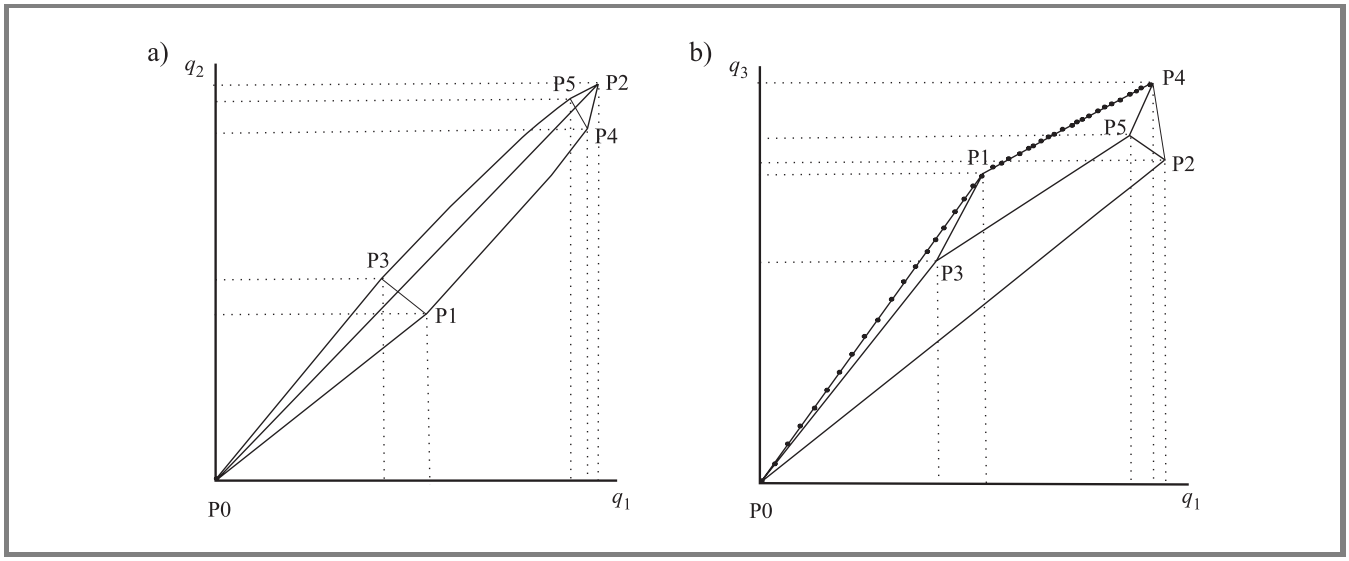


Fig. 26. Pareto sets for criteria 1 and 2 (a) or for criteria 1 and 3 (b), for example defined by Eq. (24).

Table 6
Results of nadir point approximation by method III

Number of new computations of criteria vectors	Nadir point approximation
30 000	(5.01; 0; 0)
60 000	(4.67; 0; 0)
120 000	(4.78; 0; 0)

of this example, at the same time testing the possibility of generalising method III for a larger number of criteria.

The original example from [3] is as follows:

$$\begin{aligned}
&\text{minimise}: f_1(x) = 9x_1 + 19\tfrac{1}{2}x_2 + 7\tfrac{1}{2}x_3\\
&\text{minimise}: f_2(x) = 7x_1 + 20x_2 + 9x_3\\
&\text{maximise}: f_3(x) = 4x_1 + 5x_2 + 3x_3\\
&\text{maximise}: f_4(x) = x_3\\
&1\tfrac{1}{2}x_1 + x_2 + 1\tfrac{3}{5}x_3 \le 9\\
&x_1 + 2x_2 + x_3 \le 10\\
&x_i \ge 0,\ i = 1, 2, 3.
\end{aligned} \tag{24}$$

The set of admissible decisions X_0 is the same as in the example defined by Eq. (23) – see Fig. 20. However,

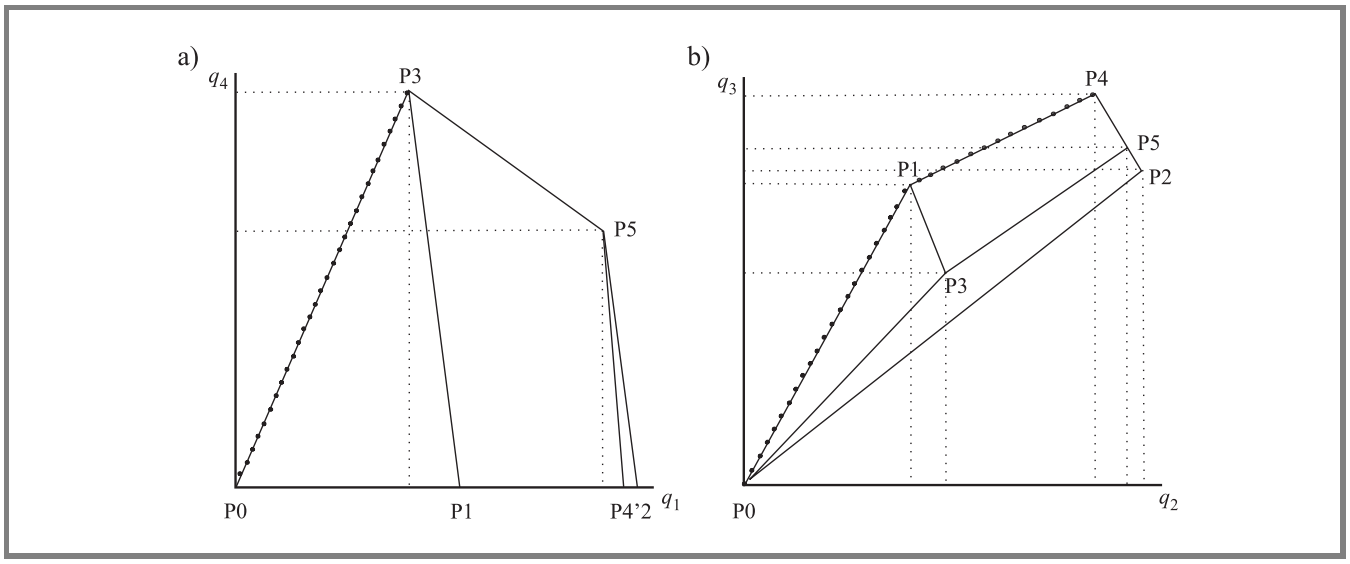


Fig. 27. Pareto sets for criteria 1 and 4 (a) or for criteria 2 and 3 (b), for example defined by Eq. (24).

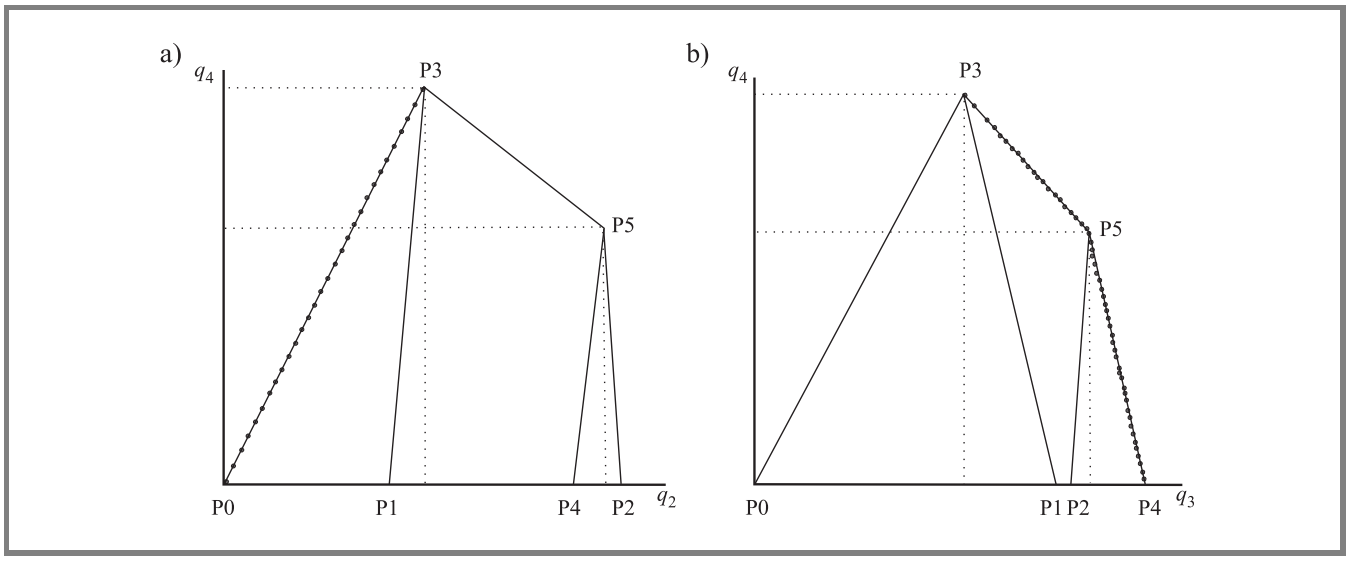


Fig. 28. Pareto sets for criteria 2 and 4 (a) or for criteria 3 and 4 (b), for example defined by Eq. (24).

utopia and particularly nadir points change with each change of criteria and they are here $q^U = \left(0; 0; 31; 5\frac{5}{8}\right)$ and $q^N = \left(94\frac{1}{2}; 96\frac{4}{11}; 0; 0\right)$. The Pareto sets for consecutive bi-criteria problems are shown in Figs. 26–28.

Utopia and nadir points obtained using evolutionary algorithm with $(\mu, \lambda) = (200, 100)$ and 200 generations, and a version of method III for four criteria: $q^U = (0; 0; 30.999; 5.625)$ and $q^N = (94.4998; 95.8747; 0; 0)$. Although the actual number of criteria value computations here increased 6 times (this is the drawback of using method III), we have obtained quite acceptable approximation of utopia and nadir points in this example.

5. Conclusions and future research

We shall point out only few conclusions, in particular those concerning future research:

- Although there is a very rich literature on evolutionary algorithms for vector optimisation, this literature focuses mostly on the tool – specific aspects of evolutionary algorithms, much less on the task – specific issues of vector optimisation, for which an evolutionary approach might be helpful.
- When concentrating on the task, evolutionary algorithms might be usefully extended – e.g. to obtain more precise approximations of selected parts

of Pareto set, or better approximations of utopia and nadir points of Pareto set.

- In such extensions of evolutionary algorithms, an essential issue is to make them more interactive (e.g. first approximating entire Pareto set, then a selected part of it). For interactive extensions of evolutionary algorithms, combining them with reference point approaches and achievement function concepts might be useful.
- A particularly difficult issue (not only for evolutionary algorithms, but also in entire vector optimisation) is the determination of nadir points. Classical evolutionary approaches are not sufficient to solve this issue. Combinations of evolutionary algorithms with other approaches of vector optimisation are necessary.
- Many issues outlined in this paper should be treated as starting points only and require deeper future research. Starting from a different perspective, concentrating more on tasks than on tools, the paper serves only as identification of future research issues.

References

[1] K. Deb, "Non-linear goal programming using multi-objective genetic algorithms". Tech. Rep. no. CI-60/98, Department of Computer Science, University of Dortmund, 1998.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Mass.: Addison-Wesley, 1989.

[3] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization", *Evol. Comput.*, vol. 3, no. 1, 1995.

[4] C. M. Fonseca and P. J Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization" in *Proc. Fifth Int. Conf. Genet. Algor.*, S. Forrest, Ed., San Mateo, USA, 1993, University of Illinois at Urbana – Champaign, Morgan Kauffman, pp. 416–423.

[5] J. Horn, *Multicriterion Decision Making*. Handbook of Evolutionary Computation, IOP & Oxford University Press, 1997.

[6] J. Horn and N. Nafpliotis, "Multiobjective optimization using the niched Pareto genetic algorithm" in *Proc. First IEEE Conf. Evol. Comput., IEEE World Congr. Comput. Intell.*, vol. 1, 1994.

[7] P. Korhonen, "Multiple objective programming support". IR-98-010, International Institute for Applied Systems Analysis, Laxenburg, 1998.

[8] Z. Kowalczuk, T. Białaszewski, and P. Suchomski, "Genetic polioptimisation in Pareto sense with ranking and niched methods" in *Proc. III Nat. Conf. Evol. Algor. Glob. Optim.*, Warsaw, Poland, 1999.

[9] M. Ehrgott and D. Tenfelde-Podehl, "Nadir values: computation and use in compromise programming". Universitat Kaiserslautern Fachbereich Mathematik, 2000.

[10] D. A. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: a history and analysis". Tech. Rep. TR-98-03, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[11] A. P. Wierzbicki, M. Makowski, and J. Wessels, *Model-Based Decision Support Methodology with Environmental Applications*. Dordrecht – Laxenburg: Kluwer – IIASA, 2000.

[12] E. Zitzle, "Evolutionary algorithms for multiobjective optimization: methods and applications". Swiss Federal Institute of Technology (ETH), Zurich, 1999.

Marcin Szczepański was born in August 10, 1976. He has been graduated as Master of Engineering at Warsaw University of Technology at the Institute of Control and Computation Engineering at Optimization and Decision Support Division, in 2001. During his studies, he was especially interested in evolutionary algorithms and decision support.

e-mail: Marcin.Szczepanski@damovo.com
Damovo Poland
Jana Olbrachta st 94
01-102 Warsaw, Poland

Andrzej Piotr Wierzbicki born June 29, 1937 in Warsaw. Graduated as Master of Engineering at the Faculty of Electronics, Warsaw University of Technology (WUT), in 1960. Ph.D. degree at this University in 1964, for a thesis on nonlinear feedback systems; D.Sc. degree in 1968, for a thesis on optimisation of dynamic systems. In 1971–75 a Deputy Director of the Institute of Automatic Control, later a Deputy Dean of the Faculty of Electronics, WUT. In 1975–78 the Dean of the Faculty of Electronics WUT. Since 1978 worked with the International Institute for Applied Systems Analysis in Laxenburg n. Vienna, Austria; 1979–84 as the chairman of the theoretical branch, Systems and Decision Sciences Program, of this Institute. From 1985 back in the Institute of Automatic Control, WUT, as a Professor of optimisation and decision theory. In 1986–91 scientific secretary, currently member of presidium of the Committee of Future Studies "Poland 2000" (in 1990 renamed "Poland in XXI Century") of P.Ac.Sc. In 1991 elected a member of the State Committee for Scientific Research of Republic of Poland and the chairman of its Commission of Applied Research; contributed to basic reforms of Polish scientific system in 1991–94. Deputy chairman of the Council of Polish Foundation for Science in 1991–94, chairman of scientific councils of NASK (National Scientific and Academic Computer Network in Poland) and PIAP (the Industrial Institute of Measurements and Control). In 1991–96 the editor in chief of the quarterly "Archives of Control Sciences" of P.Ac.SC. In 1992 received (as first European researcher) the George Cantor Award of the International Society of Multiple Criteria Decision Making for his contributions to the theory of multiple criteria optimisation and decision support. Since 1996 the General Director of the National Institute of Telecom-

munications in Poland. In 2000 nominated as a member of the ISTAG (Information Society Technology Advisory Group) at European Commission. Since 2001 chairman of Advisory Group on Scientific International Cooparation of the State Committee for Scientific Research of Poland. Beside lecturing for over 40 years and promoting more than 80 master's theses at WUT (Warsaw University of Technology), he also lectured at the Department of Mathematics, Information Science and Mechanical Engineering of Warsaw University and in doctoral studies: at WUT, the Academy of Mining and Metallurgy, at the University of Minnesota, at the Illinois Technical University, Hagen University, and at the University of Kyoto. He also promoted 18 completed doctoral dissertations. Author of over 180 publications, including 11 books (4 monographs, 7 – editorship or co-authorship of international joint publications, over 50 articles in scientific journals (over 30 in international), 80 papers at conferences (68 at international, including over 48 published as chapters in books). He also authored 3 patents granted and applied industrially. Current interests include parallelisation of optimisation algorithms using multiple criteria approaches, diverse aspects of negotiation and decision support, including e.g. applications of fuzzy set theory for describing uncertainty in decision support models, multimedia software in computer networks, telematics in education, diverse issues of information society and civilisation. Languages: English, German, Russian (each fluent, beside native Polish). Member of IEEE, ISMCDM (International Society of Multiple Criteria Decision Making), SEP (Polish Society of Electrical Engineers), PTM (Polish Mathematical Society), PSKR (Polish Association for the Club of Rome).
e-mail: A.Wierzbicki@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Paper

Fair resource allocation schemes and network dimensioning problems

Włodzimierz Ogryczak, Tomasz Śliwiński, and Adam Wierzbicki

Abstract — Resource allocation problems are concerned with the allocation of limited resources among competing activities so as to achieve the best overall performances of the system but providing fair treatment of all the competitors. Telecommunication networks are facing the increasing demand for Internet services. Therefore, a problem of network dimensioning with elastic traffic arises which requires to allocate bandwidth to maximize service flows with fair treatment of all the services. In such applications, the so-called max-min fairness (MMF) solution concept is widely used to formulate the resource allocation scheme. This guarantees the fairness but may lead to significant losses in the overall throughput of the network. In this paper we show how multiple criteria optimization concepts can be used to generate various fair resource allocation schemes. The solution concepts are tested on the network dimensioning problem and their abilities to model various preferences are demonstrated.

Keywords — telecommunication networks, network dimensioning, resource allocation, fairness.

1. Introduction

Resource allocation decisions are concerned with the allocation of limited resources so as to achieve the best system performances. In this paper, we focus on approaches that, while allocating resources, attempt to provide a fair (equal) treatment of all the competing activities [8, 13]. The problems of efficient and fair resource allocation arise in various systems which serve many users, like in telecommunication systems among others [8].

The development of the Internet has led to an increased role of the traffic carried by the IP protocol in telecommunication networks. Due to the use of packet switching, the IP protocol can provide greater network utilization (the so-called multiplexing gain). For these reasons, network management can be interested in designing networks which have a high throughput for the IP protocol.

At the same time, data traffic carried by the TCP protocol (which is the most frequently used transport protocol in IP networks) has a unique characteristic. The TCP protocol will adapt its throughput to the amount of available bandwidth. It is therefore capable to use the entire available bandwidth, but it will also be able to reduce its throughput in the presence of contending traffic. This type of network traffic has been called *elastic traffic*.

Network design today often considers the problem of designing networks that carry elastic traffic. If the network is also used for other types of communication that require guaranteed quality of service, the network design problem can be decomposed into two parts: first, design the network to carry non-elastic traffic in such a way that all demands for that communication are satisfied. Next, use the spare capacity to carry elastic traffic of the IP protocol. Resource allocation models may be used to help to solve such network design problems.

Within a telecommunication network the data traffic is generated by a huge number of nodes exchanging data. In such a network, a relatively small subset of nodes are chosen to serve as hubs which can be used as intermediate switching points [2, 6]. Given a set of hubs, data traffic generated by a service is sent from the source node to a hub first. It can be then sent along communications link between hubs, and finally reach the destination node along a link from a hub. The hub-based network organization allows the data traffic to be consolidated on the inter-hub links. The problem of network dimensioning with elastic traffic arises when there is a need to design the (inter-hub) link capacities to carry as much traffic as possible between a set of network nodes. This can occur in the case described above, when the network capacity available after considering all non-elastic demands has to be used for elastic traffic, or in another case: when the network capacity is insufficient to carry all non-elastic demands. In such a case, the problem is to determine how much traffic of the non-elastic demands can be admitted into the network. To do so, the demands can be treated as elastic traffic. The outcome of network design will also specify the limits of traffic to be admitted into the network for each demand [16].

Network management must stay within a budget of expenses for purchasing link bandwidth. Network management will want to have a high throughput of the IP network, to increase the multiplexing gains. This traffic is offered only a best-effort service, and therefore network management is not concerned with offering guaranteed levels of bandwidth to the traffic. Network dimensioning with elastic traffic can therefore be thought of as a search for such network flows that will maximize the network throughput (the sum of all flows in the network) while staying within a budget constraint for the costs of link bandwidth. However, such a problem formulation would lead to the starvation of flows between certain network nodes.

Looking at the problem from the user perspective, the network flows between different nodes should be treated as fairly as possible. The users may be interested in high available bandwidth between any two nodes of the network, or in high available bandwidth from all other network nodes to the user's node, or in high available bandwidth from the user's node to all other nodes. Whatever the user prefer-

ence, it would be expressed in terms of fairness for a certain set of criteria which depend on the individual flows. Let us first consider providing fairness for all flows between any two network nodes. Such a goal would clearly lead to lower levels of throughput, since resources must be allocated to distant nodes, which is more expensive than using the entire budget to purchase a high capacity for close nodes.

Therefore, network management must consider two goals: increasing throughput and providing fairness. These two goals are clearly conflicting, if the budget constraint has to be satisfied. Network management could therefore be interested in finding compromise solutions that do not starve network flows, and give satisfying levels of throughput.

The search for such compromise solutions has led to the development of a method that finds solutions which are fair with respect to flows in certain categories. These categories can depend on the distance between the source and destination of a flow. The details of this method will be given below; it is referred to as proportional fairness (PF) [5]. However, this method gives only one possible compromise solution. The purpose of this work is to show that there exists a methodology that allows the decision maker to explore a set of solutions that could satisfy his preferences with respect to throughput and fairness, and choose the solution which the decision maker finds best. This interactive approach to decision making is superior to a black box approach, when the decision maker has only one solution and cannot express his preferences [18].

The paper is organized as follows. In the next section we recall the network dimensioning problem. In Section 3, basic fair solution concepts for resource allocation are formally introduced. In the next section, the ordered outcomes are used to introduce LP implementable solution concepts allowing to model various fair allocation schemes. Finally, in Section 5, we report some results of our initial computational experience with this new approach.

2. The network dimensioning problem

The generic resource allocation problem may be stated as follows. There is given a set I of m services. There is also given a set Q of allocation patterns (allocation decisions). For each service $i \in I$ a function $f_i(\mathbf{x})$ of the allocation pattern $\mathbf{x}$ has been defined. This function, called the individual objective function, measures the outcome (effect) $y_i = f_i(\mathbf{x})$ of the allocation pattern for service i. In applications, we consider, an outcome usually expresses the service flow. However, outcomes can be measured (modeled) as service time, service costs, service delays as well as in a more subjective way. In typical formulations a larger value of the outcome means a better effect (higher service quality or client satisfaction). Otherwise, the outcomes can be replaced with their complements to some large number. Therefore, without loss of generality, we can assume that each individual outcome y_i is to be maximized which results in a multiple criteria maximization model.

The problem of network dimensioning with elastic traffic can be formulated as a linear programming (LP) resource allocation problem as follows. Given a network routing topology $G =< V, E >$, consider a set of pairs of nodes as the set I of services. For each service $i \in I$, the elastic flow from source u_i^s to destination u_i^d will be denoted by y_i, which is a variable representing the model outcome. For each service, we have given the information about the routing path in the network from the source to the destination. This information can be in the form of a matrix $\mathbf{A} = (a_{ei})$, which satisfies the relation: $a_{ei} = 1$ if link e belongs to the routing path connecting u_i^s with u_i^d. Further, for each link $e \in E$, marginal costs c_e of link bandwidths is given. Hence, the cost of the entire path for service i can be expressed as:

$$\kappa_i = \sum_{e \in E} c_e a_{ei}.$$

The network dimensioning problem depends on allocating the bandwidth to several links in order to maximize flows of all the services while remaining within available budget B for all link bandwidths. The decisions are usually modeled with (decision) variables: x_e – representing the bandwidth allocated to link $e \in E$. They have to fulfill the following constraints:

$$\sum_{e \in E} c_e x_e = B \tag{1}$$

$$\sum_{i} a_{ei} y_i = x_e \qquad \forall e \in E\,, \tag{2}$$

where Eq. (1) represents the budget limit while Eqs. (2) establish the relation between service flows and links bandwidth (the quantity $\sum_{i \in I} a_{ei} y_i$ is the load of link e). Certainly, all the decision and outcome variables must be nonnegative: $x_e \geq 0$ for all $e \in E$ and $y_i \geq 0$ for all $i \in I$. Alternatively, one may eliminate variables x_e formulating the problem as a simplified resource allocation model with only one constraint:

$$\sum_{i=1}^{m} \kappa_i y_i = B \tag{3}$$

and variables y_i representing directly decisions.

The model could have various objective functions, depending on the chosen approach. One may consider two extreme approaches. The first extreme approach is the maximization of the throughput (the sum of flows) $\sum_{i \in I} y_i$. Due to possible alternative formulation (3), it is apparent that this approach would choose one variable y_{i^o} which has the smallest marginal cost $\kappa_{i^o} = \min_{i \in I} \kappa_i$ and make that flow maximal within the budget limit ($y_{i^o} = B/\kappa_{i^o}$), while limiting all other flows to zero. A slightly more fair optimal solution would give equal values to all flows which have marginal costs equal to the minimal marginal cost. However, all flows that have marginal costs larger than the minimum would have to be zero in a solution that maximizes throughput.

The so-called max-min fairness solution concept is widely used to formulate fair resource allocation schemes [1, 8]. The worst performance (minimum flow) is there maximized

and additonally regularized, if necessary, with the lexicographic (sequential) maximization of the second worst performance, the third worst etc. The MMF concept is consistent with Rawlsian [15] theory of justice.

Actually, due to possible alternative formulation (3), the MMF concept would lead us to a solution that has equal values for all the flows [12]:

$$y_i^{MMF} = B/\sum_{i\in I} \kappa_i \quad \text{for } i = 1,\dots,m.$$

Allocating the resources to optimize the worst performances may cause a large worsening of the overall (mean) performances. In such a solution the throughput ($mB/\sum_{i=1}^{m}\kappa_i$) could be considerably smaller than the maximal throughput ($B/\min_{i=1,\dots,m}\kappa_i$). In an example analyzed further, we shall show that the throughput in a perfectly fair solution can be less than 50% of the maximal throughput.

Network management can be interested in seeking a compromise between the two extreme approaches discussed above. The approach called *proportional fairness* proposed in [5] maximizes the sum of logarithms of the flows y_i. The use of the logarithmic function makes it impossible to choose zero flows for any pair of nodes, and, on the other hand, makes it not profitable to assign too much flow to any individual demand. The optimization model of the PF method takes the following form:

$$\max \sum_{i=1}^{m} \log(y_i). \tag{4}$$

For the problem of network dimensioning with elastic traffic and unbounded flows, the solution found by the PF method has an interesting property [11]. The optimal flows y_i^{PF} are given by the expression:

$$y_i^{PF} = B/\kappa_i \quad \text{for } i = 1,\dots,m. \tag{5}$$

This property implies that the optimal flow in the PF model is inversely proportional to the cost of the path that the flow travels in the network. Due to this property, it is not necessary to solve nonlinear models in order to find the PF optimal solution. Also, the solution provides fairness to the flows which have the same path costs. Arguably, the PF solution is a good compromise solution to the problem, since it provides a higher throughput than the perfectly fair solution. However, network management could be interested in choosing among a larger set of compromise solutions in order to satisfy their preferences. In the following sections, we shall describe an approach that allows to search for such compromise solutions.

3. Basic fair allocation schemes

Consider a generic resource allocation problem defined as an optimization problem with m objective functions $f_i(\mathbf{x})$:

$$\max\{\mathbf{f}(\mathbf{x}) : \mathbf{x} \in Q\}, \tag{6}$$

where $\mathbf{f}(\mathbf{x})$ is a vector–function that maps the decision space $X = R^n$ into the criterion space $Y = R^m$, $Q \subset X$ denotes the feasible set, and $\mathbf{x} \in X$ denotes the vector of decision variables.

Model (6) only specifies that we are interested in maximization of all objective functions f_i for $i \in I = \{1,2,\dots,m\}$. In order to make it operational, one needs to assume some solution concept specifying what it means to maximize multiple objective functions.

Typical solution concepts for multiple criteria problems are defined by aggregation functions $g : Y \to R$ to be maximized. Thus the multiple criteria problem (6) is replaced with the maximization problem

$$\max\{g(\mathbf{f}(\mathbf{x})) : \mathbf{x} \in Q\}. \tag{7}$$

In order to guarantee the consistency of the aggregated problem (7) with the maximization of all individual objective functions in the original multiple criteria problem, the aggregation function must be strictly increasing with respect to every coordinate, i.e., for all $i \in I$,

$$g(y_1,\dots,y_{i-1},y_i',y_{i+1},\dots,y_m) < g(y_1,y_2,\dots,y_m) \tag{8}$$

whenever $y_i' < y_i$.

In order to guarantee fairness (equitability) of the solution concept, the aggregation function must be additionally symmetric (impartial), i.e. for any permutation τ of I,

$$g(y_{\tau(1)},y_{\tau(2)},\dots,y_{\tau(m)}) = g(y_1,y_2,\dots,y_m) \tag{9}$$

as well as be equitable (to satisfy the principle of transfers)

$$g(y_1,\dots,y_{i'}-\varepsilon,\dots,y_{i''}+\varepsilon,\dots,y_m) > g(y_1,y_2,\dots,y_m) \tag{10}$$

for any $0 < \varepsilon < y_{i'} - y_{i''}$. In the case of an aggregation function satisfying all the requirements (8), (9) and (10), we call the corresponding problem (7) a *fair (equitable) aggregation* of problem (6). Every optimal solution to the fair aggregation (7) of a resource allocation problem (6) defines some fair allocation scheme.

Note that symmetric functions satisfying the requirement

$$g(y_1,\dots,y_{i'}-\varepsilon,\dots,y_{i''}+\varepsilon,\dots,y_m) \geq g(y_1,y_2,\dots,y_m) \tag{11}$$

for $0 < \varepsilon < y_{i'} - y_{i''}$ are called (weakly) Schur-concave [10] while the stronger requirement of equitability (10), we consider, is related to strictly Schur-concave functions. In other words, an aggregation (7) is fair if it is defined by a strictly increasing and strictly Schur-concave function g.

The simplest aggregation functions commonly used for the multiple criteria problem (6) are defined as the sum of outcomes

$$g(\mathbf{y}) = \sum_{i=1}^{m} y_i \tag{12}$$

or the worst outcome

$$g(\mathbf{y}) = \min_{i=1,\dots,m} y_i. \tag{13}$$

In the network dimensioning problem, the former represents throughput maximization while the latter corresponds

to the MMF model. The sum (12) is a strictly increasing function while the minimum (13) is only non-decreasing. Therefore, the aggregation (7) using the sum of outcomes always generates a Pareto-optimal solution while the maximization of the worst outcome may need some additional refinement. Both the functions are symmetric and satisfy the requirement (11), although they do not satisfy the equitability requirement (10). Hence, they are Schur-concave but not strictly Schur-concave. To guarantee the fairness of solutions, some enforcement of concave properties is required.

For any strictly concave, increasing function $s : R \to R$, the function

$$g(\mathbf{y}) = \sum_{i=1}^{m} s(y_i) \tag{14}$$

is a strictly monotonic and strictly Schur-concave function [10]. This defines a family of the fair aggregations according to the following corollary [7].

Corollary 1. For any strictly convex, increasing function $s : R \to R$, the optimal solution of the problem

$$\max\left\{ \sum_{i=1}^{m} s(f_i(\mathbf{x})) : \mathbf{x} \in Q \right\} \tag{15}$$

is a fair solution for resource allocation problem (6).

In the case of the outcomes restricted to positive values, one may use logarithmic function thus resulting in the proportional fairness model (4). Various other concave functions s can be used to define fair aggregations (15) and the resulting resource allocation schemes. However, the problem of network dimensioning, we consider, is originally an LP model. Therefore, it is important if various fair allocation schemes can be generated with LP tools. We will show such LP models in the next section.

The standard maximin approach (13) may be lexicographically extended to the full MMF model where, in addition to the smallest outcome, one maximizes also the second smallest outcome (provided that the smallest one remains as large as possible), maximizes the third smallest (provided that the two smallest remain as large as possible), etc. Note that the lexicographic maximization is not applied to any specific order of the original criteria. Nevertheless, in the case of LP problems, there exists a dominating objective function which is constant on the entire optimal set of the maximin problem [9]. Hence, having solved the maximin problem, one may try to identify the dominating objective and eliminate it to formulate a restricted maximin problem on the former optimal set. Therefore, the lexicographic maximin solution to LP problems can be found by sequential maximin optimization with elimination of the dominating functions. Although, the LP models, we will present in the next section, provide us with a direct formulation for the MMF model.

4. Ordered outcomes

Multiple criteria optimization defines the dominance relation by the standard vector inequality. The theory of majorization [10] includes the results which allow us to express the relation of fair (equitable) dominance as a vector inequality on the cumulative ordered outcomes [7]. This can be mathematically formalized as follows. First, introduce the ordering map $\Theta : R^m \to R^m$ such that $\Theta(\mathbf{y}) = (\theta_1(\mathbf{y}), \theta_2(\mathbf{y}), \ldots, \theta_m(\mathbf{y}))$, where $\theta_1(\mathbf{y}) \le \theta_2(\mathbf{y}) \le \cdots \le \theta_m(\mathbf{y})$ and there exists a permutation τ of set I such that $\theta_i(\mathbf{y}) = y_{\tau(i)}$ for $i = 1, \ldots, m$. Next, apply to ordered outcomes $\Theta(\mathbf{y})$, a linear cumulative map thus resulting in the *cumulative ordering map* $\bar{\Theta}(\mathbf{y}) = (\bar{\theta}_1(\mathbf{y}), \bar{\theta}_2(\mathbf{y}), \ldots, \bar{\theta}_m(\mathbf{y}))$ defined as

$$\bar{\theta}_i(\mathbf{y}) = \sum_{j=1}^{i} \theta_j(\mathbf{y}) \quad \text{for } i = 1, \ldots, m. \tag{16}$$

The coefficients of vector $\bar{\Theta}(\mathbf{y})$ express, respectively: the smallest outcome, the total of the two smallest outcomes, the total of the three smallest outcomes, etc.

Vector $\bar{\Theta}(\mathbf{y})$ can be viewed graphically with a piece wise linear curve connecting point (0,0) and points $(i/m, \bar{\theta}_i(\mathbf{y})/m)$ for $i = 1, \ldots, m$. Such a curve represents the absolute Lorenz curve which can be mathematically formalized as follows. First, we introduce the right-continuous cumulative distribution function:

$$F_{\mathbf{y}}(d) = \sum_{i=1}^{m} \frac{1}{m} \delta_i(d), \quad \text{where} \quad \delta_i(d) = \begin{cases} 1 & \text{if } y_i \le d \\ 0 & \text{otherwise} \end{cases}$$

which for any real value d provides the measure of outcomes smaller or equal to d. Next, we introduce the quantile function $F_{\mathbf{y}}^{(-1)}$ as the left-continuous inverse of the cumulative distribution function $F_{\mathbf{y}}$:

$$F_{\mathbf{y}}^{(-1)}(\eta) = \inf\{d : F_{\mathbf{y}}(d) \ge \eta\} \quad \text{for } 0 < \eta \le 1.$$

By integrating $F_{\mathbf{y}}^{(-1)}$ one gets $F_{\mathbf{y}}^{(-2)}(0) = 0$ and

$$F_{\mathbf{y}}^{(-2)}(\eta) = \int_0^{\eta} F_{\mathbf{y}}^{(-1)}(\alpha) d\alpha \quad \text{for } 0 < \eta \le 1.$$

Graphs of functions $F_{\mathbf{y}}^{(-2)}(\eta)$ (with respect to η) take the form of concave curves (Fig. 1), the *(upper) absolute Lorenz curves*. In our case of m outcomes, the absolute Lorenz curve is completely defined by the values $F_{\mathbf{y}}^{(-2)}(i/m) = \frac{1}{m} \bar{\theta}_i(\mathbf{y})$ for $i = 1, \ldots, m$, where $F_{\mathbf{y}}^{(-2)}(1/m) = \bar{\theta}_1(\mathbf{y}) = \theta_1(\mathbf{y})$ represent the worst outcome and $F_{\mathbf{y}}^{(-2)}(1) = \frac{1}{m} \bar{\theta}_m(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} \theta_i(\mathbf{y})$.

In income economics the Lorenz curve is a cumulative population versus income curve [10]. A perfectly equal distribution of income has the diagonal line as the Lorenz curve and no outcome vector can be better. The absolute Lorenz curves, we consider, are unnormalized taking into account also values of outcomes. Vectors of equal outcomes are distinguished according to the value of outcomes. They are

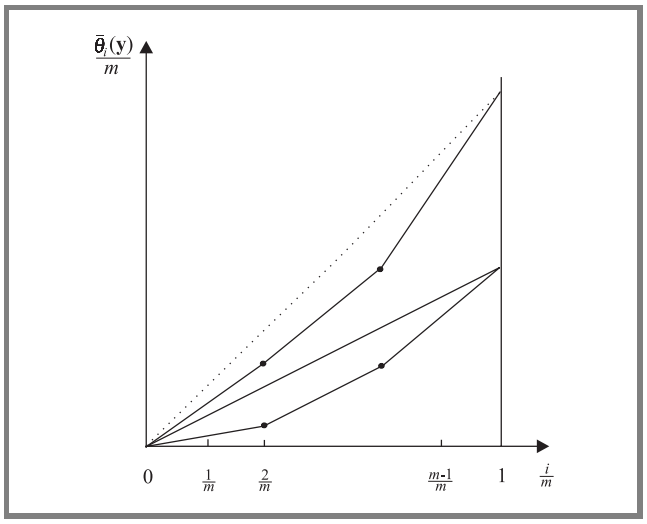


Fig. 1. Vectors $\bar{\Theta}(\mathbf{y})$ as the absolute Lorenz curves.

graphically represented with various ascent lines in Fig. 1. Hence, with the relation of fair dominance an outcome vector of large unequal outcomes may be preferred to an outcome vector with small equal outcomes.

Note that fair solutions to problem (6) can be expressed as Pareto-optimal solutions for the multiple criteria problem with objectives $\bar{\Theta}(\mathbf{f}(\mathbf{x}))$

$$\max\left\{\left(\bar{\theta}_1(\mathbf{f}(\mathbf{x})),\bar{\theta}_2(\mathbf{f}(\mathbf{x})),\ldots,\bar{\theta}_m(\mathbf{f}(\mathbf{x}))\right):\mathbf{x}\in Q\right\}. \quad (17)$$

Corollary 2. A feasible solution $\mathbf{x}\in Q$ is a fair solution of the resource allocation problem (6), iff it is a Pareto-optimal solution of the multiple criteria problem (17).

Corollary 2 provides the relationship between fair allocation schemes and Pareto-optimality. Moreover, the multiple criteria problem (17) may serve as a source of fair allocation schemes.

Although the definitions of quantities $\bar{\theta}_k(\mathbf{y})$, used as criteria in (17), are very complicated, the quantities themselves can be modeled with simple auxiliary variables and constraints. It is commonly known that the worst (largest) outcome may be defined by the following optimization: $\bar{\theta}_1(\mathbf{y}) = \max\{t : t \le y_i \text{ for } i=1,\ldots,m\}$, where t is an unrestricted variable. It turns out that this approach can be generalized to provide an effective modeling technique for quantities $\bar{\theta}_k(\mathbf{y})$ with arbitrary k [14]. Namely, for a given outcome vector $\mathbf{y}$ the quantity $\bar{\theta}_k(\mathbf{y})$ may be found by solving the following linear program:

$$\begin{aligned}\bar{\theta}_k(\mathbf{y}) \;=\; &\max\; kt-\textstyle\sum_{i=1}^m d_i\\ &\text{s.t. } t-y_i\le d_i,\; d_i\ge 0 \quad \text{for } i=1,\ldots,m,\end{aligned} \quad (18)$$

where t is an unrestricted variable while nonnegative variables d_i represent, for several outcome values y_i, their downside deviations from the value of t. Independently from the formal proof [14], this formula can be justified as follows. It is obvious that $\max(kt-\sum_{i=1}^m d_i)=\bar{\theta}_k(\mathbf{y})$ whenever no more than $k-1$ deviations d_i are strictly positive. On the other hand, for any t and d_i feasible to (18) one can define an alternative feasible values: $\tilde{t}=t-\Delta$ and $\tilde{d}_i=d_i-\Delta$ for $d_i>0$, where Δ is an arbitrary small positive number. For at least k positive values one gets $k\tilde{t}-\sum_{i=1}^m \tilde{d}_k \ge kt-\sum_{i=1}^m d_k$, which justifies (18).

Formula (18) provides us with a computational formulation for the worst conditional mean $M_{\frac{k}{m}}(\mathbf{y})$ defined as the mean outcome for the k worst-off services, i.e.:

$$M_{\frac{k}{m}}(\mathbf{y})=\frac{1}{k}\bar{\theta}_k(\mathbf{y}), \quad \text{for } k=1,\ldots,m. \quad (19)$$

Note that for $k=1$, $M_{\frac{1}{m}}(\mathbf{y})=\bar{\theta}_1(\mathbf{y})=\theta_1(\mathbf{y})=M(\mathbf{y})$ thus representing the minimum outcome, and for $k=m$, $M_{\frac{m}{m}}(\mathbf{y})=\frac{1}{m}\bar{\theta}_m(\mathbf{y})=\frac{1}{m}\sum_{i=1}^m\theta_i(\mathbf{y})=\frac{1}{m}\sum_{i=1}^m y_i=\mu(\mathbf{y})$ which is the mean outcome. Formula (18) allows us to maximize effectively the worst conditional means for various intermediate values k [13].

Note that Corollary 2 allows one to generate equitably efficient solutions of (6) as efficient solutions of problem (17). The aggregation maximizing the sum of outcomes, corresponds to maximization of the last (mth) objective in problem (17). Similar, the maximin scalarization corresponds to maximization of the first objective in (17). For modeling various fair preferences one may use some combinations of the cumulative ordered outcomes $\bar{\theta}_i(\mathbf{y})$. In particular, for the weighted sum on gets

$$\sum_{i=1}^m w_i\bar{\theta}_i(\mathbf{y}). \quad (20)$$

Note that, due to the definition of map $\bar{\Theta}$ with (16), the above function can be expressed in the form with weights $v_i=\sum_{j=i}^m w_j$ $(i=1,\ldots,m)$ allocated to coordinates of the ordered outcome vector. Such an approach to aggregation of outcomes was introduced by Yager [19] as the so-called ordered weighted averaging (OWA). When applying OWA to problem (6) we get

$$\max\left\{\sum_{i=1}^m v_i\theta_i(\mathbf{f}(\mathbf{x})):\mathbf{x}\in Q\right\}. \quad (21)$$

The OWA aggregation is obviously a piece wise linear function since it remains linear within every area of the fixed order of arguments.

If weights v_i are strictly decreasing and positive, i.e. $v_1>v_2>\cdots>v_{m-1}>v_m>0$, then each optimal solution of the OWA problem (21) is a fair solution of (6). Moreover, in the case of LP models, as the network dimensioning one, every fair allocation scheme can be identified as an optimal solution to some OWA problem with appropriate monotonic weights [7].

While equal weights define the linear aggregation, several decreasing sequences of weights lead to various strictly Schur-concave and strictly monotonic aggregation functions. Thus, the monotonic OWA aggregations provide a family of piece wise linear aggregations filling out the

space between the piece wise linear aggregation functions (12) and (13) as shown in Fig. 2. Actually, formulas (20) and (18) allow us to formulate any monotonic (not necessarily strictly) OWA problem (21) as the following LP extension of the original multiple criteria problem:

$$\max \sum_{k=1}^{m} w_k z_k \tag{22}$$

subject to $\mathbf{x} \in Q$

$$z_k = kt_k - \sum_{i=1}^{m} d_{ik} \quad \text{for } k = 1, \ldots, m \tag{23}$$

$$t_k - d_{ik} \leq f_i(\mathbf{x}), \quad d_{ik} \geq 0 \quad \text{for } i,k = 1, \ldots, m, \tag{24}$$

where $w_m = v_m$ and $w_k = v_k - v_{k+1}$ for $k = 1, \ldots, m-1$.

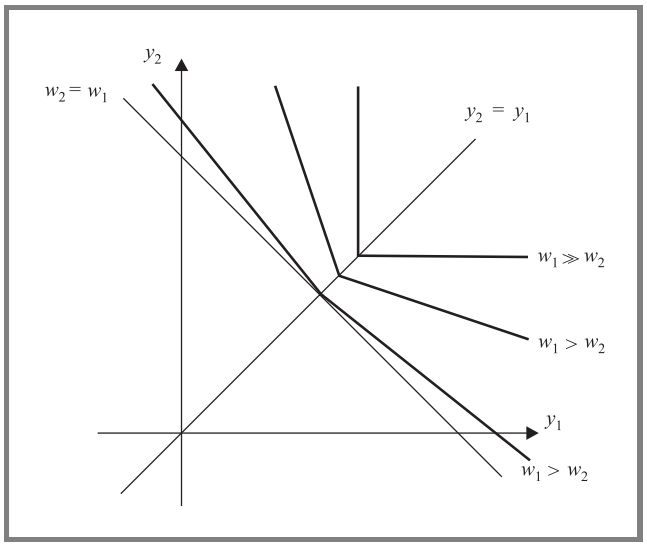


Fig. 2. Isoline contours for equitable OWA.

When differences among weights tend to infinity, the OWA aggregation approximates the lexicographic ranking of the ordered outcome vectors [20]. That means, as the limiting case of the OWA problem (21), we get the lexicographic problem:

$$\text{lexmax} \left\{ \Theta(\mathbf{f}(\mathbf{x}) : \mathbf{x} \in Q \right\} \tag{25}$$

which represents the MMF (lexicographic maximin) approach to the original resource allocation problem (6). Problem (25) is a regularization of the standard maximin optimization (13), but in the former, in addition to the worst outcome, we maximize also the second worst outcome (provided that the smallest one remains as large as possible), maximize the third worst (provided that the two smallest remain as large as possible), and so on. Due to (16), the MMF problem (25) is equivalent to the problem:

$$\text{lexmax} \left\{ \bar{\Theta}(\mathbf{f}(\mathbf{x}) : \mathbf{x} \in Q \right\}$$

which leads us to a standard lexicographic optimization with predefined linear criteria defined according to (18).

5. Computational results

First we have tested the OWA computational models (22)–(24) when applied to a generic LP resource allocation problem. We tested solution times for different size parameters. For each number of decision variables n and number of criteria (services) m we solved 20 randomly generated problems (Table 1). All computations were performed on a PC with the Pentium 200 MHz processor employing the CPLEX 6.0 package [4].

Table 1
Computation times for randomly generated problems

Services	Allocations – n					
m	5	10	20	40	60	100
10	0.05	0.10	0.10	0.15	0.15	0.20
20	0.30	0.35	0.40	0.60	0.75	1.00
30	0.80	1.00	1.55	2.15	2.65	3.35
40	1.95	2.35	3.20	5.25	6.75	9.50
60	7.30	8.80	10.95	20.75	31.30	44.95
100	49.05	54.60	65.40	104.15	173.10	278.80

Further we have analyzed sample network dimensioning problem with elastic traffic. For this purpose we have considered a network of the topology is patterned after the backbone network of a Polish ISP (Fig. 3). The network has 12 nodes, and we consider flows between any pair of these nodes (therefore, there are $144 - 12 = 132$ flows). All links have marginal costs equal to one, and the budget for link bandwidth is $B = 1000$. Since all links have equal costs of one, path cost will be equal to the link length, which is 1, 2, 3 or 4 in the example topology. All flows are unbounded. However, it is clear that due to the budget constraint no flow can exceed B.

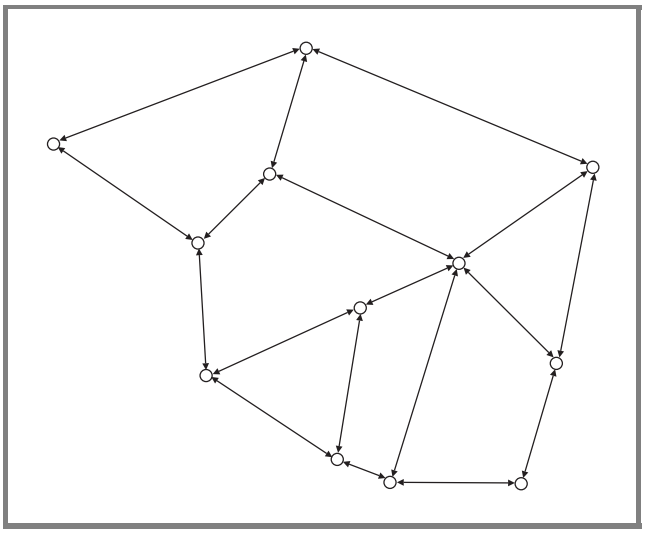


Fig. 3. Sample network topology.

To have control over the solution that will be found by the model, we decided to scale the outcomes (flows). Following the concepts of reference point methodology [17], we as-

sume that the decision maker (DM) specifies requirements in terms of aspiration and reservation levels, i.e., by introducing desired (acceptable) and required values for several outcomes. Depending on the specified aspiration and reservation levels, y_i^a and y_i^r, respectively, a special achievement function is built which can be interpreted as a measure of the DM's satisfaction with the current value of outcome the ith outcome. It is a strictly increasing function of outcome y_i with value 1 if $y_i = y_i^a$, and value 0 for $y_i = y_i^r$. Thus the partial achievement functions map the outcomes values onto a normalized scale of the DM's satisfaction. Various functions can be built meeting those requirements [18]. We use the piece wise linear function:

$$\sigma_i(y_i) = \begin{cases} \gamma(y_i - y_i^r)/(y_i^a - y_i^r), & \text{for } y_i \le y_i^r \\ (y_i - y_i^r)/(y_i^a - y_i^r), & \text{for } y_i^r < y_i < y_i^a \\ \beta(y_i - y_i^a)/(y_i^a - y_i^r) + 1, & \text{for } y_i \ge y_i^a \end{cases}$$

where β and γ are arbitrarily defined parameters satisfying $0 < \beta < 1 < \gamma$. Parameter β represents additional increase of the DM's satisfaction over level 1 when a criterion generates outcomes better than the corresponding aspiration level. On the other hand, parameter $\gamma > 1$ represents dissatisfaction connected with outcomes worse than the reservation level. The achievement function σ_i can be viewed as an extension of the fuzzy membership function to a strictly monotonic and concave utility (Fig. 4).

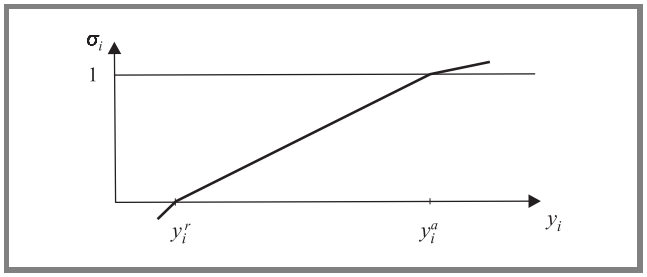


Fig. 4. Outcomes scaled with the achievement function.

The scaled flows are combined into an objective function using the OWA model. The linear program formulation of the OWA approach uses weights w_i, which are first-order differences of the weights v_i which are coefficients of the ordered outcome vector in the OWA model. In the approach used here, the weights $w_i = 1$ for all i. Thus, the OWA model has linearly decreasing weights. In the next section, we shall apply the outlined approach to search for compromise solutions of the network dimensioning problem with elastic flows using the sample topology given in Fig. 3.

The first application of the outlined approach used the same reservation and aspiration levels for all flows. Predictably, the result was a perfectly fair solution with each flow equal to 3.546, and a throughput of 468.1. This solution has a throughput which is less than 50% of the optimum throughput (equal to the budget constraint, 1000).

Next, the aspiration and reservation levels were chosen close to the values of the flows predicted by the property of the PF approach. Indeed, we got the optimal solution of the PF model, which has a throughput of 573.3. While the throughput of this solution is larger than in the perfectly fair solution, it is still not large when compared to the optimum.

Finally, the aspiration levels were set to 999 (close to the maximal flow), and the reservation levels were chosen for flows that had identical path costs in the following way: the flows with path cost equal to 1 had a reservation level of 15; flows with path cost equal to 2 had a reservation level of 2.0; flows with path cost equal to 3 had a reservation level of 1.0, and flows with path cost equal to 4 had a reservation level of 0.5. This approach resulted in a solution that had a throughput of 732.7, yet the smallest flow was larger than 1.0, and flows with equal path costs were treated fairly, like in the proportionally fair solution. The Lorenz curves of all the described solutions are shown in Fig. 5. Note that none of the solutions dominates any other.

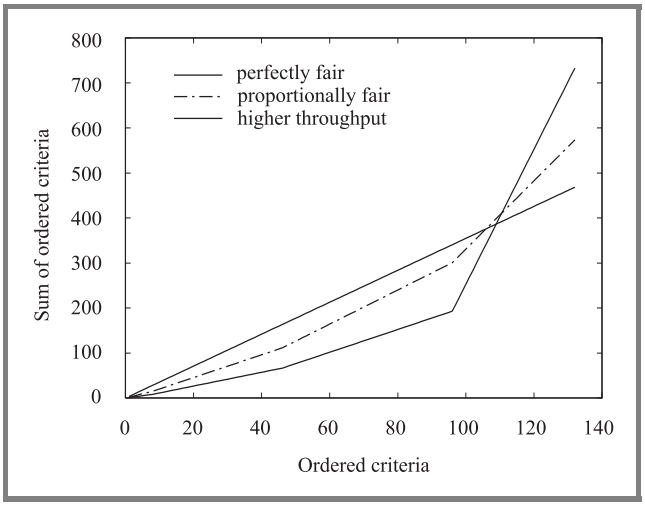


Fig. 5. Solutions obtained for the sample topology.

As was indicated in the introduction, the users of a network could be interested in fair treatment of flows between any pair of nodes, or in some other form of fairness. For example, the users could be interested in having fair amounts of available throughput from all other nodes to the user's node. This form of preferences could be expressed by the criteria:

$$n_v = \sum_{p_i=(u,v)} x_i \qquad \forall v \in V\,. \tag{26}$$

In this case, the number of criteria is reduced. Also, note that in approaches which make the value of a flow dependent on the distance between the origin and destination (like proportional fairness), nodes which are distant from all other nodes will be treated unfairly. The three solutions described above will be shown in Fig. 6. The figure plots the Lorenz curves for the 12 criteria n_v for each of the three

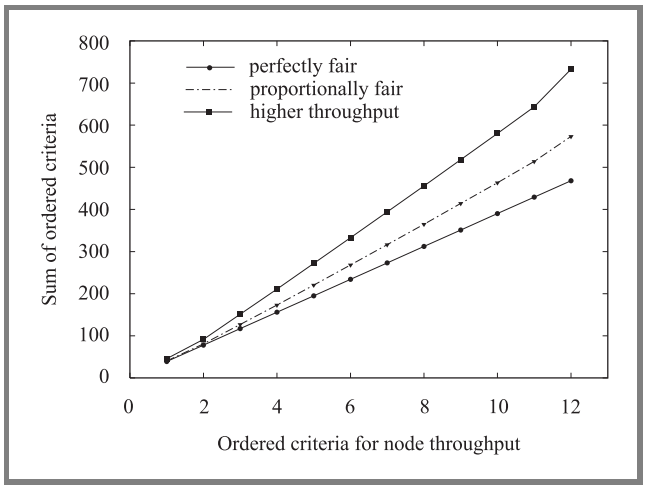


Fig. 6. The three solutions with respect to node criteria n_v.

solutions. It can be seen that the solution which increases throughput dominates the other two. This is a consequence of the design of the network topology, which is such that increasing network throughput improves the throughputs toward all the nodes. Another consequence of the topology is that all nodes have close values of criteria n_v, which is why the curves on the figure are almost straight; in more detail one could notice that the curves for proportional fairness and the OWA method have each 6 changes of slope. The perfectly fair solution predictably remains perfectly fair for the criteria n_v.

6. Concluding remarks

In various systems which serve many users, like in telecommunications systems, there is a need to respect the fairness rules, i.e. to allocate resources equitably among the competing services. Allocating the resources to optimize the worst performances may cause a large worsening of the overall (mean) performances. Therefore, several other fair allocation schemes are searched and analyzed.

The conditional mean is based on averaging restricted to the group of the worst performances defined by the tolerance level. Our earlier computational experiments with the conditional mean criterion applied to a traffic engineering model (a single ring bidirectional loading) were very promising [13]. The OWA aggregation further enriches modeling capacity offered by the conditional mean. In the case of LP models all equitable preferences may be modeled by selection of weights in the OWA aggregation.

Initial experiments with application of the OWA criterion (together with the reference point methodology) to the problem of network dimensioning with elastic traffic have confirmed the theoretical properties of the approach. We were able to generate easily allocations representing classical fairness models as well as to find new compromise solutions.

Maximization of the OWA aggregation, similar to the standard minimax approach, can be defined by optimization of a linear objective and a number of auxiliary linear inequalities. Many specific large-scale allocation models (especially discrete ones) may need some specialized exact or approximate algorithms. Thus, further research on computational aspects is necessary.

The problem of network dimensioning with elastic traffic could be extended with constraints on the individual flows. For example, network management could obtain traffic statistics that indicate the maximum throughputs which will be required between a pair of nodes. On the other hand, network statistics could also determine how much of the IP traffic requires guaranteed throughput (for example, from voice over IP applications). From this, minimal throughputs between a pair of nodes could be derived. In this work, we have analyzed in details the network design with elastic traffic without flow constraints. However, our approach allows to express such constraints in the objective function.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs: Prentice-Hall, 1987.

[2] J. F. Campbell, "Hub location and the *p*-hub median problem", *Oper. Res.*, vol. 44, pp. 923–935, 1996.

[3] E. Gourdin, "Optimizing Internet networks", *OR/MS Today*, vol. 28, pp. 46–49, 2001.

[4] ILOG Inc., Using the CPLEX Callable Library. Incline Village: ILOG Inc., CPLEX Division, 1997.

[5] F. Kelly, A. Mauloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability", *J. Oper. Res. Soc.*, vol. 49, pp. 206–217, 1997.

[6] J. G. Klincewicz, "Hub location in backbone/tributary network design: a review", *Loc. Sci.*, vol. 6, pp. 307–335, 1998.

[7] M. M. Kostreva and W. Ogryczak, "Linear optimization with multiple equitable criteria", *RAIRO Oper. Res.*, vol. 33, pp. 275–297, 1999.

[8] H. Luss, "On equitable resource allocation problems: a lexicographic minimax approach", *Oper. Res.*, vol. 47, pp. 361–378, 1999.

[9] E. Marchi and J. A. Oviedo, "Lexicographic optimality in the multiple objective linear programming: the nucleolar solution", *Eur. J. Oper. Res.*, vol. 57, pp. 355–359, 1992.

[10] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications.* New York: Academic Press, 1979.

[11] P. Nilsson and M. Pióro, "Solving dimensioning problems for proportionally fair networks carrying elastic traffic", Lund Institute of Technology at Lund University, 2002.

[12] W. Ogryczak, "Comments on properties of the minimax solutions in goal programming", *Eur. J. Oper. Res.*, vol. 132, pp. 17–21, 2001.

[13] W. Ogryczak and T. Śliwiński, "On equitable approaches to resource allocation problems: the conditional minimax solution", *J. Telecommun. Inform. Technol.*, no. 3, pp. 40–48, 2002.

[14] W. Ogryczak and A. Tamir, "Minimizing the sum of the *k* largest functions in linear time", *Inform. Proc. Lett.*, vol. 85, pp. 117–122, 2003.

[15] J. Rawls, *The Theory of Justice*. Cambridge: Harvard University Press, 1971.

[16] A. Tomaszewski, "A polynomial algorithm for solving a general max-min fairness problem" in *Proc. 2nd Pol.-Germ. Teletraff. Symp. PGTS*, 2002, pp. 253-258.

[17] A. P. Wierzbicki, "A mathematical basis for satisficing decision making", *Math. Modell.*, vol. 3, pp. 391–405, 1982.

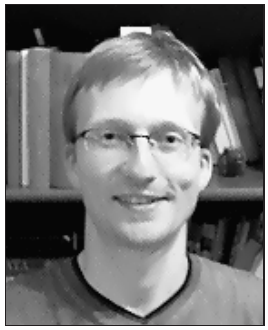
[18] A. P. Wierzbicki, M. Makowski, and J. Wessels, Eds., *Model Based Decision Support Methodology with Environmental Applications*. Dordrecht: Kluwer, 2000.

[19] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decision making", *IEEE Trans. Syst., Man Cybern.*, vol. 18, pp. 183–190, 1988.

[20] R. R. Yager, "On the analytic representation of the Leximin ordering and its application to flexible constraint propagation", *Eur. J. Oper. Res.*, vol. 102, pp. 176–192, 1997.

Włodzimierz Ogryczak is a Professor of Optimization and Decision Support in the Institute of Control and Computation Engineering (ICCE) at Warsaw University of Technology, Poland. He received both his M.Sc. (1973) and Ph.D. (1983) in mathematics from Warsaw University, and D.Sc. (1997) in computer science from Polish Academy of Sciences. His research interests are focused on theoretical research, computer solutions and interdisciplinary applications in the area of optimization and decision making with the main stress on: multiple criteria optimization and decision support, decision making under risk, location and distribution problems. He has published three books and numerous research articles in international journals. Since 2000 with ICCE, arlier with Warsaw University (Institute of Informatics).

e-mail: wogrycza@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland



Adam Wierzbicki received his B.S. in mathematics and M.S. in informatics from the University of Warsaw, in 1997 and 1998. In June 2003, he received a Ph.D. degree from the Institute of Telecommunications of Warsaw University of Technology. His Ph.D. thesis titled "Content distribution and streaming media communication on the Internet" concerned the design of content delivery networks, particularly for improving quality and performance of streaming media communication. His current research interests include Peer-to-Peer computing, application layer multicast and network design. He is employed as a research assistant at the Polish-Japanese Institute of Information Technology in Warsaw, Poland.

e-mail: adamw@icm.edu.pl
Polish-Japanese Institute of Information Technology
Koszykowa st 86
02-008 Warsaw, Poland

Tomasz Śliwiński
e-mail: tsliwins@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Paper

Some variants of projection methods for large nonlinear optimization problems

Paweł Białoń

Abstract — Two ideas of modifying projection methods for the case of smooth nonlinear optimization are presented. Projection methods were originally successfully used in solving large-scale linear feasibility problems. The proposed instantiations of projection methods fall into two groups. One of them is a decomposition approach in which projections onto sets are realized as optimization problems which themselves involve much portions of original problem constraints. There are two subproblems: one build with linear constraints of the original problem and the second one build with original nonlinear constraints. These approaches use special accelerating cuts so that the separation of nonlinear and linear constraints can be effective and some problem sparsity preserved. The second group bases on penalty-shifting/multiplier methods and draws from the observation that unconstrained subproblems obtained there may solve very slowly due to their nonsmooth character. Thus it is proposed to solve them with modified projection methods which inherit from conjugate gradient methods a multi-dimensional subspace in one epoche.

Keywords — projection methods, penalty shifting method, nonlinear optimization.

1. Introduction

Projection methods [2, 10] in their classical form serve for solving convex feasibility problems, i.e, the problems of finding a common point of several closed convex sets. They received some success, especially in image reconstruction for medical applications (see references in [2]), where problems of million sizes have been solved favourably for some time. There has been a considerable stream of research on adopting projection methods for optimization problems, which differ from feasibility problems only by the existence of a goal function. However, usually the investigations surround themselves with nondifferentiable optimization [13, 19], whereas the author sees some possibilities to aply them in more "standard" branches of optimization, i.e. nonlinear smooth, possibly large-scale optimization. Despite their success in solving large feasibility problems, projection methods have other features that seem appealing for such a usage. They do not have to involve any complex linear algebra in the case where the sets in feasibility problems are halfspaces (which means that the feasibility problem is linear). On the other hand, the large nonlinear optimization problems are usually composed of a big linear part and much smaller nonlinear part. Other attractive features of projection methods include a clear indication of how to accelerate them, easily seen from the convergence analysis: the method should make long steps. Several things seem discouraging, for example, the theoretical worst-case convergence for projection methods is not much competitive, but the author proposes some ways of taking advantage of information taken from quadratic models of the nonlinear optimization problem, which should accelerate the solving process.

In Section 2 a brief introduction to projection methods is given. The next two sections show a few-years work of the author on adapting projection methods for nonlinear optimization. Two approaches are presented. One of them is only briefly summarized in Section 3 (it was in more detail presented in [4] and [5]). This is a decomposition approach in which projections are realized via solving two different optimization subproblems with auxiliary solvers. One of the subproblems involves the linear constraints from the initial problem and the second one – the nonlinear ones. Due to the separation of constraints, specialized solvers (pure linear and nonlinear) may be used for the subproblems. The method is designed for problems in which the large size is generated mainly by the size of the linear part. The main effort in designing this method was done to generate special accelerating cuts that preserve the good features of the parts of the initial problem (sparse structure of the linear part and the low dimensionality of the nonlinear part). A very interesting feature of this method is that one of its best behaviors can be expected on the so-called nonlinear multicommodity flow problem, a classical item in telecommunication network design.

Section 4 presents an approach introducing projection methods in solving nonlinear optimization problems via the multiplier/penalty shifting method. The multiplier method produces unconstrained subproblems. Due to the spline character of these subproblems they might be sometimes very hard to solve, and the level of this hardness surprized the author who was trying to tackle them with a conjugate gradient method. The author proposes replacing the conjugate gradient method with a special kind of projection method, in which, however, a quadratic model of the minimized function and elements of the conjugate gradient method are still present and allow to obtain long steps in the projection method. This approach is formulated as a core algorithm and may have various realizations, each of them probably requiring a considerable amount of additional conceptual work.

Finally, in Section 5 some discussion and the author's observations regarding the methods as well as conclusions steming from the author's experience are given.

2. The idea of projection methods

Projection methods serve to solving the following *convex feasibility problem:*

$$\text{Find} \quad x \in S \stackrel{\text{def}}{=} \bigcap_{i=1,\ldots m} G_i\,, \tag{1}$$

where $G_i \subset \mathbb{R}^n$ are closed, convex sets. In practice, G_i are often defined as sets of points allowed by some constraints. Assume that S is nonempty. We start our description with the case of $m = 2$.

For $x \in \mathbb{R}^n$ and a closed convex nonempty $C \subset \mathbb{R}^n$ we shall denote by $P_C x$ the (unique) orthogonal projection of x onto C, $P_C x = \arg\min_{y \in C} \|x - y\|^2$. The *projection vector* for such a projection is $P_C x - x$.

The idea of searching for the solution consists in performing sequential alternate projections onto G_1 and G_2; i.e., given the starting point x^0, we produce a sequence

$$x^1 = P_{G_1} x^0, x^2 = P_{G_2} x^1, x^3 = P_{G_1} x^2, \text{ etc.} \tag{2}$$

We assume such projections are easily realizable numerically.

In the convergence analysis of projection methods it is important that the projection operator possesses the Fejér contraction property.

Definition. A finite or infinite sequence (x^i) of points in a Hilbert space H has the Fejér contraction property with respect to $C \subset H$ if

$$\|x^i - c\|^2 \geq \|x^{i+1} - c\|^2 + \|x^{i+1} - x^i\|^2 \tag{3}$$

for each $c \in C$. Similarly, operator $O : H \to H$ has this property if for each $c \in C$ and $x \in H$ $\|x - c\|^2 \geq \|Ox - c\|^2 + \|Ox - x\|^2$.

Fact. Projecting onto a nonempty closed convex set of points in $\mathbb{R}^n$ has Fejér c. p. with respect to this set.

For a proof of the above fact see calculations on page 228 in [14] with $t_{\min} = t_{\max} = 1$.

After putting $C = S$ we see that with every projection performed in our algorithm (2) we decrease the squared norm from (any but fixed) point $c \in S$ by at least the square of the appropriate step (projection vector) length . It now suffices to assure certain lengths of steps to establish the convergence[1].

In other words, the Fejér contraction property of projections in our algorithm means that we approach each solution point with an acute angle.

[1]Which is usually easy and is done with the notion of the problem geometrical property called *regularity* – see [2].

Zigzagging often slows down projection methods: we may approach the solution with an angle less than but close to $\pi/2$, making the distance from a solution decrease very slowly. This happens in an example in Fig. 1; there, moreover, consecutive projection vectors form angles close to π.

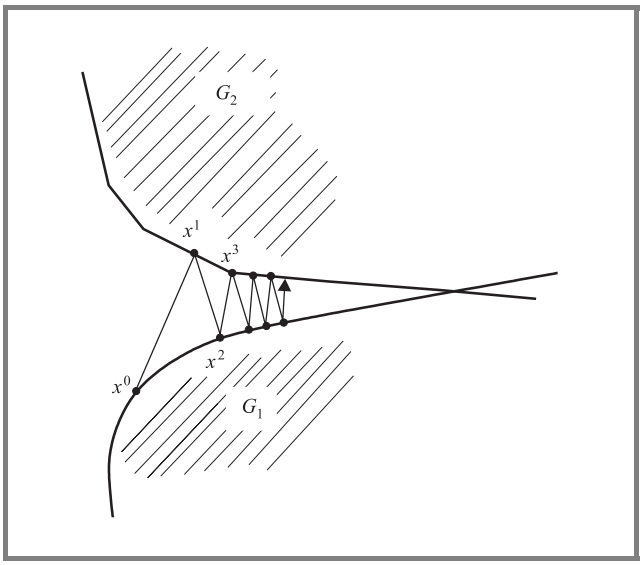


Fig. 1. Zigzagging.

Cuts serve as a standard remedy for zigzagging. Given point a and vector p in a Hilbert space, we define a cut as an inequality $\langle \cdot - a, b\rangle \geq \langle b, b\rangle \geq 0$ with fixed $a, b \in \mathbb{R}^n$; *its hyperplane* $H(a,b)$ is given as $\{x \in \mathbb{R}^n : \langle x - a, b\rangle = \langle b, b\rangle\}$, *its halfspace* – as $\{x \in \mathbb{R}^n : \langle x - a, b\rangle \geq \langle b, b\rangle\}$.

Using cuts means replacing (2) with

$$x^1 = P_{G'_1{}^1} x^0, x^2 = P_{G'_2{}^2} x^1, x^3 = P_{G'_1{}^3} x^2, \text{ etc.} \tag{4}$$

where sets G'^k_1 and G'^k_2 ($k = 1, 2, 3, \ldots$) are G_1 and G_2 narrowed by some cuts, i.e, they were obtained from G_1 and G_2 by intersecting G_1 and G_2 with halfspaces of some cuts.

A *geometric cut based on (constructed after) the projection* of $x \notin G$ onto close convex G, $G \supset S$ is defined as

$$\langle \cdot - x, P_G x - x\rangle \geq \langle P_G x - x, P_G x - x\rangle\,.$$

In Fig. 2, unlike in Fig. 1, point x^3 was obtained by projecting x^2 not onto G_2 but onto G_2 narrowed by the geometric cut constructed after projection of x^1 onto G_1. H is a hyperplane of this cut. We see that the step made is longer and we approach the solution with a smaller angle.

A cut is called *valid* or *proper* if it is satisfied for each point in the solution set S. Validity is necessary to assure that narrowed sets (i.e., G'^k_1 or G'^k_2) contain S and thus projection onto them still possesses the Fejér contraction property with respect to S; moreover, we do not want our method to degenerate by producing empty G'^k_1 or G'^k_2. Geometric cuts constructed after a projection of an $x \notin G$ onto nonempty, closed, convex $G \supset S$ can be easily shown to be proper.

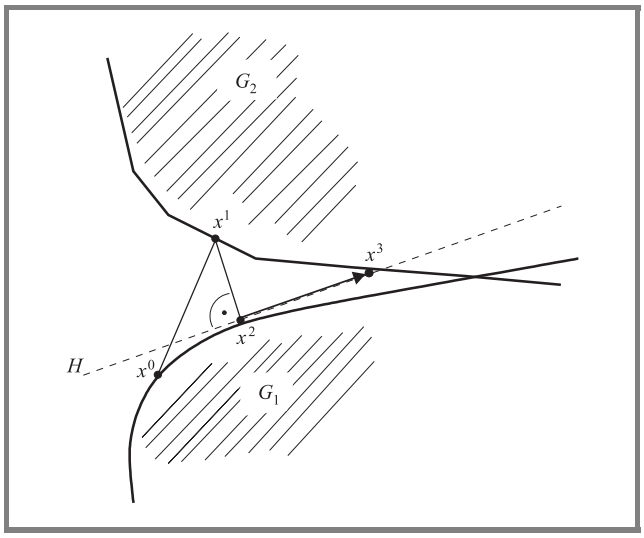


Fig. 2. A geometric cut reduces zigzagging.

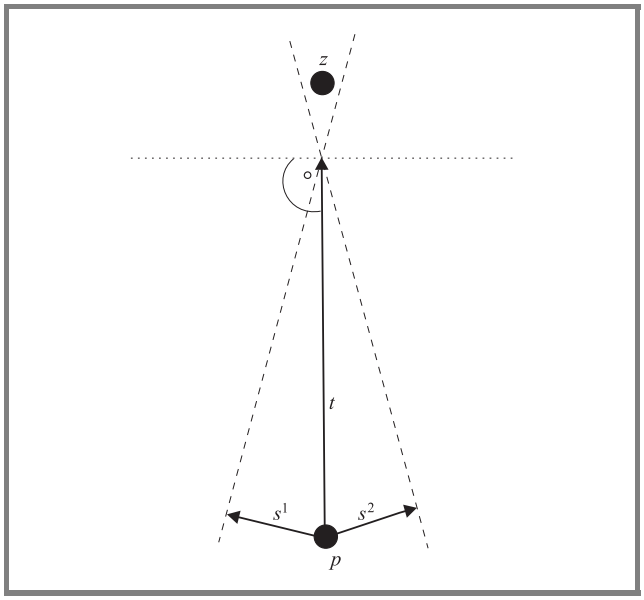


Fig. 3. Obtaining a surrogate cut (dotted line) from two cuts (dashed lines) whose normal vectors form an obtuse cone; $q = 2$.

We may narrow set G_1 or G_2 with only one cut but it may bring a profit in efficiency to narrow them with several cuts simultanously (i.e., to intersect G_1 or G_2 with the intersection of the halfspaces of several cuts). Various techniques for cuts cumulation are given in [6, 7, 12, 13, 16]. We can:

1. Project on the real intersection of halfspaces of cumulated cuts by solving an optimization problem that reflects the definition of orthogonal projection (it has a quadratic goal function).

2. Construct a valid "surrogate" cut on basis of the cumulated cuts and project on its halfspace, hopefully obtaining a long projection vector (step).

A way of obtaining a surrogate cut from q geometric cuts, say $\langle . - p, t_i \rangle \geq \langle t_t, t_i \rangle$, $i = 1, \ldots, q$ was given by Cegielski [6, 7] (and a similar approach – by Kiwiel [12]). Here p is the current iterate point.

Lemma (adopted Remark 7 on Theorem 3 in [6]). Let $p, z \in \mathbb{R}^n$, $p \neq z$. If

(a) $\mathcal{S} = \{s^i : i = 1, \ldots, q\}$ is a system of linearly independent vectors,

(b) $\forall_{i \in \{1,\ldots,q\}} \langle z - (p + s^i), s^i \rangle \geq 0$,

(c) coneconv $\mathcal{S}$ obtuse[2],

(d) t is the solution of the following equation system

$$\forall_{i \in \{1,\ldots,q\}} \langle (s^i), t - s^i \rangle = 0 \tag{5}$$

then $\langle z - (p + t), t \rangle \geq 0$.

This lemma says that the cut $\langle . - (p + t), t \rangle \geq 0$ is valid on condition that the cumulated cuts are valid[3] – see Fig. 3; this is the surrogate cut in Cegielski's method. The next iterate is obtained from p by adding t, a long vector, to it.

However, the normals s^i of cuts must form an obtuse cone. Cegielski assures it in several ways, the easiest one is to assure $\langle s^i, s^j \rangle \geq 0$ for $i \neq j$ (so s^i form a so-called regular obtuse cone), by a simple rejection of some cuts to be cumulated.

Any convex optimization problem of the form $\min_{x \in \bigcap_{j=1}^m G^j \subset \mathbb{R}^n} f(x)$ may be solved by means of projection methods for feasibility problems after reducing it to a feasibility problem of finding a common point of G^j, $j = 1, \ldots, m$ and $\{x \in \mathbb{R}^n : f(x) < Q\}$, parametrized with number Q, which must be experimentally tuned to the optimal value of the initial optimization problem within some schemme, e.g. bisection or level control [11]. Thus we actually need to solve a sequence of feasibility problems with various Qs; usually a detection of infeasibility of a feasibility problem must happen from time to time.

3. Decomposition of large problems into linear and nonlinear parts

The algorithm described in this section is multi-layered, similarly as the group of approaches described in the next section. They both combine some elements typical for smooth optimization with elements of projection methods. Here the projection layer is higher than the layer of standard smooth techniques.

[2]An *accute* cone is a cone C such that for $a \in C$, $b \in C$ $\langle a, b \rangle \leq 0$, an *obtuse* cone is a cone conjugate to some accute cone.

[3]i.e., they do not cut off any solution point represented here by z.

The solved problem after reducing to a feasibility problem with parameter Q has the following form: find $(x^\top, y^\top)^\top$ that satisfies:

$$\begin{aligned} & g(x) \le Q \\ & A(x^\top, y^\top)^\top \le b \\ & B(x^\top, y^\top)^\top = d \\ & x^{lo} \le x \le x^{up}, y^{lo} \le y \le y^{up}, \end{aligned} \tag{6}$$

where function $g : \mathbb{R}^{n_N} \to \mathbb{R}^{m_N}$ depends in one coordinate on Q, since this coordinate was made from the goal function of the optimization problem (the coordinate says how much the goal function value exceeds Q). A and B are matrices of appropriate sizes. Functions g_i are continuous quasiconvex, $x^{lo}, x^{up}, y^{lo}, y^{up}$ are constant vector bounds.

The feasibility problem has n_N nonlinear variables[4] , n_L linear variables, m_N nonlinear inequality constraints, m_{LI} linear inequality constraints, m_{LE} linear equality constraints. Let $m = m_N + m_{LI} + m_{LE}$, $n = n_L + n_N$. The better $m_N \ll m$ and $n_N \ll n$, are fulfilled, the more efficient the algorithm will be.

In order to solve the problem (6) we need to see it in the form of (1).

The following sets N and L will play the role of G_1 and G_2 in problem (1):

$$\begin{aligned} N = & \{x \in \mathbb{R}^{n_N} : g(x) \le 0 \wedge x^{lo} \le x \le x^{up}\} \\ L = & \Big\{ x \in \mathbb{R}^{n_N} : x^{lo} \le x \le x^{up} \wedge \\ & \wedge \exists_{y \in \mathbb{R}^{n_L}} \Big(y^{lo} \le y \le y^{up} \wedge \\ & \wedge A(x^\top, y^\top)^\top \le b \wedge B(x^\top, y^\top)^\top = d \Big) \Big\}. \end{aligned}$$

Notice that these are not actually the sets of points allowed by nonlinear and linear constraints but their orthogonal projections on the subspace of nonlinear variables. The projection algorithm operates in this low-dimensional subspace.

Projections on these sets are realized by solving optimization subproblems with quadratic goal functions, moreover:

1. A projection on N yields a small $(n_N \times m_N)$ subproblem with nonlinear constraints.

2. A projection on L yields a large $(n \times m_L)$ subproblem but with linear constraints.

The optimization subproblems are solved efficiently with specialized solvers, in the author's experiments nonlinear IAC-DIDASN++ [15] and quadratic HOPDM [1].

A special care is connected with using geometric cuts, but this will be only outlined here (see [4] and [5] for details). An introduction of geometric cuts in general means extending the above subproblems by adding suitable linear inequality constraints to them. Whenever we use such a cut, the inequality must be present in the subproblem realizing the projection.

We can freely construct a geometric cut based on a projection onto N: such a cut introduces at most n_L nonzeros into constraint matrices of the quadratic subproblem, which is not much by comparison with the nonzeros number in constraints from the quadratic optimization subproblem.

Using a cut based on a projection onto L should be avoided: if we wanted to use such a cut in some next projection onto N, we would introduce a bigger relative complication into the nonlinear subproblem. Thus we treat this cut only as "virtual", e.g., we state that such a cut might be constructed and would be proper, but we do not really add it to any collection of cumulated cuts that augments N. Then we cumulate such a "virtual" cut with the latest cut based on the projection onto N, obtaining a surrogate cut according to Lemma in point 2 in Section 2. The surrogate cut, called anti-zigzagging cut (or Z-cut) is used later instead of the "virtual" cut to augment subproblems; however, it augments only the quadratic subproblems so the subproblem complication is not excessive.

The successive Z-cuts can be then cumulated directly, by cumulation of constraints augmenting a subproblem (i.e., in the way described in point 1 in Section 2). Since the cumulation is full, i.e, each successive Z-cut is cumulated, the zigzagging in the method is claimed to decrease in the cited works.

A nice feature of the method is discussed in [5]. The required proportions of sizes are particularly good when we apply the method to the nonlinear multicommodity flow problem [18]. Moreover, the situation becomes better as the number of commodities distinguished in the problem grows.

4. Accelerating the multiplier method with projection methods elements

In this section we shall want to solve the following problem:

$$\min_{x \in \mathbb{R}^n} \varphi(x) \tag{7}$$

$$\text{s. t.}$$

$$g(x) \le 0 \tag{8}$$

with $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, g^i and φ continuous quasiconvex.

The idea of algorithm presented in this section bases on an observed poor behavior of a penalty shifting/multiplier method in the version for inequality constraints ([3], see also [21]) and with the Fletcher-Reeves conjugate gradient method applied to the resulting unconstrained subproblems. The author once solved a problem of the form (7)–(8) with several tens (!) of variables, a quadratic goal function and several tens or about a hundred of quadratic inequality constraints. Even on such a small example, the

[4] A *nonlinear variable* is a problem variable involved in at least one nonlinear function in the model formulation; the remaining variables will be called *linear*.

solution times sometimes reached a rank of hours. The number of iterations of the multiplier method was small, but the resulting subproblems were solved extremely slow. Probably setting a too high penalty coefficient was not the reason for this behavior, since the coefficient did not exceed the value of 1 even by several ranks of value; neither did the coefficients in the initial problem definition differ from 1 by many ranks. The probable reason for such a behavior of the method was formulated as a nonsmooth (actually spline) character of the unconstrained subproblems, caused by a similar character of the augmented Lagrangian function.

We can find several signs in the literature that seem to support the anxiety about the augmented Lagrangian for inequality constraints and the application of it being spoken. One of them is the existing collection of trials of modifying the augmented Lagrangian in order to eliminate its quadratic-spline character. An example is a construction of a cubic Lagrangian. A second one might be the way of treating nonlinear inequality constraints in the broadly recognized LANCELOT solver [8]. Instead of using the version of the multiplier method for inequality constraints directly, the nonlinear inequality constraints are first transformed to nonlinear equality constraints by an addition of bounded slack variables, similarily to the way it is done in the simplex method. Then the equality constraints are treated with the variant of the multiplier method for equality constraints, in which the Lagrangian is smooth. The constructors of the solver even agree with a possibility of obtaining nonconvex subproblems, with additional slacks and bounds (the bounds on slacks are transfered to the unconstrained subproblems of the multiplier method).

Having in mind the hardness of the subproblems with nonsmooth Lagrangian, the author of this paper decided to solve the subproblems with a variant of projection method.

If we wanted to truly treat the Lagrangian (the goal function of the unconstrained subproblem) as a nondifferentiable function, we would perhaps want to use some variant of the Polyak subgradient method [19] for minimizing a convex goal function ϕ. In an iteration of this method, one makes a projection of the current iterate point on the Q-level set of the linear underestimation of ϕ constructed on the basis of the value and the subgradient of ϕ at the current iterate; Q denotes, as previously, the current estimation of the minimal value of ϕ. But it seems better for the convergence speed if we take an advantage of a quadratic model of the minimized Lagrangian (let us denote it as $f : \mathbb{R}^n \mapsto \mathbb{R}$) which may be, at least locally, good if the initial optimization problem was smooth.

In a Polyak method iterate we actually use the information at one point and we obtain a projection vector of a certain length. In the author's proposition we first make $k \leq n$ steps (an epoche of steps) of the Fletcher-Reeves method, say, while the quadratic approximation of f seems good enough. From there we have an approximate model of the function in a whole subspace Δ of the dimensionality of k. This information allows us to find a point $\tilde{y} \in \Delta$ at which we can construct a valid cut: $\langle . - \tilde{y}, \nabla f(\tilde{y}) \rangle \leq 0$. Then we project the current iterate p onto the halfspace of this cut; since the cut is valid, the projection possesses the Fejér contraction property w.r.t. any solution point. The vector of projection of current iterate p onto the halfspace of this cut is hoped to be much longer than that in the Polyak method, since we can choose $\tilde{y}$ from the whole subspace Δ. As we remember, a big step length usually implies a quicker convergence in projection methods.

We shall manage only to outline the proposition, since it is quite sophisticated and may have many variants. We start with the heart of the proposition, which is calculating $\tilde{y}$.

Assume the epoche of conjugate gradient method generated points $x^0 \equiv p$, $x^1, \ldots, x^k \in \mathbb{R}^n$, conjugate directions: $d^0, d^1, \ldots, d^{k-1} \in \mathbb{R}^n$, gradients of f: $g^0 = \nabla f(x^0)$, $g^1 = \nabla f(x^1), \ldots, g^k = \nabla f(x^k)$, real coefficients $\beta^1, \beta^2, \ldots, \beta^k$ and the step lengths $\alpha^0, \alpha^1, \ldots, \alpha^{k-1}$.

These objects are interrelated with the following dependencies:

$$d^0 = -g^0, \tag{9}$$

$$d^i = -g^i + \beta^i d^{i-1} \quad (i = 1, \ldots, k), \tag{10}$$

$$x^{i+1} = x^i + \alpha^i d^i \quad (i = 0, \ldots, k-1). \tag{11}$$

For simplicity we assume $x^k = 0$.

We shall now treat f as a quadratic function defined with a symmetric, positive definite matrix.

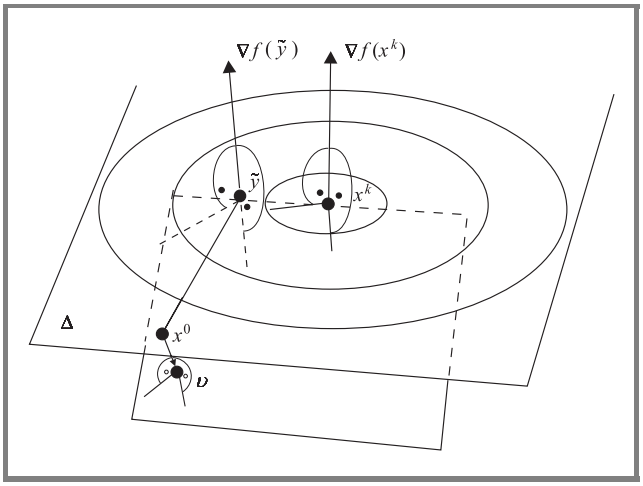


Fig. 4. Choosing point $\tilde{y}$. The ellipses denote the level sets of f, the smaller plane is the plane of the cut and v is the projection vector that should be as long as possible.

The problem of choosing $\tilde{y}$ is shown in Fig. 4. Vector v is the projection vector and we want to make it as long as possible. In order to make a small exercise try to imagine how this figure change for two choices: $\tilde{y} = x^0$ and $\tilde{y} = x^k$. Observe that x^0 is a poor candidate for $\tilde{y}$, since for such a choice the final steplen $|v|$ would be equal to 0. Neither $\tilde{y} = x^k$ is a good choice, since from theory of conjugate gradient methods we know that $g^k \perp \Delta$, which against yields $||v|| = 0$.

The heuristic taken by the author is to search for $\tilde{y}$ among points x satisfying[5]

$$\nabla_{\|} f(x) = -\tau(x - x^0), \tag{12}$$

$$x \in \Delta \tag{13}$$

for some positive parameter τ, where $\nabla_{\|} f$ denotes the part of ∇f parallel to Δ.

The solution of system (12)–(13) becomes easier when we represent x in the basis of conjugate directions: $x = Ds$, $D \equiv \left[d^0 d^1 \ldots, d^{k-1}\right]$ and thus reduce them to a search for the best s.

Under such a representation, system (12)–(13) transforms, with some elaborated calculations, to

$$(I + \tau M)s = q \tag{14}$$

with

$$M = \begin{bmatrix} \frac{1}{\alpha^0} & -\frac{\beta^1}{\alpha^1} & & & \\ \frac{\beta^1 - 1}{\alpha^0} & \frac{\beta^2 + 1}{\alpha^1} & & & \\ & \frac{-1}{\alpha^1} & \ddots & \frac{-\beta^{k-2}}{\alpha^{k-2}} & \\ & & & \frac{\beta^{k-1} + 1}{\alpha^{k-2}} & \frac{-\beta^{k-1}}{\alpha^{k-1}} \\ & & & \frac{-1}{\alpha^{k-2}} & \frac{1}{\alpha^{k-1}} \end{bmatrix} \tag{15}$$

and q being the representation of x^0 in the basis of conjugate directions: $x^0 = Dq$.

Now the search for the best $\tilde{y}$ is reduced to a search for optimal $\tau > 0$. For a candidate τ we find s by solving a quite easy system (14), with a tridiagonal matrix, compute $\tilde{y} = Ds$ and for it (based on real gradient of the minimized function f, or on its quadratic model being considered now), the projection vector. The length of the projection vector seems not to be in general a concave function of τ, but practical experiments showed that one-dimensional optimization in τ may be replaced just with examining several values of τ.

The cuts generated by the algorithm are then cumulated with the Cegielski's method of regular obtuse cone presented in Section 2, but only a limited number of them takes part in the cumulation process in order to keep the linear systems present in the method (one in the cuts cumulation process, second in searching for $\tilde{y}$) small.

5. Discussion

The method of decomposition from Section 3 has been thoroughly discussed in [4] and [5]. Several modifications of this method are possible, e.g., a possibility of augmenting set N with geometric cuts instead of set L if we decide so by more precise analysis of particular problem sizes. Other options include subtle changes in the order of cuts cummulation, which may affect the speed of convergence. The method performed quite good on an artificial multicommodity flow problem in the sense of number of iterations in the projection method layer. Thus, the decomposition of the problem into linear and nonlinear parts seems to be done well, but the overal effectiveness of the method depends on the speed of the solvers solving pure (quadratic or nonlinear) subproblems. Applying warm restarts during many runs of the quadratic solver seems to be necessary in order to make this method competitive with commercial solvers on this problem.

Regarding the method of combining projection and the conjugate gradient method from Section 4, one must be aware of a great number of technical details and further decisions that we face when trying to implement it, in particular:

1. A separate treatment of equality constraints in large problems. We constructed our method for purely nonlinear problems. We can formally represent linear constraints as nonlinear, but for large problem it usually becomes essential to treat them separately[6]. Linear constraints can be introduced directly in the form of additional cuts to be cumulated in the conical method.

2. Introducing bounds on variables would somewhat complicate the algorithm; perhaps some elements of projected gradient method would have to be used, may be the bounds would have to be added to the collection of cuts being cumulated.

Speaking about large problems, let us make an important note. Due to inserting linear constraints directly in the projection method steps and due using some conjugate gradient techniques (not variable metric) there is a chance to design the whole algorithm so that any complicated linear algebra, like an implicit inverting sparse matrices, is avoided.

Making the cuts based on points $\tilde{y}$ even deeper than in the above descriptions seems to be another important issue. In some circumstances such operations might be essential for an efficient work of the method.

1. Having an approximation Q for the optimal value of the unconstrained subproblem, one can shift any constructed cut (make it deeper) by using techniques known from the Polyak method (apply a linear model of f constructed at point $\tilde{y}$). However, f must be convex, not only quasiconvex, to make this approach proper.

[5]The demand of exact maximization of $\|v\|$ yields a multidimensional nonconvex optimization problem. Instead, in the heuristics we want to be far from the situation where angle between $(\tilde{y} - x^0)$ and $\nabla_{\|} f(\tilde{y})$ equals to $\pi/2$, as it happened with the choice of $\tilde{y} = x^k$.

[6]Remember also that the method does not support nonlinear equality constraints, so linear equality constraints cannot be represented as nonlinear. A propos, the inability of treating nonlinear equality constraints is common to both the propositions in this paper. It seems essential since it stems from the fact that projection methods work only for convex problems. Possibly, one might try some trust region approach to incorporate nonlinear equality constraints.

2. Do not use the Lagrangian in all runs of the subproblem solving, but instead its different valid underestimations, which are easy enough to obtain due to the spline character of the Lagrangian. This may introduce some perturbations in the cuts positions (usually deepening) resulting in long steps after the cuts cumulation. A difficulty occurs in such a case: one must stop conjugate gradients run when we go below the optimal value of the subproblem, since otherwise obtained cuts would be invalid.

Many technical decisions, some of them mentioned earlier, certainly must be made in order to make the process of finding τ work properly.
Another concern must be connected with the work of the method of cuts cummulation itself. It might be augmented or tuned for some interesting patterns configuration of cuts, frequently observed during experiments (closeness of the angles between the majority of cut normal pairs to $\pi/2$; very obtuse cones observed for some problems).

References

[1] A. Altman and J. Gondzio, "Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization", *Optim. Meth. Softw.*, no. 11–12, pp. 275–302, 2000.

[2] H. Bauschke and J. Borwein, "On projection algorithms for solving convex feasibility problems", *SIAM Rev.*, vol. 38, no. 3, pp. 367–426, 1996.

[3] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. A Series of Monographs and Textbooks: *Computer Science and Applied Mathematics*. New York, London: Academic Press, 1982.

[4] P. Białoń, "Large-scale nonlinear projection algorithm using projection methods", *Discuss. Math. Differ. Inclus., Control Optim.*, vol. 20, no. 2, pp. 171–194, 1999.

[5] P. Białoń, "A proposition to exploit the partially linear structure of the nonlinear multicommodity flow optimization problem", *J. Telecommun. Inform. Technol.*, no. 3, pp. 49–56, 2002.

[6] A. Cegielski, "A method of projection onto an acute cone with level control in convex minimization", *Math. Progr.*, vol. 85, pp. 469–490, 1999.

[7] A. Cegielski, *Metody relaksacyjne w problemach optymalizacji wypukłej*. Monografie, nr 67. Zielona Góra: Wyższa Szkoła Inżynierska, 1993 (in Polish).

[8] A. Conn, N. Gould, and P. Toint, "Numerical experience with LANCELOT package (Release A) for large-scale nonlinear optimization", *Math. Progr.*, vol. 73, pp. 73–110, 1996.

[9] R. Dylewski, "Numerical behavior of the method of projection onto an acute cone with level control in convex optimization", *Discuss. Math. Differ. Inclus., Control Optim.*, vol. 20, pp. 147–158, 1999.

[10] S. Flåm and J. Zowe, "Relaxed outer projections, weighted averages and convex feasibility", *BIT*, vol. 30, pp. 289–300, 1990.

[11] S. Kim, A. Hyunsil, and C. Seong-Cheol, "Variable target value subgradient method", *Math. Progr.*, vol. 49, pp. 356–369, 1991.

[12] K. Kiwiel, "Monotone Gram matrices and deepest surrogate inequalities in accelerated relaxation methods for convex feasibility problems", *Lin. Algebra Appl.*, vol. 215, pp. 27–33, 1997.

[13] K. Kiwiel, "The efficiency of subgradient projection methods for convex optimization", Part I: "General level methods", *SIAM Control Optim.*, vol. 34, no. 2, pp. 660–676, 1996.

[14] K. Kiwiel, "Block-iterative surrogate projection methods for convex feasibility problems", *Lin. Algebra Appl.*, vol. 15, pp. 225–259, 1995.

[15] T. Kręglewski, J. Granat, and A. P. Wierzbicki, *IAC-DIDAS-N – A Dynamic Interactive Decision Analysis and Support System for Multicriteria Analysis of Nonlinear Models, v. 4.0*. Collaborative Paper, CP-91-101. Laxenburg (Austria): International Institute for Applied Systems Analysis, June 1991.

[16] C. Lemaréchal, A. S. Nemirovskii, and Yu. Nesterov, "New variants of bundle methods", *Math. Progr.*, vol. 69, pp. 111–147, 1995.

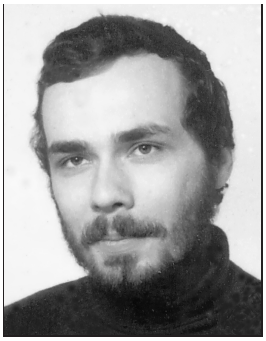
[17] M. Makowski, *LP-DIT data interchange tool for linear programming problems (version 1.20)*. Working Paper, WP-94-36. Laxenburg (Austria): International Institute for Applied Systems Analysis, 1994.

[18] M. Minoux, "Network synthesis and optimum network design problems: models, solution methods and applications", *Network*, vol. 19, pp. 313–360, 1989.

[19] T. Polyak, "Minimization of unsmooth functionals", *Zh. Vychisl. Mat. Fiz.*, vol. 9, *Mat. Fiz.*, vol. 9, pp. 14–29, 1969.

[20] M. Shchepakin, "On a modification of a class of algorithms for mathematical programming", *Zh. Vychisl. Mat., Mat. Fiz.*, vol. 19, pp. 1387–1395, 1979 (in Russian).

[21] A. Wierzbicki, "A penalty function shifting method in constrained static optimization and its convergence properties", *Archiw. Autom. Telemech.*, vol. XVI, no. 4, pp. 396–416, 1971.



Paweł M. Białoń was born in Warsaw in 1971. He received his M.Sc. in computer science from Warsaw University of Technology in 1995. Currently he is employed by National Institute of Telecommunications in Warsaw. His research focuses on nonlinear optimization methods and on decision support and is directed towards the Ph.D. degree. He has participated in several projects applying decision support methods in the telecommunications, agricultural and environmental areas.
e-mail: P.Bialon@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Paper

A new multiple objective dynamic routing method using implied costs

José Craveirinha, Lúcia Martins, Teresa Gomes, Carlos H. Antunes, and João N. Clímaco

Abstract — There are advantages in considering the routing problem in integrated communication networks as a multiobjective shortest path problem, having in mind to grasp eventual conflicts and trade-offs among distinct objectives and quality of services (QoS) constraints. On the other hand the utilisation of dynamic routing methods in various types of networks is well known to have significant impact on network performance and cost, namely in overload and failure conditions. This paper presents the detailed formulation of a proposal of a multiple objective dynamic routing method (MODR) of periodic state dependent routing type, enabling to represent distinct QoS related metrics and requirements in a consistent manner. The MODR method present formulation is based on a multiple objective shortest path model with constraints and is prepared to use implied costs as one of the metrics. Alternative paths for each traffic flow are changed as a function of periodic updates of certain QoS related parameters estimated from real time measurements on the routes and trunks of the network. Such paths are computed by a specialised and efficient variant of a bi-objective shortest path constrained algorithm, developed for the MODR, enabling to incorporate flexible requirements on the QoS metrics. The architecture of the routing system is discussed together with the features of its main modules. An illustrative example of application of the MODR path calculation module to a circuit-switched type network using blocking probability and implied cost as metrics, is also presented, considering different overload conditions.

Keywords — dynamic routing, multicriteria decision support systems, traffic management.

1. Introduction

The evolution of multiservice network functionalities leads, in terms of teletraffic engineering, to the necessity of dealing with multiple, fine grain and heterogeneous QoS requirements. When applied to routing mechanisms this concern led, among other developments, to a new routing paradigm designated as QoS routing, which involves the selection of a chain of network resources satisfying certain QoS requirements and seeking simultaneously to optimise route associated metrics (or a sole function of different metrics) such as a cost, delay, number of hops or blocking probability. This trend makes it necessary to consider explicitly distinct metrics in routing algorithms such as in references [1, 2, 3] or [4]. In this context the path selection problem is normally formulated as a shortest path problem with a single objective function, either a single metric or encompassing different metrics. QoS requirements are then incorporated into these models by means of additional constraints and the path selection problem (or routing problem in a strict sense) is solved by resorting to different types of heuristics. Since the mathematical models have inherently a network structure which renders them to be tackled in an effective way by specialised and efficient algorithms, the introduction of additional constraints destroys some underlying properties and implies a heavier computational burden.

Therefore there are advantages in considering the routing problem of this type, subject to multiple constraints as a multiple objective problem. Note that in a multiple objective context involving multiple, potentially conflicting, incommensurate objective functions, the concept of optimal solution in single objective problems (unique in general) gives place to the concept of non-dominated solutions (feasible solutions for which no improvement in any objective function is possible without worsening at least one of the other objective functions). Multiple objective routing models thus enable to grasp the trade-offs among distinct QoS requirements by treating in a consistent manner the comparison among different routing alternatives. This type of approach was previously proposed by the authors [5] to solve a static routing problem, formulated as a multiple objective shortest path problem, and using a particularly efficient algorithmic approach.

On the other hand the utilisation of dynamic routing in various types of networks is well known to have a quite significant impact on network performance and cost, namely considering time-variant traffic patterns, overload and failure conditions (see for example [6] and [7]).

The objective of this paper is to present a formulation of a multiple objective dynamic routing method (or MODR) that may be envisaged as a new type of periodic state dependent routing (PSDR) method. The MODR method is based on a multiple objective routing paradigm and incorporates a dynamic alternative routing principle, as well as the utilisation of the concept of implied cost in [8] as one of the metrics of the routing problem model. Other feature of MODR is the capability of defining preference regions (concerning the search for alternative paths) which may change dynamically, through variable boundary values. The paper is organised as follows. Section 2 is a concise review of the multiple objective static routing model in [5]. The main features of the proposed MODR method are presented in Section 3 and the main modules and functionalities of a centralised MODR architecture, are discussed. The char-

acteristics of the route calculation algorithm developed for the MODR, are presented in Section 5. Section 6 gives the model proposed for calculating dynamically changing estimates of the coefficients needed by a bi-objective version of the MODR method. An example of application of the MODR method to a fully meshed circuit-switched network is shown is Section 7 in order to illustrate relevant features of the proposed alternative route calculation method and its inherent capabilities. Finally, some conclusions and lines for further work on this matter will be outlined in the conclusion section.

2. Review of a multiple objective routing principle

The static routing principle and the basic algorithm from which the MODR routing method was derived, were proposed in [5]. This approach formulates the static routing problem as a multiple objective shortest path problem and uses a particularly efficient algorithmic approach. This algorithm computes non-dominated paths by optimising weighted-sums of the multiple objective functions, to determine solutions which belong to the boundary of the convex hull of the union of the set $\mathcal{Z}$ of the non-dominated solutions with $\{z \in \mathbb{R}^K | z \geq 0\}$, namely vertex solutions. It uses a very efficient k-shortest path algorithm [9], to search for unsupported non-dominated solutions within duality gaps (which are solutions located to the inside of the convex hull). Also it enables that QoS requirements may be expressed as additional (soft) constraints on the objective functions values in terms of requested and acceptable thresholds for each metric, which define preference regions in the objective functions space. Recalling the general formulation of the multiple objective shortest path problem with K-objective functions, where each function is associated with a particular metric:

$$\min\ z^n = \sum_{l_k=(v_i,v_j)\in L} \mathcal{C}_k^n x_{ij} \quad (n=1,\ldots,K) \tag{1}$$

s.t.

$$\begin{aligned}
&\sum_{v_j\in V} x_{sj} = 1\\
&\sum_{v_i\in V} x_{ij} - \sum_{v_q\in V} x_{jq} = 0 \quad \forall v_j \in V,\ (v_j \neq s,t)\\
&\sum_{v_i\in V} x_{it} = 1\\
&x_{ij} = 0,1 \quad \forall l_k = (v_i,v_j) \in L\\
&\text{(Problem} \quad \mathcal{P}^{(K)}\text{)},
\end{aligned} \tag{2}$$

where $\mathcal{C}_k^n$ is the cost associated with metric n ($n = 1,2,\ldots,K$) on arc $l_k = (v_i,v_j) \in L$ of the graph (V,L), V is the node set and L the arc set of the network structure; the variables x_{ij} enable to define a solution (path) p from node s to node t by taking the value 1 if the arc $(v_i,v_j) \in p$ and 0 otherwise. Note that the cost of a path is a real-valued vector $\mathcal{C}_p = (\mathcal{C}_p^1,\ldots,\mathcal{C}_p^K)$ with $\mathcal{C}_p^n = \sum_{l_k\in p} \mathcal{C}_k^n$ being the cost associated with metric n. In general there is no feasible solution which minimises all objective functions simultaneously. Since there is no guarantee of the existence of this ideal optimal solution, the resolution of this static multiple objective routing problem aims at finding a best compromise path from the set of non-dominated solutions, according to some relevant criteria defined by the decision maker. Non-dominated solutions can be computed by optimising a scalar function which is a convex combination of the K-objective functions:

$$\min\ z = \sum_{l_k\in L} \mathcal{C}_k x_{ij} \tag{3}$$

with the same constraints of the original problem and $\mathcal{C}_k = \sum_{n=1}^K \varepsilon_n \mathcal{C}_k^n$ where $\varepsilon = (\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_K) \in \mathcal{E} = \{\varepsilon : \varepsilon_n \geq 0,\ n = 1,\ldots,K \wedge \sum_{n=1}^K \varepsilon_n = 1\}$. However, by using this form of scalarization only supported non-dominated paths (that is those which are located on the boundary of the convex hull) may be found. Nevertheless non-dominated solutions located in the interior of the convex hull may exist. The mentioned algorithmic approach implemented for two objective functions, designated hereafter as basic multi-objective routing algorithm (BMRA) resorts to an extremely efficient *k*-shortest path algorithm [9] to search for this specific type of non-dominated paths. It must be noted that in the calculation of non-dominated solutions, namely unsupported non-dominated solutions, it seems useful considering reference point approaches. However in the case of shortest-path problems, the recent development of an extremely efficient algorithm (the MPS algorithm [9]) for the *k*-shortest path problem, creates the possibility of developing very efficient techniques for calculating supported and unsupported non-dominated solutions in this particular context.

Blocking probability is a key metric to be considered in the case of traffic flows working on a loss basis. This metric can be easily transformed into an additive metric (as required by the algorithm) by associating with each arc l_k, $-\log(1-B_k)$, B_k being the blocking probability on l_k. Other common metrics such as cost, delay and hop-count follow the additive aggregation function. Also path bandwidth may be a metric of the model by using the concave aggregation rule [5]. The other features and the details of the BMRA, are described in [5].

3. The MODR method

The multiple objective dynamic routing method proposed in this paper may be envisaged as a new type of periodic state dependent routing [10] method based on a multiple objective routing paradigm. The PSDR class of routing methods [10] is based on a centralised type of control which provides routing decisions for each pair of exchanges based on periodical updates of the number of free circuits in each trunk of the network using a typical update period of 10 s.

In its general formulation the MODR here discussed has the following main features: i) paths are changed dynamically as a function of periodic updates of certain QoS related parameters obtained from real-time measurements, using a multiple objective principle which enables to consider, in a consistent manner, eventually conflicting QoS metrics; ii) it uses a very efficient version of the BMRA, designated hereafter as a modified multiobjective routing algorithm (MMRA), prepared to deal with the selection of alternative paths in a dynamic alternative routing context; iii) the present version of the method uses implied costs in the sense defined by Kelly in [8] as one of the metrics to be incorporated in the underlying multiple objective model; iv) it enables to specify required and/or requested values for each metric (associated with predefined QoS criteria) as in the BMRA; such values define preference regions on the objective functions space, which may change dynamically, in a flexible way, through variable boundary values; this capability is attached to a routing management system (described in the next section) and enables to respond to various network service features and to variable working conditions.

As for the way in which the paths are selected in the MODR method, the first path is always the direct route whenever it exists. The remaining routes for traffic flows between an exchange pair are selected from the MMRA, taking into account the defined priority regions. This may be easily formalised in the following manner. Let R be the number of routes attempted by a call of each traffic flow $(r^1(f), r^2(f), \ldots, r^R(f))$ and $S(f)$ be the ordered set of solutions selected by MMRA $\{s_1, s_2, \ldots, s_R\}$ for flow f as a function of the defined priority regions for flow f and $r_d(f)$ the possible direct route:

$$\text{1st case}: r_d(f) = \emptyset \Rightarrow r^i(f) = s_i, \quad (i = 1, 2, \ldots, R) \quad (4)$$

$$\text{2nd case}: r_d(f) \neq \emptyset \Rightarrow \begin{cases} r^1(f) = r_d(f) \\ r^i(f) = s'_{i-1}, \quad (i = 2, \ldots, R) \end{cases} \quad (5)$$

with $S'(f) = S(f) \backslash \{r_d(f)\} = \{s'_1, \ldots, s'_{|S'(f)|}\}$. All these features aim at turning more effective and flexible the application of the multiple objective routing approach to a dynamic routing method, having in mind the multifaceted nature of traffic flows and the variability of a network working conditions.

4. Architecture of the MODR system

A periodic centralised routing technique must be able of computing, every T seconds, for every traffic flow f associated with each exchange pair of the network, the routing tables better fitted to the network state, having in mind to obtain the best possible network performance according to the routing method. For this purpose the MODR routing system must receive from the network nodes the necessary QoS related measurements. As can be seen in Fig. 1, there are two following main subsystems:

Routing control/real time management, which is the core of the MODR method architecture. The core of this subsystem is the MMRA path calculation algorithm (it constitutes the basis of the alternative path calculation module), described in the next section. The inputs to the MMRA are the current values of the coefficients of the objective functions and the associated (soft) constraints which define preference regions in the objective function space. The routing control also includes a **network data** module that contains all the necessary information about the network configuration that is important for the coefficient calculation.

Routing management system which operates on a wider time scale as compared with the previous subsystem. The main functions of this subsystem are the following: the specification of relevant parameters for the routing control such as the path update period T and the frequency $1/\tau$ of real time measurements of QoS related parameters; a change in the maximum number R of alternative paths is also possible; the specification of threshold values for the route metrics (typically **required** and/or **acceptable** values) which enable to define the preference regions for alternative path selection according to the MMRA (see Section 5). Such values may be modified empirically at any time, as a result of the intervention of the operator of the routing management system (an essential part of any traffic management system). Those values, namely the required (or "desirable") values, also may vary periodically (with period T) as a result of the changes in the marginal optimal values of the objective functions using a criterion as the one defined in Section 5 for the priority regions; these functions are associated with the parametrisation module in Fig. 1.

Also, other more specialised mechanisms, namely related to the functional and/or transport network levels, may also be included in this subsystem in order to reinforce the network survivability under particular failure or overload conditions through the module designated in Fig. 1 as other routing management mechanisms. Finally the parametrisation of service protection mechanisms such as circuit reservation could be performed through this module.

The core of the MODR method, i.e., the routing control/real time management subsystem, can be, in principle, decentralised to the network nodes without too much effort when the network is totally meshed, assuming each node has an associated path calculation module. In this case, some additional signalling messages, which must include the values of the implied costs and blocking probabilities on the links, must be exchanged between the nodes because each node must have information about these data related to all the links, in order to be able to compute multi-objective paths via the MMRA. It must be noted that the implied cost for the adjacent links of each node can be computed in this case because each node knows the implied cost for all links in the network needed for its path calculations.

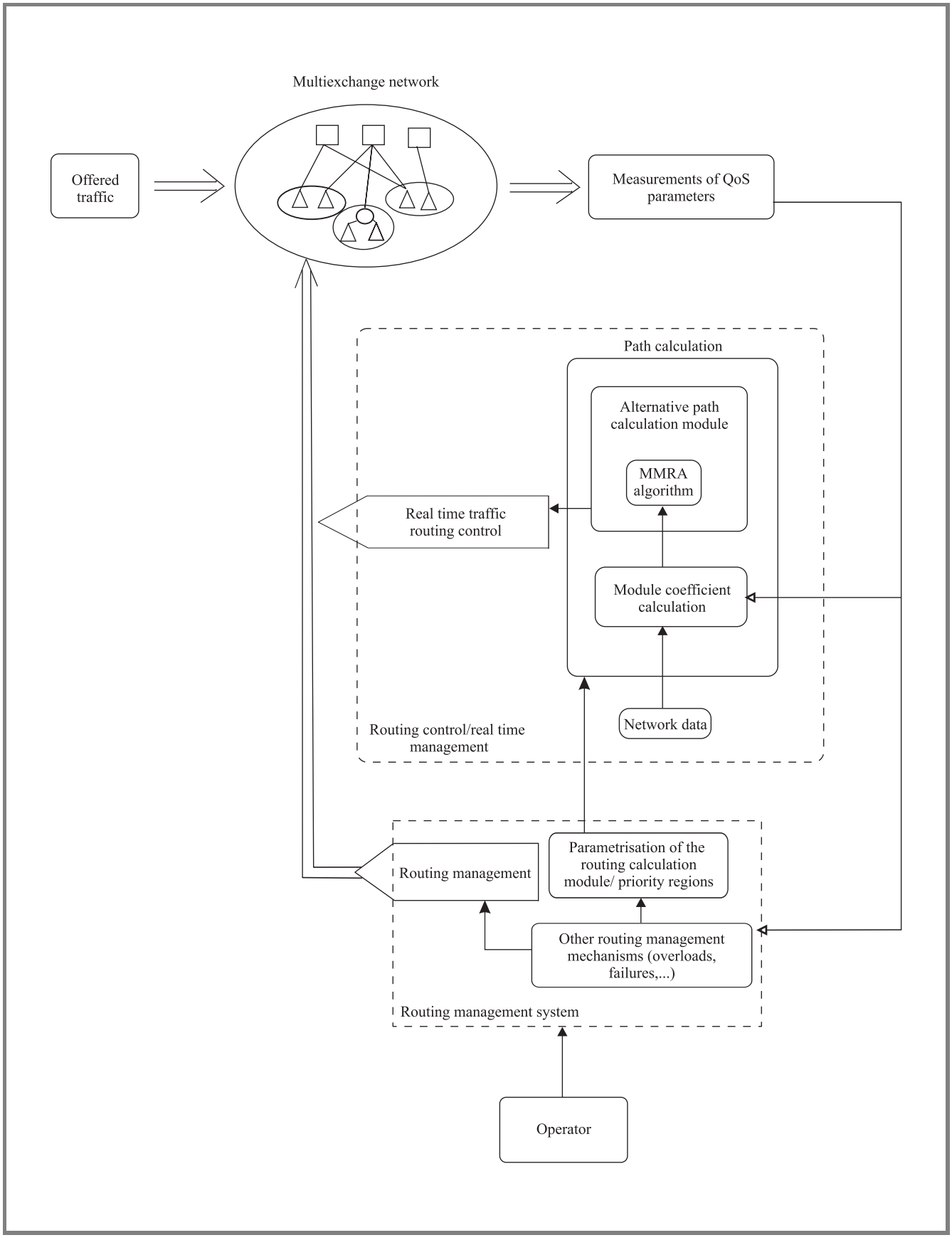


Fig. 1. Architecture of the MODR system.

5. The route calculation algorithm MMRA

The modified multiobjective routing algorithm is a new variant of the BMRA proposed in [5], adapted and optimised to the needs of the MODR method. Its basic features and differences with respect to the BMRA are the following: i) it enables to search for and select non-dominated or dominated paths for alternative routing purposes; ii) it uses as sub-algorithm for calculating k-shortest paths a new variant of the k-shortest path algorithm in [9], developed in [11] by some of the authors for solving the k-shortest path problem with a constraint on the maximum number of arcs per path since this is a typical constraint considered in practical routing methods; iii) the search direction in the objective function space is a 45° straight line; this is justified by the variable nature of the metrics in an integrated service network environment and the possibility of dynamic variation of the priority regions; iv) the priority regions for alternative path selection have a flexible configuration that varies dynamically as a result of the periodic alterations in the objective function coefficients; furthermore the bounds of those regions may also be changed through some of the functionalities associated with the parametrisation module of the routing and management system.

Concerning the specification of the requested and/or acceptable values for the metrics, distinct cases should be envisaged. In the case of blocking probabilities, delays and delay jitter for example, such values can be obtained from network experimentation and/or from ITU-T standardisation or recommendations for various types of networks and services. On the other hand, in the case of costs, namely implied costs, included in the present model, it is more difficult to define a priori such values, since no general criteria are known for these quantities. In the application example described in Section 7 for a circuit-switched network with loss traffic, the following approach was used. As for the path blocking probabilities, having in mind that alternative routing is used, the value required for path blocking, B_{req} is obtained from an approximation based on the mean call blocking on the trunks, calculated when the network is dimensioned for a typical end-to-end blocking probability such as 0.5%:

$$B_{req} = 1 - \prod_{k=1}^{D} \left(1 - B_{k_{med}}\right) = 1 - \left(1 - B_{k_{med}}\right)^{D} \quad (6)$$

$$\text{by considering } B_{k_{med}} = \frac{1}{|L|} \sum_{l_k \in L} B_k^d , \quad (7)$$

B_k^d being the calculated average call congestion on link l_k resulting from the dimensioned network and D the maximum number of links per path. Note that this criterion intends to guarantee that the constraint B_{req} is satisfied by any path selected by the MMRA in the priority regions for which $B \leq B_{req}$. As for the implied costs obtained from the model described in Section 6, analogous criterion leads to the required implied cost path value:

$$c_{req} = \sum_{k=1}^{D} c_{k_{med}} = D c_{k_{med}} \quad (8)$$

$$\text{by considering } c_{k_{med}} = \frac{1}{|L|} \sum_{l_k \in L} c_k^d , \quad (9)$$

c_k^d being the implied cost value obtained for link l_k, using an adequate form of numerical fixed point iteration for the engineered network. For obtaining the acceptable values B_{acc} and c_{acc} for the associated path metrics an analogous procedure was used by dimensioning the network for a typical end-to-end blocking value such as 1%.

Taking into account the variability in time of the marginal optimum Op^n of each objective function z^n, the following cases may occur regarding the priority regions in the bi-objective model, designating by M a generic metric: i) $M_{req}^n > Op^n$ for $(n = 1,2)$ in which case there are 5 priority regions, analogously to the static routing example in [5]; ii) if $Op^n < M_{req}^n < M_{acc}^n$ for one of the objective functions and $M_{req}^m < Op^m < M_{acc}^m$, $(m \neq n)$ then there are 3 priority regions as illustrated in Fig. 2; iii) if $M_{req}^n < Op^n < M_{acc}^n$ for $(n = 1,2)$ then there are two priority regions only, of type C and D; iv) if $M_{acc}^n < Op^n$ for $(n = 1,2)$, case in which there is only region D for searching for last chance route(s), defined by the intervals $[Op^n, M_L^n]$.

It is assumed the following convention: A_n is a region which satisfies both requirements (M_{req}^n and M_{acc}^n) for z^n $(A \equiv A_1 \cap A_2)$; B_n a region which satisfies M_{req}^n and M_{acc}^n but does not satisfy M_{req}^m although satisfying M_{acc}^m $(m \neq n)$; C a region which satisfies M_{acc}^n $(n = 1,2)$ but not M_{req}^n $(n = 1,2)$ and D is the last priority region, corresponding to search for a last chance route in the cases where particu-

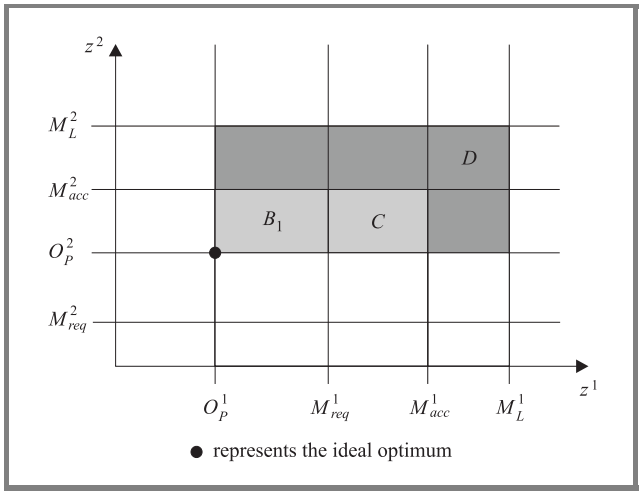


Fig. 2. Case *ii*) for dynamic priority regions (B_1, C, D).

larly unfavorable network conditions occur. M_L^n is the more relaxed bound considered for z^n.

6. Estimates of the model coefficients

The periodical computations for updating the node routing tables for the nth time interval, of duration T, are based on QoS related measurements obtained from the network nodes in the $(n-1)$th time interval and possibly on values obtained in previous time intervals. Moving-average iterations, as suggested in [8], may be used for this purpose as will be explained next.

One of the parameters to be estimated is the blocking probability on each link l_k. In the time interval, $[(i-1)\tau, i\tau]$ – τ is the measurement period – an estimate of the blocking probability is $\mathcal{B}_k(i) = N_{i_{loss}}^T / N_i^T$, where $N_{i_{loss}}^T$ is the number of lost calls and N_i^T is the number of call attempts in that interval. For simplicity it will be assumed that the updating period τ of the measurements coincides with the routing updating period T. If $\beta_k(n)$ is an estimator of blocking probability for link l_k for the nth time interval then it can be calculated through the first order moving-average iteration:

$$\beta_k(n) = (1-b)\beta_k(n-1) + b\mathcal{B}_k(n-1), \qquad (10)$$

where $b \in [0,1]$ reflects a balance between accuracy of estimation and speed of response. This estimation is carried out in the module **measurements of QoS parameters**. The information concerning the values $\beta_k(n)$, obtained at one of the adjacents nodes of l_k should then be conveyed to the centralised routing system. The forecasted carried traffic on link l_k for the nth time interval, $y_k(n)$, can also be estimated in a similar manner, i.e., $y_k(n) = (1-b')y_k(n-1) + b'Y_k(n-1)$, where $Y_k(n)$ is the estimate for the average carried traffic in the $[(n-1)T, nT]$ period, which was communicated to the **measurements of QoS parameters** module. The carried traffic estimate for a path $r^i(f)$, $x_{r^i(f)}(n)$, associated with traffic flow f between a certain pair of nodes can be, in principle, estimated in a similar manner. Nevertheless, in dynamic routing the set of routes available for each pair of nodes may change in successive periods, so the moving-average iterations must be adapted to cope with this: if a path was selected as a possible route in the $(n-1)$th time interval but not in the $(n-2)$th interval, then only the measurements in the $(n-1)$th interval should be used in the estimation scheme for the nth interval. Therefore b should be made equal to 1 in this situation. The details of the easy adaptation of these estimation schemes to the cases in which it is used a measurement updating period τ shorter than T, are explained in [12].

Two possible approaches for calculating implied costs estimates in the context of MODR with $R = 2$, are now presented for a circuit-switched type network with single circuit calls.

The first one, using a moving-average iteration will be described next. Let $c_k(n)$ be an estimate for c_k, the implied cost associated with link l_k and $s_{r^i(f)}(n)$ be an estimate for $s_{r^i(f)}$, the surplus value of a call on route $r^i(f)$ $(i = 1,2)$, for the nth time interval. Designating by $w(f)$ the expected revenue obtained from an accepted call of traffic flow f, then one may easily obtain (see details in [12]) from equation (7.11) in [8] the following iterative scheme:

$$\begin{aligned}
c_k(n) &= (1-a)c_k(n-1) + \\
&+ aF_k(n)\left[\sum_{f:l_k\in r^1(f)} \frac{x_{r^1(f)}(n)}{y_k(n)}\left(c_k(n-1) + s_{r^1(f)}(n-1)\right) + \right. \\
&\left. + \sum_{f:l_k\in r^2(f)} \frac{x_{r^2(f)}(n)}{y_k(n)}\left(c_k(n-1) + s_{r^2(f)}(n-1)\right)\right] \qquad (11)\\
s_{r^2(f)}(n) &= w(f) - \sum_{l_j\in r^2(f)} c_j(n-1) \\
s_{r^1(f)}(n) &= w(f) - \sum_{l_j\in r^1(f)} c_j(n-1) + \\
&- \left(1 - L_{r^2(f)}(n)\right) s_{r^2(f)}(n),
\end{aligned}$$

where

$$F_k(n) = z_k(n)\left[E(z_k(n), C_k - 1) - E(z_k(n), C_k)\right]. \qquad (12)$$

Here $E(A,C)$ is the value of the Erlang-B function for offered traffic A and C circuits, $z_k(n)$ is the estimate of the offered traffic on link l_k given by $y_k(n)/(1-\beta_k(n))$ and $L_{r^i(f)}(n)$ the blocking probability estimate of $r^i(f)$, for the nth time interval. The meaning of the auxiliary parameter a is analogous to b.

The second approach, more rigorous, although heavier in terms of required numerical calculations, is based on the execution of a fixed point iteration at the beginning of each period of duration T. Let $c_k^{j_n}(n)$ designate an estimate for c_k, and $s_{r^i(f)}^{j_n}(n)$ an estimate for the surplus value of a call on route $r^i(f)$, for the nth time interval, using this approach.

Then the calculation procedure is the following:

$$\begin{aligned}
c_k^{j+1}(n) &= (1-a')c_k^j(n) + \\
&+ a'F_k(n)\left[\sum_{f:l_k\in r^1(f)} \frac{x_{r^1(f)}(n)}{y_k(n)}\left(c_k^j(n) + s_{r^1(f)}^j(n)\right) + \right. \\
&\left. + \sum_{f:l_k\in r^2(f)} \frac{x_{r^2(f)}(n)}{y_k(n)}\left(c_k^j(n) + s_{r^2(f)}^j(n)\right)\right] \qquad (13)\\
s_{r^2(f)}^{j+1}(n) &= w(f) - \sum_{l_j\in r^2(f)} c_j^j(n) \\
s_{r^1(f)}^{j+1}(n) &= w(f) - \sum_{l_j\in r^1(f)} c_j^j(n) + \\
&- \left(1 - L_{r^2(f)}(n)\right) s_{r^2(f)}^{j+1}(n)
\end{aligned}$$

with $j = 0, 1, \ldots, j_n - 1$ and $c_k^0(n) = c_k^{j_{n-1}}(n-1)$, where j_n is the number of iterations used to calculate $c_k(n)$. The parameter a' in this approach is the damping parameter of the fixed point iteration scheme. Here a' should be chosen in order to guarantee the convergence of the iterations in (13).

7. Application example

A fully-meshed 6 node circuit-switched network with single circuit calls was dimensioned according to the method in [13] for 0.005 end-to-end blocking probability, assuming one alternative path to the direct route. The obtained network is characterised in Table 1. For the definition of the priority regions bounds, the required values for each metric of the paths, are given according to Eqs. (6), (8): $B_{req} = 1 - (1 - B_{k_{med}})^2$, $c_{req} = 2c_{k_{med}}$, where $B_{k_{med}}$ is the average link blocking probability and $c_{k_{med}}$ is the average implied cost of the links, both obtained for the network in Table 1, using fixed point iteration schemes. The acceptable bounds are obtained by a similar approach for the same network topology dimensioned for end-to-end blocking probability of 0.01 (for the same traffic offered as in Table 1).

Table 1
Network of the application example

O-D pair	Link capac.	Offered traf.	Intermediate node
1-2	36	27	3
1-3	13	6	4
1-4	33	25	5
1-5	27	20	6
1-6	31	20	2
2-3	29	25	4
2-4	17	10	5
2-5	37	30	6
2-6	25	20	1
3-4	17	11	5
3-5	14	8	6
3-6	19	13	1
4-5	13	9	6
4-6	27	20	1
6-6	18	12	1

These bounds are marked in Figs. 3 (a) and (b) where the search direction is a 45° straight line. Two examples for illustrating the application of the MMRA model have been selected, showing the results of the search for two paths with at most two links. The following notes may be drawn from this experimental study. Blocking probability and implied cost may be conflicting criteria, although in general they are not orthogonal. In example (a) described in Table 2 (network with 5% overload) it is shown that the three first generated solutions are non-dominated. In example (b) described in Table 3 (network with 10% overload in all traffic flows from node 1) although the first feasible solution is the ideal optimal solution (meaning that in this case the two metrics are not conflicting and lead to the same optimal solution), the second and the third solutions are dominated solutions not comparable in a multicriteria sense. In fact, in various fully meshed networks dimensioned by the same algorithm as the network in Table 1, more than 50% of node pairs, for the first and/or the second path, path blocking probability and path implied cost were conflicting objectives. This fully justifies the potential advantages of the MODR principle. In both examples

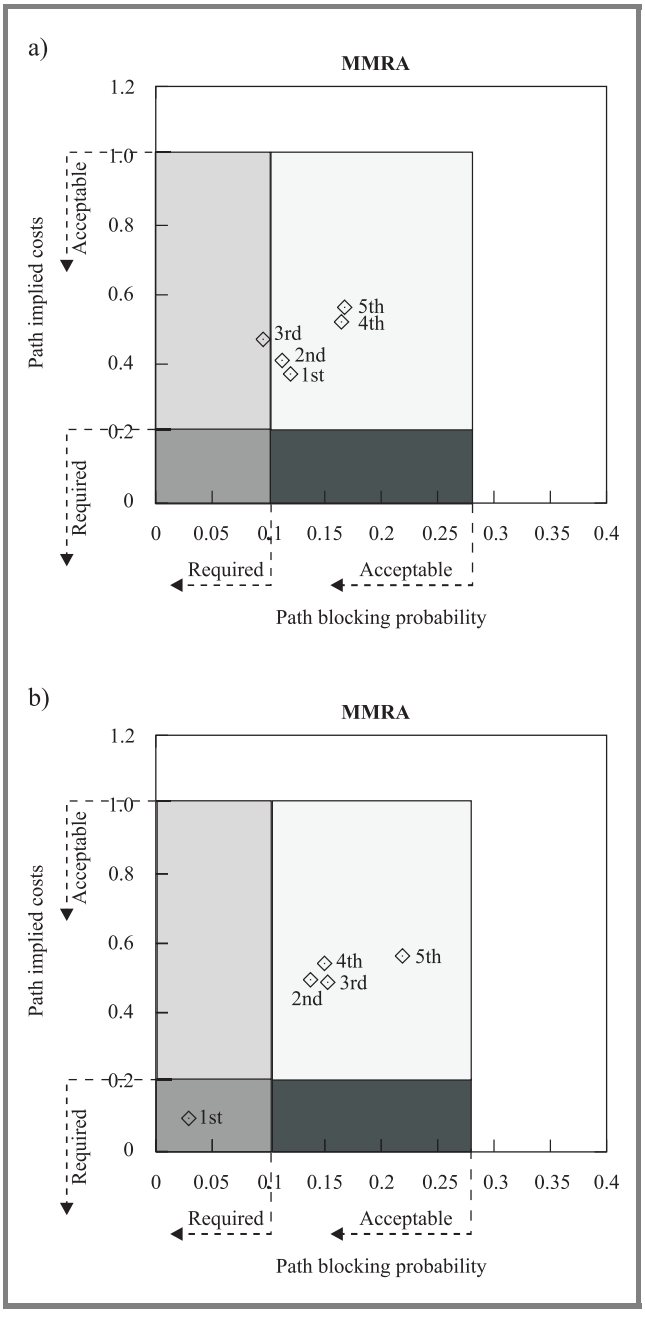


Fig. 3. (a) Network with 5% overload; (b) network with 10% overload in all traffic flows from node 1.

Table 2
Network with 5% overload

i	Blocking	Implied cost	Generated paths	Selected paths	Type	Preference region type
1	0.119621	0.371881	$2 \to 3$	$r^1(f)$	Non-dominated	C
2	0.112106	0.410792	$2 \to 1 \to 3$		Non-dominated	C
3	0.0953536	0.471365	$2 \to 5 \to 3$	$r^2(f)$	Non dominated	B_1
4	0.164738	0.521878	$2 \to 6 \to 3$		Dominated	C
5	0.167548	0.563675	$2 \to 4 \to 3$		Dominated	C

Table 3
Network with 10% overload in traffic flows from node 1

i	Blocking	Implied cost	Generated paths	Selected paths	Type	Preference region type
1	0.0292787	0.0950544	$5 \to 3$	$r^1(f)$	**Ideal solution**	A
2	0.137195	0.493963	$5 \to 2 \to 3$	$r^2(f)$	Dominated	C
3	0.152468	0.487164	$5 \to 6 \to 3$		Dominated	C
4	0.149753	0.541269	$5 \to 1 \to 3$		Dominated	C
5	0.218845	0.562676	$5 \to 4 \to 3$		Dominated	C

represented graphically in Figs. 3 (a) and (b) the number of generated paths depends on the fact that the MMRA algorithm does not stop searching for paths while there is the possibility of finding a solution in a lower preference region not yet fully covered.

The results in Tables 2 and 3 are graphically presented in Figs. 3 (a) and (b) with the preference regions clearly marked. Note that the last choice region D is not represented in these graphics.

8. Conclusions and further work

A new MODR method is proposed having as basis a multiple objective shortest path model, tackled by a specialised and very efficient algorithm which enables to find a predefined number of alternative paths which may change periodically as function of QoS related parameter measurements. The present formulation of the method uses implied costs as one of the metrics, which enable to represent the knock-on effects of accepting a call on a given route upon the other routes (see [8]), in the context of the MODR. The modules and functionalities of a MODR centralised architecture were also outlined as well as the possibility of decentralising some of its basic functions in the case of fully meshed networks. Other important feature of the method is the capability of defining in a dynamic and flexible way, preference regions for selection of alternative routes between every pair of nodes.

An application example of the MODR principle to a fully meshed circuit-switched network was also presented which showed that path implied cost and blocking probability may be conflicting objectives in many practical network working conditions, namely in cases of global or local overload. This fully justifies, in our opinion, potential advantages of a MODR type method.

Further work should be focused on a number of open issues, namely: the evaluation of network performance under MODR in different traffic conditions using an appropriate simulation platform, in the context of multiservice networks. Also the parametrisation of the method namely in terms of the tunning of the updating periods for the measurements and routing tables should be addressed through simulation. Also the incorporation of service protection mechanisms, already foreseen in the routing architecture should be addressed in the near future having in mind the known significant impact of these mechanisms in network performance, as shown in [14] in the case of adaptative dynamic routing. The extension of the MODR principle to broadband networks is being developed at the present.

Acknowledgements

We thank Tiago de Sá for allowing us to use his implementation of the dimensioning method in [13].

This work was partially supported by FCT, project PRAXIS/P/EEI/13219/1998, “A study on state dependent and multi-objective dynamic routing in multi-service networks” and the project POCTI/32379/MAT/2000.

References

[1] W. C. Lee, M. G. Hluchyj, and P. A. Humblet, "Routing subject to quality of service constraints in integrated communication networks", *IEEE Network*, pp. 46–55, July/Aug. 1995.

[2] R. Vogel, R. G. Herrtwich, W. Kalfa, H. Wittig, and L. C. Wolf, "QoS – based routing of multimedia streams in computer networks", *IEEE J. Selec. Areas Commun.*, vol. 14, no. 7, pp. 1235–1244, 1996.

[3] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications", *IEEE J. Selec. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, 1996.

[4] C. Pornavalai, G. Chakraborty, and N. Shiratori, "Routing with multiple QoS requirements for supporting multimedia applications", *Telecommun. Syst.*, no. 9, pp. 357–373, 1998.

[5] C. H. Antunes, J. Craveirinha, J. Clímaco, and C. Barrico, "A multiple objective routing algorithm for integrated communication networks" in *ITC-16 Teletraffic Engineering in a Competitive World*, P. Key and D. Smith, Eds., Elsevier Science B.V., June 1999, vol. 3b, pp. 1291–1300.

[6] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*. Addison-Wesley, 1990.

[7] G. R. Ash, *Dynamic Routing in Telecommunications Networks*. McGraw-Hill, 1998.

[8] F. P. Kelly, "Routing in circuit-switched networks: optimization, shadow prices and decentralization", *Adv. Appl. Probab.*, vol. 20, pp. 112–144, 1988.

[9] E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. Santos, "Deviation algorithms for ranking shortest paths", *Int. J. Foundat. Comput. Sci.*, no. 10, pp. 247–263, 1999.

[10] ITU-T Group 2, "Dynamic routing interworking". Draft text for Rec. E.1XX, 1997.

[11] T. Gomes, L. Martins, and J. F. Craveirinha, "An efficient algorithm for calculating k shortest paths with a maximum number of arcs", *Investig. Oper.*, no. 21, pp. 235–244, 2001.

[12] J. Craveirinha, L. Martins, T. Gomes, C. H. Antunes, and J. Clímaco, "Formulation of a multiple objective dynamic routing method using implied costs – architecture and algorithms", Tech. Rep. ET-N8-3, INESC-Coimbra, Feb. 2001.

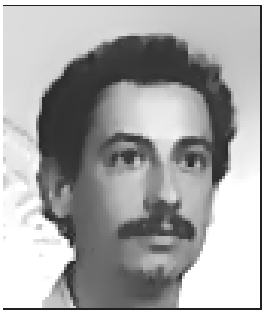
[13] G. R. Ash, R. H. Cardwell, and R. P. Murray, "Design and optimization of networks with dynamic routing", *Bell Syst. Tech. J.*, vol. 60, no. 8, pp. 1787–1820, 1981.

[14] G. R. Ash, "An analytical model for adaptative routing networks", *IEEE Trans. Commun.*, vol. 41, no. 11, pp. 1748–1759, 1993.

José Craveirinha received the M.Sc. (1981) and Ph.D. degrees in E.E.S. at the University of Essex (UK) (7/1984) and the tittle of "Agregacao" in E.E.S. Telecommunications at the University of Coimbra (7/1996). He is a Professor at the Department of Electrical Engineering and Computers, of the University of Coimbra and has coordinated a research group in Teletraffic Theory & Network Planning at INESC-Coimbra R&D Institute since 1986. He was a director of this institute in 1994–1999. His main scientific areas of research have been stochastic modelling of teletraffic, reliability analysis and planning of telecommunication networks. His main present interests are in traffic modelling and routing in Internet and multiple objective routing in multiservice networks.
e-mail: jcrav@deec.uc.pt
Departamento de Engenharia Electrotécnica
e Computadores
Universidade de Coimbra
Pinhal de Marrocos, 3030-290 Coimbra, Portugal
INESC Coimbra
Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

Lúcia Martins received her M.Sc. degree in computer science from the University of Coimbra, in 1988. She worked as a development engineer for six years in Portugal Telecom public operator. She is an assistant lecturer at the Department of Electrical Engineering and Computers, University of Coimbra, and a researcher at INESC Coimbra. She is completing a Ph.D. in telecommunications at the University of Coimbra in the area of multiple objective routing in multiservice networks.
e-mail: lucia@deec.uc.pt
Departamento de Engenharia Electrotécnica
e Computadores
Universidade de Coimbra
Pinhal de Marrocos, 3030-290 Coimbra, Portugal
INESC Coimbra
Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

Teresa Gomes received her Ph.D. degree in electrical engineering (telecommunications and electronics) from the University of Coimbra, in 1998. She is an Assistant Professor at the Department of Electrical Engineering and Computers, University of Coimbra, and a researcher at INESC Coimbra. Her research areas include reliability/quality of service analysis of telecommunications networks, teletraffic engineering and simulation.
e-mail: teresa@deec.uc.pt
Departamento de Engenharia Electrotécnica
e Computadores
Universidade de Coimbra
Pinhal de Marrocos, 3030-290 Coimbra, Portugal
INESC Coimbra
Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal



Carlos Henggeler Antunes received his Ph.D. degree in electrical engineering (optimisation and systems theory) from the University of Coimbra, in 1992. He is an Associate Professor at the Department of Electrical Engineering and Computers, University of Coimbra, and a researcher at INESC Coimbra. His research areas include multiple objective programming, decision support systems, energy planning and telecommunication network planning.
e-mail: ch@deec.uc.pt
Departamento de Engenharia Electrotécnica
e Computadores
Universidade de Coimbra
Pinhal de Marrocos, 3030-290 Coimbra, Portugal
INESC Coimbra
Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

João N. Clímaco is a Professor at the School of Economics of the University of Coimbra, Portugal and researcher at the INESC-Coimbra, Portugal. He obtained his M.Sc. (1978) in control systems from the Imperial College, University of London and his Ph.D. (1982) in electrical engineering – optimisation and systems theory from the University of Coimbra and the tittle of "Agregacao" in optimisation and systems theory at the University of Coimbra (7/1996). He was responsible for a R&D group in decision support systems at INESC and was a director of this institute in 1988–94. His current research interests are in multiple criteria decision aiding, decision support systems, location analysis and telecommunications planning.
e-mail: jclimaco@inescc.pt
Faculdade de Economia
Universidade de Coimbra
Av. Dias da Silva 3000 COIMBRA, Portugal

Paper

Implementation and performance of a new multiple objective dynamic routing method for multiexchange networks

Lúcia Martins, José Craveirinha, João N. Clímaco, and Teresa Gomes

Abstract — The paper describes new developments of a multiple objective dynamic routing method (MODR) for circuit-switched networks previously presented, based on the periodic calculation of alternative paths for every node pair by a specialised bi-objective shortest path algorithm (MMRA). A model is presented that enables the numerical calculation of two global network performance parameters, when using MMRA. This model puts in evidence an instability problem in the synchronous path computation model which may lead to solutions with poor global network performance, measured in terms of network mean blocking probability and maximum node-to-node blocking probability. The essential requirements of a heuristic procedure enabling to overcome this problem and select "good" routing solutions in every path updating period, are also discussed.

Keywords — dynamic routing, multiple objective routing, multiexchange telecommunication network performance.

1. Introduction

The evolution of multi-service telecommunications network functionalities has led to the necessity of dealing with multiple, fine grain and heterogeneous grade of service requirements. When applied to routing mechanisms this concern led, among other developments, to a new routing concept designated as QoS routing, which involves the selection of a chain of network resources satisfying certain QoS requirements and seeking simultaneously to optimise route associated metrics (or a sole function of different metrics) such as cost, delay, number of hops or blocking probability. This trend makes it necessary to consider explicitly distinct metrics in routing algorithms such as in references [12, 13] or [11]. In this context the path selection problem was normally formulated as a shortest path problem with a single objective function, either a single metric or encompassing different metrics. QoS requirements were then incorporated into these models by means of additional constraints and the path selection problem (or routing problem in a strict sense) was solved by resorting to different types of heuristics.

Therefore there are potential advantages in modelling the routing problem of this type as a multiple objective problem. Multiple objective routing models enable to grasp the trade-offs among distinct QoS requirements by enabling to represent explicitly, as objective functions, the relevant metrics for each traffic flow and treat in a consistent manner the comparison among different routing alternatives.

On the other hand, the utilisation of dynamic routing in various types of networks is well known to have a quite significant impact on network performance and cost, namely considering time-variant traffic patterns, overload and failure conditions (see for example [6] and [2]).

In a previous paper [5] the authors presented the essential features of a multiple objective dynamic routing method (MODR) of periodic state dependent routing type, based on a multiple objective shortest path model. In its initial formulation for multiexchange circuit-switched networks the model uses implied costs and blocking probabilities as metrics for the path calculation problem. Alternative paths for each node-to-node traffic flow are calculated by a specialised bi-objective shortest path algorithm, designated as modified multiple objective routing algorithm (MMRA), as a function of periodic updates of certain QoS related parameters estimated from real time measurements on the network. In other network environments in terms of underlying technologies and supplied services other QoS metrics can be easily integrated in this type of routing model.

The main objective of this paper is to present new developments of the MODR method, for circuit-switched networks, including a model for network performance evaluation under MODR and the discussion of a path instability problem associated with the MMRA model and of its consequences in terms of global network performance measured by two criteria (network mean blocking probability and maximal node-to-node blocking probability).

The paper begins by reviewing the main features of the MODR method and of the core node-to-node route calculation algorithm MMRA, based on a bi-objective shortest path model. Then it outlines an analytical model the numerical resolution of which gives the global network performance measured in terms of total traffic carried and maximal node-to-node blocking probability, when using MMRA and periodically time varying traffic matrices, for one class of service. This model enabled to put in evidence an instability problem in the synchronous path computation module, expressed by the fact that the paths computed by MMRA for all node pairs in each period tend to oscillate between a few sets of solutions many of which lead

to a poor global network performance. Having in mind to explicit this instability/inefficiency which results from the interdependencies between implied costs, blocking probabilities and computed paths and from the discrete nature of the multiple objective shortest path problem, a model (of bi-objective nature) for the global network performance evaluation, was developed. The essential requirements of a heuristic procedure enabling to overcome this problem and select "good" routing solutions in every path updating periods, are discussed. This heuristic will have to be based on an adequate selection of candidate second choice paths for possible change in each updating period. A criterion for selecting such paths will be proposed.
Finally the main conclusions from this paper will be drawn together with the presentation of the lines of undergoing developments of this work.

2. Review of the basic features of the MODR method

The MODR method [5] is based on the formulation of the static routing problem (calculation of the paths for a given pair of nodes assuming fixed cost coefficients in the objective functions) as a bi-objective shortest path problem, including "soft constraints" (that is constraints not directly incorporated into the mathematical formulation) in terms of requested and/or acceptable values for the two metrics. The formulation of the problem for circuit-switched networks uses as metrics, for loss traffic, implied cost (in the sense defined by Kelly [9]) and blocking probability. The implied cost c_k associated with arc $l_k = (v_i, v_j) \in L$ (where $v_i, v_j \in V$, L is the set of arcs of the graph (V,L) defining the network topology and V is the node set where each node represents a switching facility or exchange and each arc or link represents a transmission system) represents the expected number of the increase in calls lost (on all routes of all traffic flows using l_k) as a result of accepting a call of a given traffic flow, on arc l_k. Therefore the bi-objective shortest path problem is:

$$\min\ z^n = \sum_{l_k=(v_i,v_j)\in L} \mathcal{C}_k^n x_{ij} \quad (n=1,2) \tag{1}$$

s.t.

$$\begin{aligned}
&\sum_{v_j\in V} x_{sj} = 1 \\
&\sum_{v_i\in V} x_{ij} - \sum_{v_q\in V} x_{jq} = 0 \quad \forall v_j \in V, (v_j \neq s,t) \\
&\sum_{v_i\in V} x_{it} = 1 \\
&x_{ij} \in \{0,1\}, \quad \forall l_k = (v_i,v_j) \in L \\
&\left(\text{Problem} \quad \mathcal{P}^{(2)}\right),
\end{aligned} \tag{2}$$

where

$$\mathcal{C}_k^1 = c_k \quad \text{and} \quad \mathcal{C}_k^2 = -\log(-B_k).$$

B_k being the call congestion on arc l_k and the log being necessary for obtaining an additive metric.

The multiple objective dynamic routing method proposed in [5] is as a new type of periodic state dependent routing method based on a multiple objective routing paradigm. In its general formulation MODR has the following main features: i) paths are changed dynamically as a function of periodic updates of certain QoS related parameters obtained from real-time measurements, using a multiple objective shortest path model which enables to consider, in an explicit manner, eventually conflicting QoS metrics; ii) it uses a very efficient algorithmic approach, designated as modified multiple objective routing algorithm, prepared to deal with the selection of one alternative path for each node pair in a dynamic alternative routing context (briefly reviewed later in this section) by finding adequate solutions of $\left(\mathcal{P}^{(2)}\right)$; iii) the present version of the method uses estimates of implied costs as one of the metrics to be incorporated in the underlying multiple objective model; iv) it enables to specify required and/or requested values for each metric (associated with predefined QoS criteria), values which define priority regions on the objective functions space. This capability is attached to a routing management system (see [5]) and enables to respond to various network service features and to variable working conditions. As for the way in which the paths are selected in the MODR method, the first path is always the direct route whenever it exists. The remaining routes for traffic flows between an exchange pair are selected from the MMRA, taking into account the defined priority regions.

In general there is no feasible solution which minimises both objective functions of $\left(\mathcal{P}^{(2)}\right)$ simultaneously. Since there is no guarantee of the feasibility of this ideal optimal solution, the resolution of this routing problem aims at finding a best compromise path from the set of non-dominated solutions, according to some relevant criteria defined by the decision maker. Non-dominated solutions can be computed by optimising a scalar function which is a convex combination of the bi-objective functions:

$$\min\ z = \sum_{l_k\in L} \mathcal{C}_k x_{ij} \tag{3}$$

with the same constraints of $\mathcal{P}^{(2)}$ and $\mathcal{C}_k = \sum_{n=1}^{2} \varepsilon_n \mathcal{C}_k^n$, where $\varepsilon = (\varepsilon_1, \varepsilon_2) \in \varepsilon = \{\varepsilon : \varepsilon_n \geq 0,\ n = 1,2 \wedge \sum_{n=1}^{2} \varepsilon_n = 1\}$.
However, by using this form of scalarization only supported dominated paths (that is those which are located on the boundary of the convex hull) may be found. Nevertheless non-dominated solutions located in the interior of the convex hull may exist. MMRA resorts to an extremely efficient *k*-shortest path algorithm [10] to search for this specific type of non-dominated paths.

The basic features of MMRA are the following: i) it enables to search for and select non-dominated or dominated paths for alternative routing purposes; ii) it uses as subalgorithm for calculating *k*-shortest paths a new variant of

the k-shortest path algorithm in [10], developed in [7] for solving the k-shortest path problem with a constraint on the maximum number of arcs per path since this is a typical constraint considered in practical routing methods; iii) the search direction in the objective function space is a 45° straight line; this is justified by the variable nature of the metrics in an integrated service network environment and the possibility of dynamic variation of the priority regions; iv) the priority regions for alternative path selection have a flexible configuration that varies as a result of periodic alterations in the objective function coefficients.

Concerning the specification of the requested and/or acceptable values for the metrics, distinct cases should be envisaged. In the case of blocking probabilities, delays and delay jitter for example, such values can be obtained from network experimentation and/or from ITU-T standardisation or recommendations for various types of networks and services. On the other hand, in the case of costs, namely implied costs, included in the present model, it is more difficult to define a priori such values, since no general criteria are known for these quantities. In the illustrative example described in [5] the requested and acceptable values for z^1 and z^2, were obtained from calculations for the network dimensioned by the classical heuristic [3] for typical network mean blocking probabilities in nominal and overload conditions. The non-dominated and possible a dominated solution corresponding to an alternative path for a given node pair, are selected by MMRA in the higher priority regions. Further details on MMRA and the architecture of MODR method may be seen in [5].

3. Model of network performance

The MODR model described so far, overlooks a question which will be shown to have significant impact on network performance: the interdependencies between implied costs, blocking probabilities and the paths chosen between every node pair. For understanding this and other related problems a model for global network performance evaluation, is now presented.

Denote by: $A_t(f)$ the traffic offered by flow f from node v_i to node v_j at time period t (in Erlangs); $\mathcal{R}_t(f) = \{r^1(f), r^2(f), \ldots, r^M(f)\}$ (in the present model $M = 2$) the ordered set of paths (or routes) which may be used by traffic flow f in time t; $\overline{R}_t = \{R_t(f_1), \ldots, R_t(f_{|\mathcal{F}|})\}$ ($\mathcal{F}$ is the set of all node to node traffic flows); C_k the capacity of link l_k; $R_k = \{r(f) \in R_t(f_1) \cup \cdots \cup R_t(f_{|\mathcal{F}|}) : l_k \in r(f)\}$ the set of routes which, at a given time , may use arc l_k; $\overline{A}_t$ a matrix of elements $A_t(f)$, $f = (v_i, v_j)$; $\overline{C}$ the vector of link capacities C_k; $\overline{B}$ the vector of link call blocking probabilities B_k; $\overline{c}$ the vector of link implied costs c_k and $L_{r^i(f)}$ the blocking probability of route $r^i(f)$. From the definitions and analytic results in [9] and in the previous paper [5] one may obtain a system of implicit equations in B_k and c_k, of the general form:

$$\begin{cases} B_k = \beta_k\left(\overline{B}, \overline{C}, \overline{A_t}, R_k\right) & \text{(S1a)} \\ c_k = \alpha_k\left(\overline{c}, \overline{B}, \overline{C}, \overline{A_t}, R_k\right) & \text{(S1b)} \\ (k = 1, 2, \ldots, |L|) \end{cases}$$

First important elements of the MODR model are a fixed point iterative scheme enabling the numerical computation of $\overline{B}$ and a similar fixed point iterator to calculate $\overline{c}$ given the network topology (V, L), $\overline{C}$, $\overline{A}_t$ and $\overline{R}_t$ (therefore all R_k are also known), which resolve the systems (S1a) and (S1b) respectively, in this order. The convergence of these numerical procedures designated hereafter as fixed point iterators (or simply, iterators) is guaranteed in most cases of practical interest as a consequence of the results in [8, 9]. Taking into account that the algorithm MMRA calculates $\overline{R}_t$ at every period $t = nT$ $(n = 1, 2, \ldots)$, where T is the path updating period, the functional interdependencies between the mathematical entities involved in the MODR may be expressed through:

- $\overline{R}_{t_0} = \overline{R}_0$,
- Recalculate $\overline{c}$, $\overline{B}$ with the iterators for previous $\overline{R}_t$,
- $\overline{R}_t$=MMRA$(\overline{c}, \overline{B})$,

where $\overline{R}_0$, the initial route set should be defined from a suitable network dimensioning method, such as in [3], for given nominal traffic matrix $\overline{A}_{t_0}$.
The next point to be addressed is the definition of the global network performance criteria. The first criterion is the maximisation of the total traffic carried in the network A_c:

$$\max_{\overline{R}_t} A_c = \sum_{f \in \mathcal{F}} A_t(f)\left(1 - B(f)\right), \tag{4}$$

where $B(f)$ is the marginal blocking experienced by traffic flow f in the network at time t:

$$B(f) = L_{r^1(f)} L_{r^2(f)}. \tag{5}$$

The maximisation of A_c is equivalent to the minimisation of the network mean blocking probability:

$$B_m = \sum_{f \in \mathcal{F}} \frac{A_t(f) B(f)}{A_t^0}, \tag{6}$$

where $A_t^0 = \sum_{f \in \mathcal{F}} A_t(f)$ is the total traffic offered; note that (4) is the objective of all "classical" single objective routing methods. The second proposed criterion is the minimisation of the maximal marginal call congestion:

$$\min_{\overline{R}_t} B_M = \max_{f \in \mathcal{F}} \left\{B(f)\right\}. \tag{7}$$

In many situations in alternative routing networks the minimisation of B_m is associated with a penalty on $B(f)$ for "small" traffic flows $A_t(f)$, leading to an increase in B_M. In conventional single objective routing models this effect

is usually limited by imposing upper bounds on $B(f)$. Note that minimising z^1 in $\mathcal{P}^{(2)}$ corresponds to maximising A_c, when searching for a path for flow f only if all the remaining conditions in the network (namely the paths assigned to all other flows and all the link implied costs) were maintained constant, which is not really the case. Similar analysis applies for the minimisation of z^2 in $\mathcal{P}^{(2)}$, concerning the search for the minimisation of B_M. It is therefore important to analyse the effects of the functional interdependencies in terms of global network performance. To illustrate these effects, with respect to z^1 and z^2 separately, and concerning the performance criteria A_c (4) some results are shown in Fig. 1 for a network designated as network B with six nodes, dimensioned according to the method in [3] and described in Appendix. These values in the graphics are the minimum, maximum and average values of A_c obtained for each traffic load factor, by performing $100 \cdot 30$ iterations of minimisation of z^1 (calculation of the shortest path in terms of implied cost) where each iteration corresponds to the calculation of the alternative path for a given node pair. Analogous results are presented in Table 1 for a network designated as network A (given in the Appendix), with the same topology as network B (six node complete graph) but different traffic matrix $\overline{A}_{t_0}$, with link capacities calculated by the same method [3].

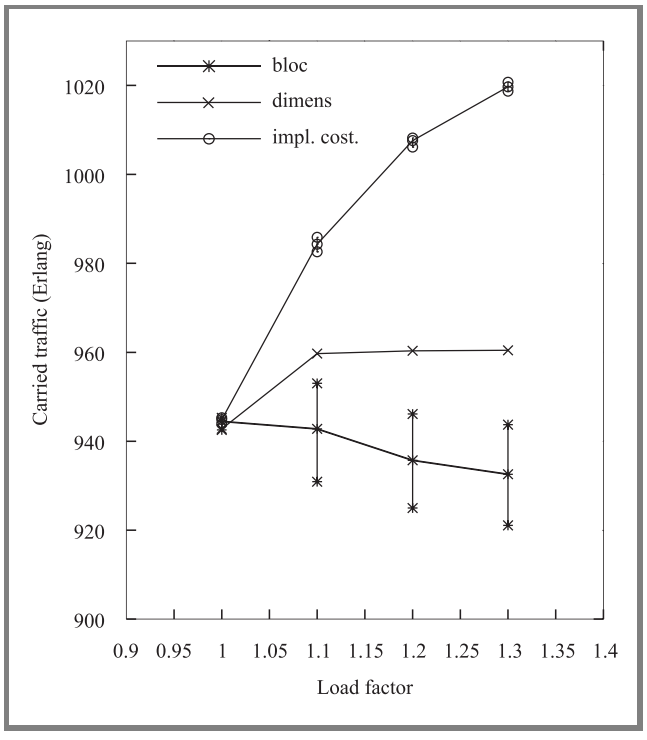


Fig. 1. Oscillations in total carried traffic when z^1 ("impl.cost") and z^2 ("bloc") are minimised separately, for network B.

The following conclusions may be drawn from these results: i) the minimisation of the path implied costs tends to maximise the network carried traffic; ii) there is an instability in the obtained solutions, leading to significant variations in the associated network performance metrics A_c (or B_m) and B_M; iii) the minimisation of the path blocking probabilities leads to relatively small (hence "poor") values of A_c. Analogous conclusions are obtained by calculating paths which minimise z^2 (shortest paths in terms of blocking probability) as illustrated in Table 1, and replacing the network criteria A_c by B_M (maximal node-to-node blocking probability). All these results (similar to those obtained for other networks) are consistent with the assumptions and implications of the model.

Table 1
Variations in B_m and B_M for network A when z^1 ("impl.cost") and z^2 ("bloc") are minimised separately; the $\min B_m$ and $\min B_M$ obtained for the two sets of experiments are indicated in bold

Overload factor [%]	Blocking probability	$\min z^2$ (bloc)		$\min z^1$ (impl.cost.)	
		$\min B_M$	$\min B_m$	$\min B_M$	$\min B_m$
0	B_m	0.00384	**0.00368**	0.00371	0.00369
	B_M	**0.00504**	0.00605	0.00554	0.00613
10	B_m	0.0312	0.0301	0.0306	**0.0297**
	B_M	**0.0372**	0.0442	0.0395	0.0515
20	B_m	0.0946	0.0899	0.0849	**0.0796**
	B_M	**0.109**	0.169	0.140	0.209
30	B_m	0.162	0.149	0.148	**0.128**
	B_M	**0.177**	0.232	0.214	0.258

4. Path instability and network performance

Similarly to the phenomena observed in the previous section for the single objective models based either on implied cost or on blocking probability it could be expected that direct application of MMRA would generate unstable solutions, possibly leading to poor network performance (under the bi-objective approach (A_c, B_M)). In fact direct application of the previous MODR formulation (involving the determination by MMRA of the "best" compromise alternative paths for all origin-destination node pairs as a function of the network state) leads to situations where certain links or paths that were "best" candidates according to the MMRA working, will be in the following path updating period, in a "bad" condition as soon as they are selected as paths of a significant number of O-D pairs. This behaviour leads typically to situations where paths chosen by the routing calculation system may oscillate between a few sets of solutions such that in a certain updating period certain links will be very loaded (i.e. they will contribute to many paths) while others are lightly loaded and in the following period the more loaded and the less loaded links will reverse their condition. This phenomena is a new and specific "bi-

objective" case of the known instability problem in single objective adaptive shortest path routing models of particular importance, for example in packet switched networks (see for example [4], Chap. 5). In our case this behaviour (which may imply inefficiency of the solutions $\overline{R}_t$, from the point of view of global network performance) results from the interdependencies between implied costs, blocking probabilities and paths computed by MMRA and from the discrete nature of the bi-objective shortest path problem.To illustrate these questions Table 2 shows the minimal, maximal and average values of B_m and B_M obtained for network B by executing MMRA 100 times for all node pairs, for each traffic matrix overload factor.

Table 2
Oscillations in B_m and B_M given by MMRA for network B

Overload factor [%]	B_m			B_M		
	Minimum	Maximum	Average	Minimum	Maximum	Average
0	0.00430	0.00748	0.00495	0.00852	0.0510	0.0192
10	0.0814	0.105	0.0925	0.176	0.321	0.243
20	0.160	0.183	0.172	0.274	0.469	0.371
30	0.223	0.250	0.238	0.350	0.599	0.452
40	0.280	0.303	0.292	0.416	0.673	0.504
50	0.327	0.349	0.338	0.444	0.690	0.557

The following conclusions may be drawn from the results: i) there is a significant range of variation in the values of B_m and B_M for each overload factor thereby confirming the instability and potential inefficiency of the solutions; ii) the MMRA solutions correspond in most cases to intermediate values in comparison with the values of $\min B_m$ and $\min B_M$ given by the corresponding shortest path models, as should be expected. Nevertheless in one apparently "odd" case the $\min B_m$ in the table was slightly less than the corresponding value obtained through the minimisation of z^1 in the same number of iterations. This situation although rare in the set of the extensive experimentation performed with the models can be explained by the complexity of the aforementioned functional interdependencies (and the discrete nature of the problem – see Section 3) there is no guarantee that by minimising z^1 (or z^2) any finite number of times, the optimal values of B_m (or B_M) might be obtained.

5. Requirements for a heuristic of synchronous route selection

A heuristic procedure will have to be developed for selecting path sets $\overline{R}_t$ $(t = nT; \quad n = 1,2,\dots)$ capable of overcoming the described path instability problem and guaranteeing a good compromise solution in terms of the two global network performance criteria (B_m, B_M), at every updating period. The foundation of such procedure will be to search for the subset of the alternative path set

$$\overline{R}^a_{t-T} = \{r^2(f), \quad f \in \mathcal{F}\} \tag{8}$$

the elements of which should be possibly changed in the next updating period, seeking to minimise B_m while simultaneously not letting that smaller intensity traffic flows be affected by excessive blocking probability $B(f)$. A first possible criterion for choosing candidate paths for "improvement" was suggested by Kelly [9] for use in an adaptive routing environment: $(1 - L_{r^2(f)})s_{r^2(f)}$. This corresponds to choose paths with a lower value of non-blocking probability multiplied by the corresponding path surplus per call, $s_{r^2(f)}$. Extensive experimentation with the model lead us to propose another criterion for this purpose, depending explicitly both on the first choice path $r^1(f)$ (which in MODR is the direct arc from origin to destination whenever it exists) and on the alternative path $r^2(f)$:

$$\xi(f) = F_1 F_2 = \left(2C^1_{r^1(f)} - C^1_{r^2(f)}\right)\left(1 - L_{r^1(f)} L_{r^2(f)}\right), \tag{9}$$

$$C^1_{r^i(f)} = \sum_{l_k \in r^i(f)} c_k. \tag{10}$$

The objective expressed by the factor F_1 is to favour (with respect to the need to change the 2nd route) the flows for which the 2nd route has a high implied cost and the 1st route a low implied cost. The factor 2 of $C^1_{r^1(f)}$ was introduced for normalising reasons taking into account that $r^1(f)$ has one arc and $r^2(f)$ two arcs, in the considered fully meshed networks. The second factor F_2 expresses the objective of favouring the flows with worse end-to-end blocking probability. The second point to be addressed in the heuristic procedure will be to specify how many and which of the second choice routes $r^2(f)$ with smaller value of $\xi(f)$ should possibly be changed by applying MMRA once again. In any case, among the recalculated routes only those which lead to lower B_m and/or lower B_M should be finally selected by the procedure as routes to be changed in each path updating period. This requires that the effect of each candidate route, in terms of network performance, be previously estimated.

Another mechanism to be introduced in MODR is a specific service protection scheme, aimed at preventing excessive network blocking degradation in overload situations, associated with the utilisation of alternative routes for all node-to-node traffic flows. This mechanism here designated as alternative path removal (APR) is based on the elimination of the alternative paths of all traffic flows for which the value of the scalar function z (3) of the multiple objective model is greater than or equal to a certain parameter z_{APR}. This parameter will have to be carefully "tunned" for each specific network by performing a previous analytical evaluation of the network performance and represents a practical absolute threshold above which the use of alternative routing is no longer justified.

6. Conclusions and ongoing work

A description has been made of new developments of a multiple objective dynamic routing method of periodic state-dependent type for circuit-switched networks, previously presented, aiming to overcome its limitations in terms of global network performance.

A model was presented the numerical resolution of which gives the global network performance measured in terms of total traffic carried and node-to-node blocking probability, when using MMRA and periodically time varying traffic matrices, for one class of service. This model enabled to put in evidence an instability problem in the synchronous path computation module, expressed by the fact that the paths computed by MMRA for all node pairs in each period tend to oscillate between a few sets of solutions many of which lead to a poor global network performance.

Also the essential requirements of a heuristic procedure aiming to overcome this instability problem and obtain acceptable compromise solutions in terms of the global network performance, were presented. Work is progressing with respect to the specification of a heuristic satisfying these requirements and enabling to obtain "good" solutions in terms of the two global network performance criteria B_m (network mean blocking probability) and B_M (maximal node-to-node blocking probability). The performance of the global routing method incorporating this heuristic (MODR-1) is being tested by comparing (for single channel traffic) the obtained global performance network metrics, in three case study networks, with the corresponding results given by a discrete event simulation model for a reference dynamic routing method, real-time network routing (RTNR) developed by AT&T, known for its efficiency and sophistication in terms of service protection mechanisms. Preliminary experiments with the current version of the heuristic, involving the comparison of the analytical results of MODR-1 with simulation results for RTNR for various test networks suggest that MODR-1 might perform better with respect to network mean blocking probability and/or maximum node-to-node blocking probabilities in a very wide variety of network overload conditions. To confirm these results an extensive simulation study with MODR-1 will be carried out for three test networks. Also some modifications are being introduced in the model of periodic recalculation of the boundary values of the priority regions of MMRA which will change dynamically in order to reflect the current loading conditions in the links.

An important conclusion of this work is that a multiple-objective (and indeed a single objective) dynamic routing method where the coefficients of the objective functions of the core multiple objective algorithm depend on the calculated paths (beyond possible intrinsic interdependencies between cost coefficients) have an inherent instability problem which can significantly degrade the "quality" of the obtained solutions in terms of global network performance. This problem, previously overlooked, is a new and specific, "bi-objective case" of the classical instability problem in single objective adaptive routing models, of particular importance, for example, in the case of packet switched networks. This phenomena results from the interdependencies between the calculated paths and the objective functions coefficients and from the discrete nature of the routing problem. To overcome its effects in MODR it is necessary to develop a suitable procedure of heuristic nature enabling to select a final solution at each updating period, with a "good" quality (in terms of the adopted network performance criteria). We think that similar type of heuristics could be applied to different dynamic routing models with similar instability problems.

Further work is also taking place concerning the extension of MODR-1 formulation to multi-service networks, based on appropriate generalisation of the concept of implied cost and appropriate multiclass traffic models, associated with adequate quality of service (traffic dependent) metrics.

Finally the "tuning" of important parameters of the method, namely the path updating period and service protection mechanism parameters, such as z_{APR} in the aforementioned alternative path removal scheme, will have to be tackled through extensive use of a simulation test-bed.

Appendix

Test networks

Calculation results for networks A and B are presented in Tables 3 and 4, respectively.

Table 3
Network A

O-D pair	Link capac.	Offered traf.	Intermediate node
1-2	36	27	3
1-3	13	6	4
1-4	33	25	5
1-5	27	20	6
1-6	31	20	2
2-3	29	25	4
2-4	17	10	5
2-5	37	30	6
2-6	25	20	1
3-4	17	11	5
3-5	14	8	6
3-6	19	13	1
4-5	13	9	6
4-6	27	20	1
6-6	18	12	1

Table 4
Network B

O-D pair	Link capac.	Offered traf.	Intermediate node
1-2	41	27.47	3
1-3	13	6.97	4
1-4	276	257.81	5
1-5	33	20.47	6
1-6	45	29.11	2
2-3	29	25.11	4
2-4	112	101.61	5
2-5	88	76.78	6
2-6	94	82.56	1
3-4	18	11.92	5
3-5	11	6.86	6
3-6	21	13.25	1
4-5	87	79.42	6
4-6	94	83.0	1
6-6	137	127.11	1

Acknowledgements

We thank Tiago Sá for the implementation of the network design model based on the algorithm [3].
This work was partially supported by FCT, project POSI/SRI/37346/2001, "Models and algorithms for treating uncertainty in decision support systems".

References

[1] C. H. Antunes, J. Craveirinha, J. Climaco, and C. Barrico, "A multiple objective routing algorithm for integrated communication networks" in *ITC-16 Teletraffic Engineering in a Competitive World*, P. Key and D. Smith, Eds., Elsevier Science B.V., 1999, vol. 3b, pp. 1291–1300.

[2] G. R. Ash, *Dynamic Routing in Telecommunications Networks*. McGraw-Hill, 1998.

[3] G. R. Ash, R. H. Cardwell, and R. P. Murray, "Design and optimization of networks with dynamic routing", *Bell Syst. Tech. J.*, vol. 60, no. 8, pp. 1787–1820, 1981.

[4] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1992.

[5] J. Craveirinha, L. Martins, T. Gomes, C. H. Antunes, and J. Clímaco, "A new multiple objective dynamic routing method using implied costs", *J. Telecommun. Inform. Technol.*, no. 3, pp. 50–59, 2003.

[6] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*. Addison-Wesley, 1990.

[7] T. Gomes, L. Martins, and J. F. Craveirinha, "An efficient algorithm for calculating *k* shortest paths with a maximum number of arcs", *Invest. Oper.*, vol. 21, pp. 235–244, 2001.

[8] F. P. Kelly, "Blocking probabilities in large circuit-switched networks", *Adv. Appl. Probab.*, vol. 18, pp. 473–505, 1986.

[9] F. P. Kelly, "Routing in circuit-switched networks: optimization, shadow prices and decentralization", *Adv. Appl. Probab.*, vol. 20, pp. 112–144, 1988.

[10] E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. Santos, "Deviation algorithms for ranking shortest paths", *Int. J. Found. Comput. Sci.*, vol. 10, pp. 247–263, 1999.

[11] C. Pornavalai, G. Chakraborty, and N. Shiratori, "Routing with multiple QoS requirements for supporting multimedia applications", *Telecommun. Syst.*, vol. 9, pp. 357–373, 1998.

[12] R. Vogel, R. G. Herrtwich, W. Kalfa, H. Wittig, and L. C. Wolf, "QoS – based routing of multimedia streams in computer networks", *IEEE J. Selec. Areas Commun.*, vol. 14, no. 7, pp. 1235–1244, 1996.

[13] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications", *IEEE J. Selec. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, 1996.

Lúcia Martins, **José Craveirinha**, **Teresa Gomes** – for biography, see this issue, p. 58.

João N. Clímaco – for biography, see this issue, p. 59.

Paper

On the connections between optimal control, regulation and dynamic network routing

Andrzej Karbowski

Abstract — The paper is devoted to studying general features of dynamic network routing problems. It is shown that these problems may be interpreted as receding horizon optimal control problems or simply regulation problems. In the basic formulation it is assumed, that the nodes have no dynamics and the only goal of the optimization mechanism is to find the shortest paths from the source to the destination nodes. In this problem the optimization mechanism (i.e. the Bellman-Ford algorithm) may be interpreted as a receding horizon optimal control routine. Moreover, there is one-to-one correspondence between the Bellman optimal cost-to-go function in the shortest path problem and the Lyapunov function in the regulation problem. At the end some results of the application of the routing optimization algorithm to an inverted pendulum regulation problem are presented.

Keywords — stabilization, nonlinear control, optimal control, dynamic programming, data networks, routing algorithms.

1. General optimal control problem formulation

We consider a deterministic stationary discrete-time, dynamic system described by the state equation:

$$x_{k+1} = f(x_k, u_k), \;\; k = 0,1,2,...,\tau \tag{1}$$

where x_k, u_k, such that

$$x_k \in S \tag{2}$$

$$u_k \in U \tag{3}$$

are, respectively, the state and control vectors, and

$$f : S \times U \to S. \tag{4}$$

By S, U we denoted the subsets of some vector spaces of dimensions n and m, respectively.
For this system we would like to find a closed-loop control strategy

$$\pi = \{\mu_0, \mu_1, \dots, \mu_\tau\}, \tag{5}$$

where $\mu_k(.), k = 0,1,\dots,\tau$, is the kth stage control rule, admissible in the sense of state and control constraints, that is

$$u_k = \mu_k(x_k) \in U, \;\; \forall x_k \in S, \tag{6}$$

that minimizes the cost functional:

$$J(x_0) = \sum_{k=0}^{\tau} g\left(x_k, u_k\right) \tag{7}$$

with respect to both the policy π and the terminal time τ (i.e., the control horizon is free).
Let us select a point $\bar{x}$ from the state space S. We will assume, that for all $x \neq \bar{x}$ and any $u \in U$

$$g(x,u) > 0 \tag{8}$$

and there exists $\bar{u} \in U$ such that:

$$f(\bar{x}, \bar{u}) = \bar{x} \tag{9}$$

with

$$g(\bar{x}, \bar{u}) = 0\,. \tag{10}$$

For instance g may be a quadratic function:

$$\begin{aligned} g(x,u) = (x-\bar{x})'Q(x-\bar{x}) + \\ +(u-\bar{u})'R(u-\bar{u})\,, \end{aligned} \tag{11}$$

where the matrix Q is positive semidefinite and the matrix R is positive definite.
Summing up, we consider an optimal control problem with a fixed terminal state, but free terminal time, defined by

$$\min_{\pi} \left\{ J(x_0) = \sum_{k=0}^{\tau} g\left(x_k, u_k\right) \right\} \tag{12}$$

$$x_{k+1} = f\left(x_k, u_k\right) \tag{13}$$

$$u_k = \mu_k(x_k) \in U \tag{14}$$

$$x_0 = \mathrm{x} \tag{15}$$

$$x_\tau = \bar{x} \tag{16}$$

where $\forall k \;\; x_k \in S$.
We assume, that the system (13)–(15) is controllable to the point $\bar{x}$ from every point of the state space.

2. Analysis

We will apply an analysis method inspired by Luenberger [6].

First, let us notice, that in our problem all functions are time-invariant (stationary). It means, that the solution will not depend on time, either. More precisely, the optimal trajectory from a given state x to the endpoint $\bar{x}$ is independent of the time k_0 at which $x_{k_0} = \mathrm{x}$. That is, if $x_0 = \mathrm{x}$ leads to the optimal trajectory $\{\tilde{x}_k\}$ for $k > 0$ with final time $\tau(\mathrm{x})$, then the condition $x_{k_0} = \mathrm{x}$ must lead to the trajectory $\{\tilde{x}_{k+k_0}\}$ with final time $\tau(\mathrm{x}) + k_0$. The delay of the initial time causes only delaying of the whole solution and the terminal time (i.e., the time of reaching the state $\bar{x}$) is simply an unknown function of the initial state only.

The optimal control rule is also a stationary function, that is for every k

$$u_k = \mu^*\left(x_k\right). \tag{17}$$

It must be so, because the initial control, as we have just stated, depends only on the initial state, not on the initial time, and we can repeat this reasoning at each time instant. Because of the assumptions (8)–(10) there will be:

$$\mu^*\left(\bar{x}\right) = \bar{u}. \tag{18}$$

If $\mu^*(.)$ is the optimal control rule, then we will obtain the following closed-loop system equation:

$$x_{k+1} = f\left(x_k, \mu^*\left(x_k\right)\right). \tag{19}$$

Let us notice, that due to Eqs. (18) and (9) the point $\bar{x}$ is an equilibrium point of the system (19) and according to the construction of the rule $\mu^*(.)$ the system eventually reaches $\bar{x}$. Hence, the system is stable.

Now, let us analyze formally the stability of the system and consider the optimal value function (that is the Bellman function, sometimes called "the optimal cost-to-go") $V_k(x)$ for this problem, expressed as:

$$V_k\left(x_k\right) = \sum_{l=k}^{k+\tau(x_k)} g\left(x_l, \mu^*(x_l)\right), \tag{20}$$

where the function $g(.,.)$ is defined by Eq. (10). This is the optimal (minimal) cost of the passage to $\bar{x}$ at time $k+\tau(x_k)$ when the initial point is x_k with time k. This function satisfies the following conditions:

(i) $V_k(\bar{x}) = 0$

(ii) $V_k(x) > 0$ for $x \neq \bar{x}$

(iii) $V_{k+1}(x_{k+1}) - V_k(x_k) = -g\left(x_k, \mu^*(x_k)\right) < 0$ for $x_k \neq \bar{x}$

Thus V – the Bellman function is a Lyapunov function and we proved the stability of the system.

3. Discrete-state version

In this section we will assume, that the sets S and U are finite and have, respectively, $T+1$ and $V+1$ elements. For the sake of simplicity we denote them by subsequent integers, that is:

$$S = \{0, 1, 2, 3, \ldots, T\} \tag{21}$$

$$U = \{0, 1, 2, 3, \ldots, V\} \tag{22}$$

Consequently we will have:

$$x_k \in S \subset \mathbf{Z}^n \tag{23}$$

$$u_k \in U \subset \mathbf{Z}^m \tag{24}$$

In these circumstances, for any state $x_k = i \in S$, a control $u_k = u \in U$ can be associated with a transition from the state $x_k = i$ to the state $f(i,u) = j \in S$. This passage is characterized by a cost:

$$c_{ij} = \min_{\substack{u \in U \\ f(i,u)=j}} g(i,u) \tag{25}$$

We assumed, that in the case when there are several controls $u \in U$, such that:

$$f(i,u) = j \tag{26}$$

we choose as the passage cost (25) the minimal cost among all costs corresponding to this passage.

Let us define now as a destination state $T \in S$. We will assume that the system may remain in this state, that is

$$\exists u^T \in U \quad f(T, u^T) = T \tag{27}$$

and that the cost of being in this state equals zero, that is:

$$g(T, u^T) = 0. \tag{28}$$

In these conditions the state T is absorbing, that is if the system (1) passes to it, it remains in it for ever.

With this notation, we can interpret our deterministic optimal synthesis problem as a shortest path problem from an initial state 0 to the terminal state T (Fig. 1).

Let us denote now by $N(i)$ the set of all direct neighbours of the node i. The optimized dynamic programming algorithm for this problem will take the form:

$$J(i) = \min_{j \in N(i)} \{c_{ij} + J(j)\} \tag{29}$$

with the terminal condition:

$$J(T) = 0. \tag{30}$$

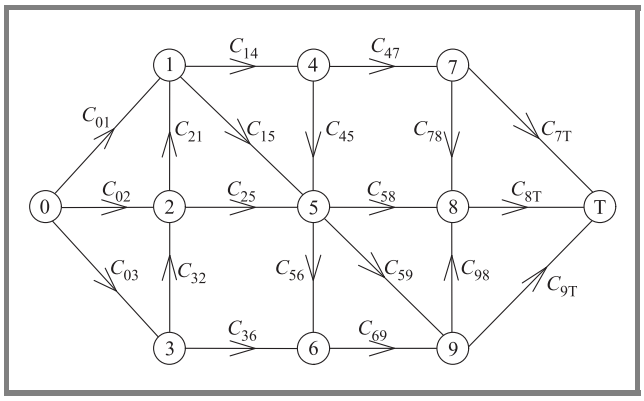


Fig. 1. Graph describing deterministic discrete optimal control problem with terminal state.

4. Routing problem and the Bellman-Ford asynchronous algorithm

The shortest path problem presented above, with a special structure resulting from the original optimal control formulation, may be immersed into a broader class of problems, namely into the class of routing problems. It consists in finding for every node from the set $S = \{0,1,2,\ldots,I\}$ a table defining direct neighbours to which a load or message addressed to some remote node r should be transmitted. A destination may be every node from the set S.
Usually this problem is solved with the help of the asynchronous Bellman-Ford algorithm. This algorithm may be shortly described in the following way [1, 2, 4].
Let us denote the set of all arcs (i,j) between elements of the set S by A. Every arc from A can be characterized by the weight representing its length c_{ij}. The problem is to compute for every node $i \in S$ vectors x_{ir} of shortest distances from this node to the node r. We assume, that every arc in the directed graph $G = (S,A)$ has positive length and that there exists at least one path from every node to others. Then the shortest distances correspond to the unique fixed point of the monotone mapping $F : \mathbf{R}^{I+1} \times \mathbf{R}^{I+1} \to \mathbf{R}^{I+1} \times \mathbf{R}^{I+1}$ defined by $F_{rr}(x) = 0, r \in S$ and

$$F_{ir}(x) = \min_{\{j|(i,j)\in A\}} (c_{ij} + x_{jr}), \;\; i \in S. \tag{31}$$

The Bellman-Ford algorithm consists in the iteration

$$x_{ir} := F_{ir}(x) = \min_{\{j|(i,j)\in A\}} (c_{ij} + x_{jr}), \;\; \forall i, r \in S \tag{32}$$

or in the vector notation:

$$x := F(x) \tag{33}$$

and can be shown to converge to a fixed point

$$x^* = F(x^*) \tag{34}$$

when initialized with $x_{ij} = \infty \;\; \forall i \neq j$.

The convergence takes place also in the case of an asynchronous implementation [1, 4].

5. Integration

Taking into account conclusions drawn from the previous sections, we can write the following:

1. The optimal control policy in the receding horizon control problem for stationary systems with a Lagrange-type performance index is stationary.

2. When the terminal time is free, the optimal closed-loop control problem consists in finding the minimal cost trajectory from any point of the state space to a given point $\bar{x}$.

3. The deterministic closed-loop discrete optimal control problem with a fixed terminal state but with free terminal time (i.e. horizon) can be represented as a shortest path problem.

4. The shortest path may be solved with the help of the Bellman-Ford algorithm designed for routing problems, that might be implemented asynchronously (as in the Internet protocols RIP, IGP or Hello [2]).

Thus, having discretized the problem (12)– (16), connecting all resulting nodes according to the state equation (13) and solving the shortest path problem from all nodes to the node representing the point $\bar{x}$, we can transform the receding horizon optimal control problem into the routing problem and vice-versa.

6. Application of the routing algorithm to the stabilization of an inverted pendulum

To confirm experimentally the equivalence between routing algorithms and the feedback regulation the presented approach was tested on an example taken from [5].
A control law synthesis problem for a simple inverted pendulum was considered. The state variables of this system are the angle ξ and the angular velocity $\dot{\xi}$. The input u is a torque in the shaft, which is bounded to such an amount, that the pendulum cannot directly be turned from the hanging into the upright position. Instead, it is first necessary to "gain enough momentum", which requires a complex trajectory planning, even for this simple system. This nonlinearity posses the main difficulty for the feedback design in this example.
The system is described by the state equations:

$$\dot{x_1}(t) = x_2(t) \tag{35}$$

$$\dot{x_2}(t) = \sin x_1(t) + h(u(t)), \tag{36}$$

where $x_1 = \xi$, $x_2 = \dot{\xi}$ and $h(.)$ is the linear function with saturation, when the module of its argument exceeds 0.7, that is

$$h(u) = \begin{cases} -0.7 & u \le -0.7 \\ u & -0.7 < u < 0.7 \\ 0.7 & u \ge 0.7 \end{cases} \tag{37}$$

An interesting feature of the above system is that a continuous state feedback, which asymptotically stabilizes the system for all initial conditions, does not exist! The reason is, that for any continuous feedback there is a different than origin equilibrium point. More precisely, this point has a nonzero first coordinate. It must be so, because the function

$$f(x_1) = \sin x_1 + h(\mu(x_1, 0)) \tag{38}$$

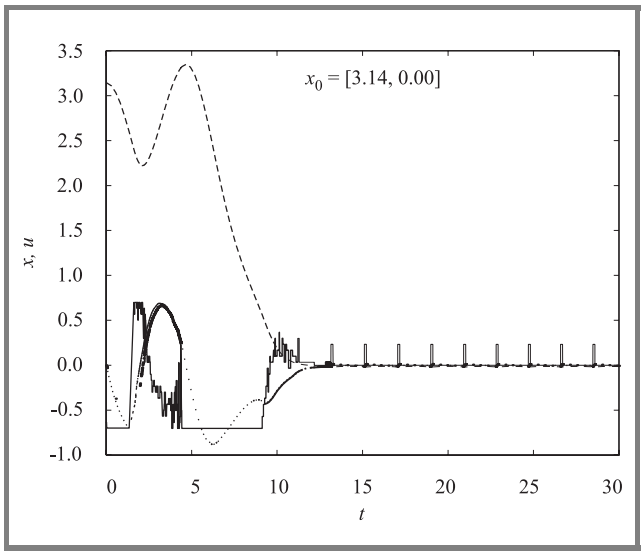


Fig. 2. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[\pi, 0]$ and RB controller.

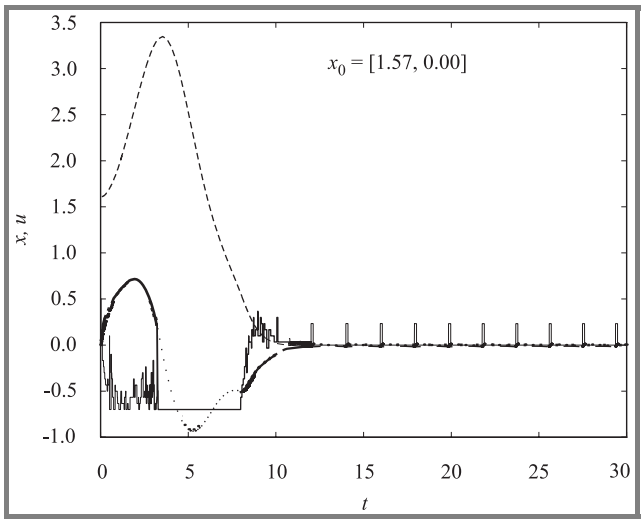


Fig. 3. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[\frac{\pi}{2}, 0]$ and RB controller.

has the positive sign for $x_1 = \pi - arc\sin 0.8$ and the negative sign for $x_1 = \pi + arc\sin 0.8$. It means (from the Darboux theorem) that this function has a root in the interval $[\pi - arc\sin 0.8,\ \pi + arc\sin 0.8]$. In other words, the dynamic system (35)–(36) has an equilibrium point with a zero second and a nonzero first coordinate.

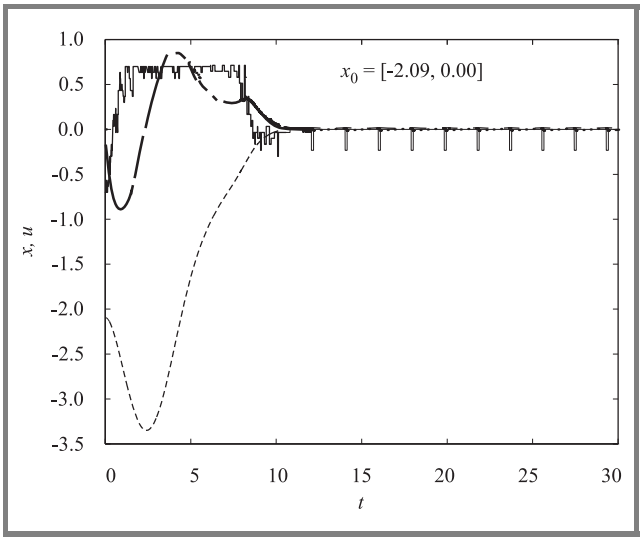


Fig. 4. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[-\frac{2}{3}\pi, 0]$ and RB controller.

The system (35)–(36) was discretized under the following conditions:

- the conversion to the discrete-time representation was obtained via the Euler scheme for a sampling interval $T_s = 0.5$;
- as the state coordinate x_1 space, the interval $[-4, 4]$ was taken; it was discretized into 221 levels;
- as the state coordinate x_2 space, the interval $[-1.6, 1.6]$ was taken; it was discretized into 121 levels;
- the control space (the interval $[-0.7, 0.7]$) was divided into 20 equal subintervals;
- the cost function $g(x(t),\ u(t))$ was assumed to be quadratic, that is

$$g(x,u) = x'Qx + u'Ru \tag{39}$$

with

$$Q = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} \tag{40}$$

and $R = 2$.

It is worth noting, that according to the state equations (35)–(36), for $u = 0$, except of the origin, there are many other equilibrium points, those of coordinates: $[k\pi,\ 0]$, $k = 0, 1, 2, \ldots$. For instance, in the domain, there

are two other (actually it is the same point, where the pendulum is hanging freely) such points.

Several experiments for different initial points were performed. All of them finished in the origin.

The resulting trajectories of the state and control variables are presented in Figs. 2–4. The abbreviation RB means routing based (controller).

For comparison, next figures (Figs. 5–7) present the same trajectories, obtained with the help of LQ methodology, without saturation of the function $h(.)$ (that is, it was replaced by identity). In those experiments, the system (35)–(36) was linearized in the origin, then the optimal static feedback matrix K (that is $u = K \cdot x$) was calculated, with the help of the Matlab Control Toolbox (procedure 'lqr').

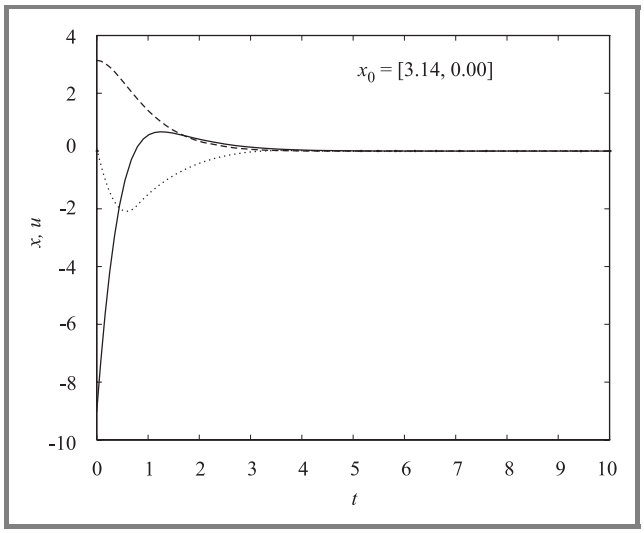


Fig. 5. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[\pi, 0]$ and LQ controller.

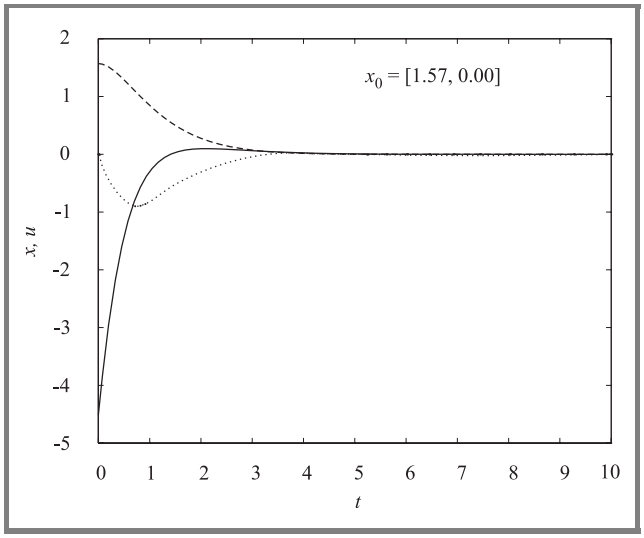


Fig. 6. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[\frac{\pi}{2}, 0]$ and LQ controller.

It is seen, that although in all cases the LQ controller was able to stabilize the pendulum, the control u was very big, out of the admissible interval $[-0.7, 0.7]$ of the previous (RB) case.

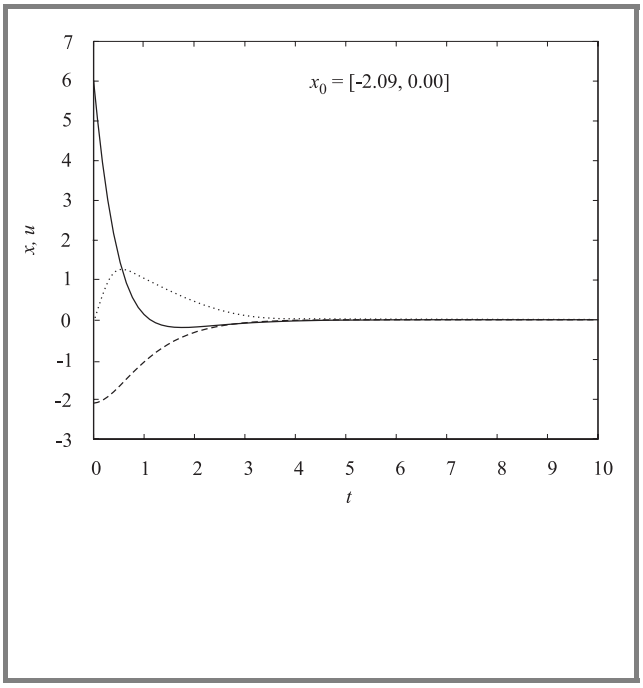


Fig. 7. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for initial condition $[-\frac{2}{3}\pi, 0]$ and LQ controller.

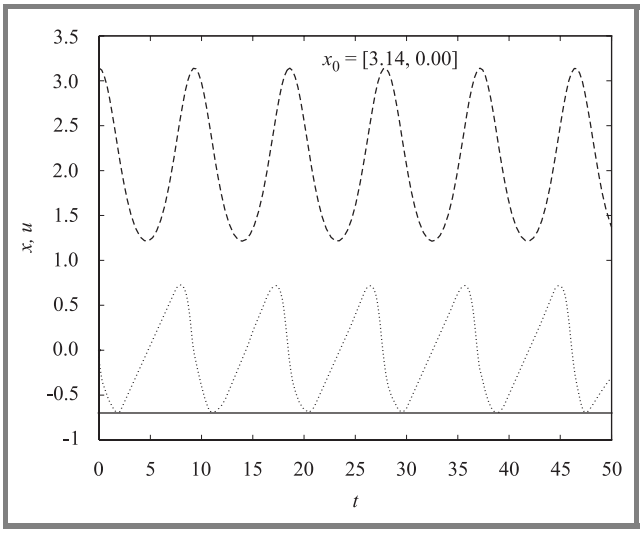


Fig. 8. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for moving pendulum and LQ controller with saturation for initial condition $[\pi, 0]$.

After the series of experiments it turned out, that in the case when the control constraints are taken into account while implementing the LQ control law, even for much greater values of the coefficient R, it is impossible to conduct the pendulum from the free ($[\pi, 0]$) to the upright position (Fig. 8). Let us recall, that it was not a prob-

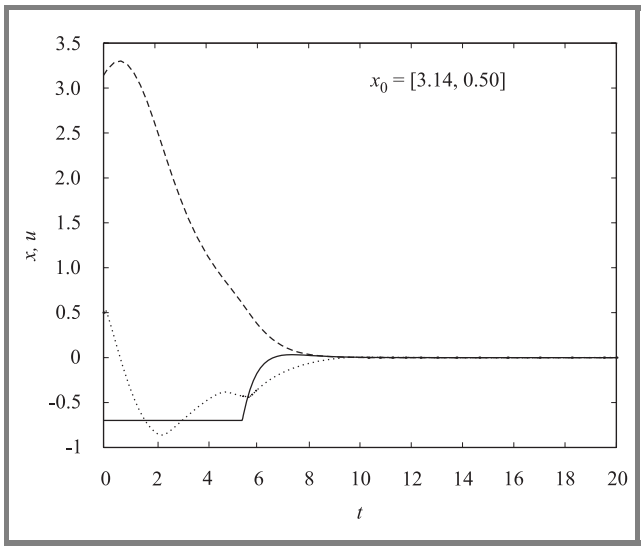


Fig. 9. Trajectories x_1 (---), x_2 ($\cdots$), u (—) for moving pendulum and LQ controller with saturation for initial condition $[\pi, 0.5]$.

lem for RB controller (Fig. 2). However, after giving the pendulum some momentum, the LQ controller with saturation succeeded in regulating the pendulum to this position (Fig. 9).

7. Conclusions

The paper presented connections between a nonlinear stabilization problem and a network routing problem. The may idea lies in the formulation of the original regulation problem as a set of discrete-time receding horizon control problems, solved for all initial states. The optimal control rule may then be calculated (after state discretization) by the application of the Bellman-Ford algorithm, which is an elementary method for calculation of the shortest paths in networks.

An inverted pendulum case of study results showed, that the regulator obtained in this simple way has some advantages over classical LQ approach: it requires much smaller controls to move the state of the system to the equilibrium point neighbourhood, and it can successfully control the system even for initial conditions lying very far from the equilibrium point (that is, it is global). The drawbacks of this regulator are small oscillations around the terminal point, caused by discretization, and the longer time of regulation. Because of that, the best solution in the case of continuous nonlinear systems would be probably a hybrid regulator: discrete – routing based for points lying far from the terminal point and continuous – LQ methodology based, in its neighbourhood.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood-Cliffs: Prentice-Hall, 1992.

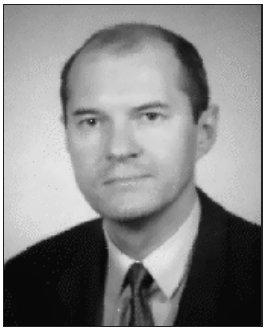
[2] D. E. Comer, *Internetworking with TCP/IP 4/e*. Vol. I : *Principles, Protocols, and Architecture*. Englewood Cliffs: Prentice-Hall, 2000.

[3] A. Isidori, *Nonlinear Control Systems*. London: Springer, 1995.

[4] A. Karbowski, E. Niewiadomska-Szynkiewicz, Red., *Obliczanie równoległe i rozproszone*. Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej, 2001.

[5] G. Kreisselmeier and T. Birkhölzer, "Numerical nonlinear regulator design", *IEEE Trans. Autom. Contr.*, vol. 39, pp. 33–46, 1994.

[6] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models and Applications*. New York: Wiley, 1979.



Andrzej Karbowski received his M.Sc. degree in electronic engineering (specialization automatic control) from Warsaw University of Technology (Faculty of Electronics) in 1983. He received the Ph.D. in 1990 in automatic control and robotics. He works as adjunct both at Research and Academic Computer Network (NASK) and at the Faculty of Electronics and Information Technology (at the Institute of Control and Computation Engineering) of Warsaw University of Technology. His research interests concentrate on data networks management, optimal control in risk conditions, decomposition and parallel implementation of numerical algorithms.

e-mail: A.Karbowski@ia.pw.edu.pl
Research and Academic Computer Network (NASK)
Wąwozowa st 18
02-796 Warsaw, Poland

Paper

Heuristic algorithms in topological design of telecommunication networks

Piotr Karaś

Abstract — The paper addresses the generic topological network design problem and considers the use of various heuristic algorithms for solving the problem. The target of the optimisation is to determine a network structure and demand allocation pattern that would minimise the cost of the network, which is given by fixed installation costs of nodes and links and variable link capacity costs described by linear or concave functions. Input data for the optimisation consists of a list of potential node and link locations and their costs and a set of demands defined between the nodes. Since the problem is known to be NP-hard, the use of specialised heuristic algorithms is proposed. The presented approaches encompass original ideas as well as selected methods described in literature and their enhancements. The algorithms are based on the following ideas and methods: shifting of individual flows, local and global restoration of flows from chosen links or nodes, Yaged algorithm for finding local minima, Minoux greedy algorithm, simulated allocation and genetic algorithms. Efficiency of each of the proposed methods is tested on a set of numerical examples.

Keywords — topological design, network optimisation, heuristic algorithms, genetic algorithms.

1. Introduction

Topological design of telecommunication networks encompasses a range of problems related to localisation of links and nodes of a network. The target of the optimisation is to determine a network structure and demand allocation pattern that would minimise the cost of the network, given a list of potential node locations and a list of admissible interconnections between these nodes. The objective function to be minimised is given as a sum of fixed installation costs of nodes and links and variable link capacity costs (a function of link capacity).

Two subproblems can be distinguished – namely the link localisation problem (LLP), where only link localisation is to be optimised, and the more general transit node and link localisation problem (TNLLP), where localisation of links and transit nodes is subject to optimisation. In literature the LLP is also referred to as the optimal network design problem. This paper addresses both variants, however it concentrates on the more complex TNLLP.

Below the inputs and objectives of the optimisation are described in more detail.

Input data for the optimisation:

- a list of nodes, where access nodes (which originate or terminate demands) and transit nodes (which may transit flows) are distinguished;
- a list of all allowable interconnections between nodes (access links for connecting access nodes to transit nodes and transit links for interconnecting transit nodes);
- demands defined between access nodes, which are to be satisfied;
- link costs – a fixed link installation fee and a variable cost (function of link capacity);
- node costs – a fixed transit node installation fee.

Objectives of the optimisation:

- derive a set of necessary transit nodes;
- derive a set of necessary transit links and access links;
- find an optimal routing of demands in the network;
- minimise the objective function, which is given as a sum of costs of all actually installed links and nodes.

The discussed topological network design problem is generic. It may be interpreted as a task of optimising the topology of a backbone network. Such a design task is a difficult combinatorial optimisation problem and is known to be NP-hard. As the use of exact methods is, in this case, limited to very simple network examples, the heuristic approach has to be considered. Moreover, due to the local optima problem, only more sophisticated methods can prove to be effective.

The paper provides a mathematical formulation of the considered problem and proposes a set of adequate heuristic algorithms. The presented approaches encompass original ideas as well as select methods described in literature and their enhancements.

The proposed algorithms can deal with problems where link cost functions are either linear functions of link capacity with a fixed cost (e.g. link cost may depend on the geographical distance between nodes and a fixed link installation fee) or concave functions of link capacity (reflecting the economy of scale phenomenon common in telecommunication systems).

Numerical examples considered in this paper follow an assumption that the demands are directed and links are undirected (link capacity is taken as a sum of flows in both directions). However all of the algorithms could be easily adapted to solve problems with directed links if required. Similarly sets of transit nodes and access nodes are basically assumed to be disjoint but the algorithms allow for access nodes with transiting capability (mixed functionality).

The paper is organised as follows. Section 2 of the paper provides a mathematical formulation of the considered problem. Next, in Section 3, a range of specialised heuristic algorithms for topological network design is described in detail. Efficiency of each of the proposed methods is tested on a set of numerical examples – the network examples, obtained results and calculation times are presented and discussed in Section 4. Finally, the concluding remarks can be found in Section 5.

2. Problem formulation

Below, a general formulation of the transit node and link localisation problem is given. The link localisation problem, which is a special case of TNLLP, can be obtained by assuming a fixed transit node configuration and setting null transit node installation costs.

The TNLLP formulation presented below uses the link-path notation. The node-link notation is also available (c.f. [1]) and may be more suitable for MIP solvers.

As mentioned above, link cost can be given either by a linear or a concave function of link capacity. In order to obtain a MIP formulation, the $c_e y_e$ term should be used in (1) instead of $f_e(y_e)$, which allows for a nonlinear case.

TNLLP (link-path formulation)

indices

$d = 1, 2, \ldots, D$	demands
$j = 1, 2, \ldots, J_d$	paths for flows realising demand d
$e = 1, 2, \ldots, E$	links
$v = 1, 2, \ldots, V$	transit nodes

constants

h_d	volume of demand d
a_{edj}	1 if link e belongs to path j realising demand d, 0 otherwise
k_e	fixed link e installation cost
$f_e(y_e)$	variable cost of link e (a function of load y_e of link e)
b_{ev}	1 if link e is incident with transit node v, 0 otherwise
l_v	fixed transit node v installation cost
Y_e	upper bound of the capacity of link e (not active)
G_v	upper bound of the degree of transit node v (inactive)

variables

x_{dj}	flow realising demand d allocated to path j (non-negative continuous variable)
y_e	capacity of link e (non-negative continuous variable)
σ_e	1 if link e is provided, 0 otherwise (binary variable)
ε_v	1 if node v is provided, 0 otherwise (binary variable)

objective

$$\min \sum_e \left(f_e(y_e) + k_e \sigma_e \right) + \sum_v l_v \varepsilon_v \tag{1}$$

constraints

$$\sum_j x_{dj} = h_d \quad d = 1, 2, \ldots, D, \tag{2}$$

$$\sum_d \sum_j a_{edj} x_{dj} = y_e \quad e = 1, 2, \ldots, E, \tag{3}$$

$$y_e \leq Y_e \sigma_e \quad e = 1, 2, \ldots, E, \tag{4}$$

$$\sum_e b_{ev} \sigma_e \leq G_v \varepsilon_v \quad v = 1, 2, \ldots, V. \tag{5}$$

Constraints (2) guarantee realisation of demands imposed on the network. Constraints (3) and (4) ensure that zero capacity is assigned to all links that are not provided. Consistent link and node allocation is enforced by constraints 5). In the considered problems parameters Y_e and G_v are assumed to be inactive – they do not limit the capacity of edges and the degree of the nodes, respectively.

Localisation of access nodes is determined by the set of demands imposed on the network. Null installation fee is assumed for these nodes, since it does not have any influence on the produced solutions (other than shifting all of the results by a fixed value).

3. The methods

All of the considered heuristic methods and their modifications are presented below. A short description is provided in each case.

3.1. Flow shifting and rerouting

Most of the proposed flow shifting methods are new implementations of approaches already presented in literature (c.f. [1, 2]), in many places original modifications have been introduced. The flow shifting algorithms are rather simple and serve as basis for some of the more complex methods described in the following subsections.

3.1.1. Individual flow shifting

Individual flow shifting (IFS) was proposed in [1] as a simple heuristic for solving the LLP and TNLLP.

Firstly an initial allocation of demands is performed with a greedy type of algorithm. Subsequent demands are taken

in a random order and allocated to shortest paths. The paths are computed according to incremental cost of routing the demand in question on a given link (in particular if a link is empty its installation cost is added, otherwise only the variable cost is counted).
Secondly, in the main loop of IFS, subsequent demands are checked in a random order and reallocated if their rerouting results in a decrease of the total network cost.
The algorithm stops when the flow rerouting procedure can achieve no further network cost improvement.

3.1.2. Minoux greedy algorithm

The algorithm proposed by Minoux in [2] dealt with the LLP (ONDP). Here the Minoux greedy algorithm (MGA) is applied to the TNLLP.
Firstly initial allocation of demands is done, which may proceed as described for IFS.
Each iteration of the MGA main loop consists in assigning restoration costs δ_e to all of the links carrying some load. The δ_e value for a given link e is calculated as the cost of locally restoring the capacity of the link to an alternative shortest path. The link with the lowest negative restoration cost δ_e is switched off and its flows are redirected.
The algorithm stops when, in a given iteration, all of the links have a positive restoration cost δ_e, meaning that no further improvement of the total network cost can be achieved.

3.1.3. Accelerated Minoux greedy algorithm

The accelerated Minoux greedy algorithm (aMGA) [2] is similar to MGA, however here changes of restoration costs δ_e for subsequent iterations are assumed to be monotonic – hence it is no longer necessary to recalculate all of the δ_e in each iteration. Although the above assumption may not be true for all types of link cost functions, aMGA can be used in all cases and usually provides the same results as MGA while requiring fewer relocation cost computations (which, in turn, makes it significantly faster).

3.1.4. Bulk flow shifting

Initial allocation of demands is performed as described for IFS.
In the main bulk flow shifting (BFS) loop, the algorithm looks through all of the links in a random order. The currently analysed link is switched off and demands crossing it are disconnected. Subsequently an attempt is made to reallocate these demands to new paths. If the reallocation is not feasible or causes an increase of network cost then rollback to the previous configuration is performed.
The algorithm stops when no further improvements are achievable.
In the case of MGA flows from the removed link were rerouted locally and jointly. In case of BFS they are rerouted globally and individually which usually produces a better result. This algorithm differs from the bulk flow shifting approach presented in [1] where the deallocation decision was based on δ_e computations as described for MGA.
Several variants of the algorithm have been analysed. The variant described above is marked as BFS/1. BFS/n stands for a similar method where flows are deallocated on per node basis rather than per link basis. In BFS/b all of the nodes and links are checked in each iteration and the best (not the first better) option is chosen for switch off.
Another option to consider is whether to switch appropriate links/nodes permanently off or rather allow for flow allocation on these links/nodes in the following iterations. These options are indicated as BFS.off and BFS.on, respectively.

3.2. Yaged algorithm

The Yaged algorithm (YAG) proposed in [3] has been used to find locally optimal solutions of network design problems characterised by concave link cost functions over a set of linear flow constraints. The method is known to converge, in a finite number of steps, to local optimum points compliant with the Kühn-Tucker conditions.
Extensions of the algorithm have been presented in [1]. Here the method is applied to the TNLLP.
First an initial allocation of demands is done.
Subsequent iterations of the main YAG loop consist in recalculation of shortest paths for all of the demands with marginal link costs $(\mathrm{d}f_e(y_e)/\mathrm{d}y_e)$ taken as link weights for the Dijkstra algorithm.
The algorithm stops when a fixed point is reached, i.e. two subsequent iterations yield exactly the same flow pattern.

3.3. The notion of adaptive function loop

The adaptive function loop (AFL) approach has been used in [1], combined with various versions of the Yaged algorithm, to deal with design problems characterised by concave link cost functions (requiring global optimisation). Similar cost smoothing techniques were mentioned in [1].
The idea consists in placing any of the simple methods (SM) such as IFS, MGA, aMGA, BFS or YAG in an outer loop called the adaptive function loop. The AFL modifies the link cost function for consecutive SM runs. The idea is to partially linearise the link cost function according to the following formula:

$$C_e(y_e) = \begin{cases} f_e(y_e) & \text{for } y_e \geq y^{th} \\ \beta_e^{th} \cdot y_e & \text{for } y_e < y^{th} \end{cases}$$

y_e – load of link e
$f_e(y_e)$ – original cost function of link e
y^{th} – threshold link load for linearisation
$\beta_e^{th} = f_e(y^{th})/y^{th}$

In subsequent AFL iterations y^{th} value is gradually decreased, hence the first iteration is performed with a linear link cost function (high y^{th} value) and the final one with the original link cost function ($y^{th} = 0$). Solution obtained

by SM in each AFL step serves as starting point for the next iteration (Fig. 1).

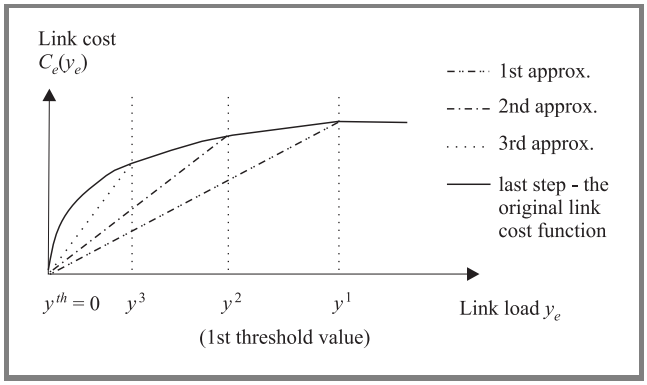


Fig. 1. Cost function linearisation concept in AFL.

The AFL approach has two advantages. Due to the linearisation, the fixed cost is partially included in computations of relocation gain (this tends to give better results). Moreover, by changing the initial y^{th} value and/or the number of adaptive steps n^{adapt}, the algorithm may be directed to various areas of the solution space, thus dealing with the local minimum problem.

Since it is not possible to determine y^{th} and n^{adapt} values that would work equally well for all kinds of problems, performing multiple AFL loop runs for various combinations of these parameters seems to be a good idea. Of course such an approach results in a proportional increase of computation time.

As mentioned above the AFL approach can be combined with various simple methods. The following options may be formed:

Individual flow shifting with adaptive function loop (**IFS-AFL**)

Minoux greedy algorithm with adaptive function loop (**MGA-AFL**)

Accelerated Minoux greedy algorithm with adaptive function loop (**aMGA-AFL**)

Bulk flow shifting with adaptive function loop (**BFS-AFL**)

Yaged algorithm with adaptive function loop (**YAG-AFL**)

3.4. Simulated allocation

The simulated allocation (SAL) algorithm (c.f. [5]) has been already applied to the TNLLP in [1]. Here a simple version of the method along with some enhancements has been implemented for comparison with other methods.

The simulated allocation works with partial allocation states. In each step it decides either to allocate or deallocate a randomly chosen demand (allocation is chosen with a higher probability in order to proceed towards full allocation states). Current value of α factor (defined as fraction of allocated flows) regulates the probability of allocation P_a and deallocation P_d. For example, for $\alpha < 0.8$ use $P_a = 1$ and for $\alpha \geq 0.8$ use $P_a = 0.7$.

Every now and then, when a complete allocation state is reached, a bulk deallocation of demands is performed (e.g. half of all the demands are deallocated). Bulk deallocation enables the algorithm to explore the solution space. Best complete allocation state that is reached is this manner is stored as the final solution.

The algorithm stops if the result is not improved for a predefined number of consecutive full allocation states.

One of the introduced enhancements consists in the way the bulk deallocation is performed. Instead of operation on per flow basis, the procedure deallocates all flows from randomly chosen links or nodes until the requested value of $\alpha = 0.5$ is reached. An on/off approach, similar to the one introduced in BFS.on and BFS.off, was also attempted but did not prove to be effective.

3.5. Genetic algorithm

The proposed genetic algorithm (GA) is based on the $(\mu + \lambda)$ evolution strategy [5]. It involves the use of problem-specific encoding and genetic operators.

The population consists of chromosomes that represent various structures of the network. Each chromosome is built of E binary genes corresponding to each link e of the network. When a link is allocated in a given network structure, the corresponding gene value of the considered chromosome is set to 1. Genes set to 0 indicate not allocated links.

The initialisation procedure builds the initial parent population P_0 by creating μ chromosomes. Half of the P_0 is generated by the greedy initialisation procedure already described for IFS, where shortest paths for the demands are computed in a random order, thus providing diverse solutions. The other half is derived by routing the demands on shortest paths obtained for randomly generated link metrics. In this manner diversity is introduced into P_0.

Inside the main program loop an offspring population O_n consisting of λ chromosomes is formed by copying randomly chosen chromosomes from parent population P_n and mutating them. Mutation consists in introduction of gene variations corresponding to changes in the structure of the network. The following mutation operators may be applied to randomly chosen links and/or nodes: node switch on/off, single link switch on/off and multiple links switch on/off. Each of these actions is performed with a defined probability. After mutation, O_n is evaluated (the demands are routed in the derived network structure and the total network cost is calculated for each chromosome) and the algorithm proceeds to the selection procedure. Here μ best-fitted (i.e. characterised by a minimal network cost) chromosomes are chosen and they become the parent population P_{n+1} for iteration $n+1$.

The loop is exited and the program ends when one of the following two conditions occurs: the algorithm is unable to improve the current best solution for a specified number of

Table 1
Parameters of the tested networks

Network	Number of transit nodes	Number of access nodes	Number of links	Number of demands	min h_d	max h_d	$\sum_d h_d$
N7	7	5	90	42	240	1 920	34 320
N14	14	11	418	182	120	7 560	172 320
N28	28	15	1 050	756	120	30 240	892 492
N9	9	19	132	171	30	150	14 220

generations (so called patience) or it exceeds the maximal allowable number of generations.
The GA version described above is rather simple and there is a lot of room for further enhancements. Implementation of crossover operators could be considered. Explorative properties of the algorithm should be analysed versus its ability to find locally optimal solutions – in this way values of the μ and λ parameters as well as mutation probability could be tuned.

4. Numerical results

Effectiveness of all of the presented methods has been tested on a range of numerical examples of diverse complexity. The algorithms have been implemented in ANSI C and executed on a PC equipped with Athlon 900 MHz processor and 256 MB of RAM.

4.1. The example networks

Three example network structures characterised by different number of nodes have been analysed: N7, N14 and N28 (available from [7]). Link costs were assumed to be linear with a fixed installation cost. Another example network N9 was used with concave link cost functions. The basic parameters of the networks are given in Table 1. All links are potentially available.
Additionally several variants of N7, N14 and N28 were introduced in order to analyse a range of fixed and variable link cost relations (for LLP and TNLLP) as well as the influence of various node installation costs (for TNLLP).
The unit cost c_e of link e is proportional to its geographical length. The fixed installation cost is given by $k_e = c_e \cdot 10^n$. For LLP analysis $n \in \{0,2,3,6\}$, whereas for TNLLP analysis $n \in \{4,5\}$ (and fixed costs of links are additionally multiplied by 3 for access links and by 2 for transit links). For TNLLP, the fixed installation cost of transit nodes is given by $l_v = 10^k$, where $k \in \{4,5,6\}$. There is no installation fee for access nodes.
Results obtained for all of the algorithms, along with the computation times, are summarised in the tables presented in subsections 4.2 and 4.3. In order to draw conclusions on algorithm average performance 10 runs of each of the algorithms have been conducted – each with a different seed value for the randomiser.

Bold font indicates the cases when the optimal solution or the best suboptimal solution has been achieved. Due to the problem complexity, an exact solution (taken from [1]) is available only for the smallest networks and is given in the EX row. It is worth noting that for all of the considered network examples the BFS-AFL algorithm has managed to provide either the optimal solution or the best suboptimal solution.

4.2. TNLLP results

Tables 2–5 refer to the TNLLP problem where link cost function are linear with a fixed installation cost. Table 2 contains the average solutions from 10 runs. The best solutions obtained with each method are noted in Table 3. Table 4 shows the relation between the average solution and the best one known (given by (*average* – *best_known*)/*best_known*). Average calculation times are gathered in Table 5.
BFS-AFL algorithms provided the best average results. BFS/l.on variant proved to be especially effective. SAL.on also performed well as far as the average results are concerned – especially for the simpler networks.
Optimal results or best suboptimal results were also achieved by simple algorithms from the BFS group, however their average results were somewhat worse. Taking into consideration short calculation times for algorithms of this kind, a random approach consisting in performing a number of runs and choosing the best solution could be applied in this case.
Effectiveness of the considered methods is well depicted by the "relative distance from the best" presented in Table 4 and the computation times from Table 5.
The simple algorithms (IFS, MGA, aMGA, BFS and YAG) are characterised by very short computation times, however they produce solutions way below expectations. The only significant exception here is the BFS algorithm group, which requires only slightly more time to provide already acceptable solutions. Especially, the BFS/b.on variant can be considered as a good alternative to more complex and time-consuming algorithms such as SAL and GA.
Introduction of the AFL loop has significantly improved the results provided by most of the simple methods. The AFL loop worked very well with IFS, BFS and YAG. The results for MGA, with or without AFL, are disappointing for TNLLP. For IFS and YAG a huge improvement, as

Table 2
TNLLP: average results (10 runs average) [cost unit 10^6]

Network	N14						N28					
n	4			5			4			5		
k	4	5	6	4	5	6	4	5	6	4	5	6
IFS	80.16	81.15	91.05	342.10	343.09	352.99	298.21	299.56	313.06	904.80	906.15	919.65
MGA	78.39	79.38	89.28	309.94	310.93	320.83	306.36	307.71	321.21	861.10	862.45	875.95
aMGA	78.41	79.40	89.30	311.52	312.51	322.41	306.36	307.71	321.21	861.90	863.25	876.75
BFS/l.off	58.96.	59.84	69.36	271.03	271.98	281.43	270.79	272.07	287.04	742.13	743.46	756.69
BFS/l.on	59.35	60.34	70.24	266.16	267.15	277.05	263.29	264.64	278.14	742.74	744.09	757.59
BFS/n.on	59.07	60.06	69.96	267.56	268.55	278.45	263.02	264.37	277.87	748.91	750.26	763.76
BFS/b.on	58.49	59.48	69.38	265.49	266.48	276.38	258.54	259.89	273.39	735.17	736.52	750.02
YAG	80.16	81.15	91.05	342.10	343.09	352.99	298.21	299.56	313.06	904.80	906.15	919.65
IFS-AFL	61.94	63.10	76.67	334.77	335.76	349.52	254.25	256.63	274.03	774.74	782.42	786.01
MGA-AFL	78.58	79.57	89.47	306.21	307.20	317.10	302.58	303.93	317.43	877.32	878.67	892.17
aMGA-AFL	78.58	79.57	89.47	306.21	307.20	317.10	302.58	303.93	317.43	877.32	878.67	892.17
BFS/l.off-AFL	57.86	58.82	**67.14**	265.81	266.80	276.52	258.16	260.31	273.93	730.33	731.77	745.87
BFS/l.on-AFL	**57.61**	**58.60**	**67.14**	**265.32**	**266.31**	**276.21**	**250.78**	**252.24**	**265.96**	**725.13**	**726.48**	**741.12**
BFS/n.on-AFL	**57.61**	**58.60**	67.60	**265.32**	**266.31**	**276.21**	256.13	257.28	271.99	727.67	728.62	742.70
YAG-AFL	63.28	64.27	74.17	338.63	339.62	349.52	254.92	256.27	269.77	782.58	783.93	797.43
SAL.on	**57.61**	**58.60**	68.50	**265.32**	**266.31**	**276.21**	262.38	263.73	277.23	738.74	740.09	753.59
SAL.off	65.23	65.87	72.06	306.00	306.79	314.67	287.64	288.72	298.37	845.65	846.76	857.70
GA	57.61	58.60	67.67	265.59	266.58	276.78	261.57	263.73	276.04	746.60	747.86	760.04
EX	**57.61**	**58.60**	**67.14**									

Table 3
TNLLP: best results (10 runs data) [cost unit 10^6]

Network	N14						N28					
n	4			5			4			5		
k	4	5	6	4	5	6	4	5	6	4	5	6
IFS	70.29	71.28	81.18	295.74	296.73	306.63	272.39	273.74	287.24	819.12	820.47	833.97
MGA	68.12	69.11	79.01	287.33	288.32	298.22	283.71	285.06	298.56	780.98	782.33	795.83
aMGA	68.12	69.11	79.01	288.30	289.29	299.19	283.71	285.06	298.56	780.98	782.33	795.83
BFS/l.off	**57.61**	**58.60**	**67.14**	**265.32**	**266.31**	**276.21**	261.60	262.86	275.46	733.00	734.26	746.86
BFS/l.on	**57.61**	**58.60**	68.50	**265.32**	**266.31**	**276.21**	255.81	257.16	270.66	729.96	731.31	744.81
BFS/n.on	**57.61**	**58.60**	68.50	**265.32**	**266.31**	**276.21**	259.79	261.14	274.64	733.14	734.49	747.99
BFS/b.on	**57.61**	**58.60**	68.50	**265.32**	**266.31**	**276.21**	254.79	256.14	269.64	726.48	727.83	741.33
YAG	70.29	71.28	81.18	295.74	296.73	306.63	272.39	273.74	287.24	819.12	820.47	833.97
IFS-AFL	59.85	60.02	71.21	304.49	305.48	315.38	252.31	255.60	270.99	747.33	748.68	756.98
MGA-AFL	68.12	69.11	79.01	287.33	288.32	298.22	283.71	285.06	298.56	779.85	781.20	794.70
aMGA-AFL	68.12	69.11	79.01	287.33	288.32	298.22	283.71	285.06	298.56	779.85	781.20	794.70
BFS/l.off-AFL	**57.61**	**58.60**	**67.14**	**265.32**	**266.31**	**276.21**	255.76	258.60	272.98	726.31	727.66	741.16
BFS/l.on-AFL	**57.61**	**58.60**	**67.14**	**265.32**	**266.31**	**276.21**	**250.11**	**251.74**	**265.02**	**723.54**	**724.89**	**738.23**
BFS/n.on-AFL	**57.61**	**58.60**	**67.14**	**265.32**	**266.31**	**276.21**	254.01	255.79	271.13	726.31	727.66	741.16
YAG-AFL	58.93	59.92	69.82	304.49	305.48	315.38	253.87	255.22	268.72	761.03	762.38	775.88
SAL.on	**57.61**	**58.60**	68.50	**265.32**	**266.31**	**276.21**	257.93	259.28	272.78	731.47	732.82	746.32
SAL.off	63.21	63.93	70.69	285.20	286.10	295.10	275.61	276.69	287.49	801.09	802.26	813.96
GA	**57.61**	**58.60**	**67.14**	**265.32**	**266.31**	**276.21**	257.14	261.52	271.70	731.62	732.97	750.79
EX	**57.61**	**58.60**	**67.14**									

Table 4
TNLLP: average – best_known/best_known [%]

Network	N14						N28					
n	4			5			4			5		
k	4	5	6	4	5	6	4	5	6	4	5	6
IFS	39	38	36	29	29	28	19	19	18	25	25	25
MGA	36	35	33	17	17	16	22	22	21	19	19	19
aMGA	36	35	33	17	17	17	22	22	21	19	19	19
BFS/l.off	2	2	3	2	2	2	8	8	8	3	3	3
BFS/l.on	3	3	5	0	0	0	5	5	5	3	3	3
BFS/n.on	3	2	4	1	1	1	5	5	5	4	3	3
BFS/b.on	2	1	3	0	0	0	3	3	3	2	2	2
YAG	39	38	36	29	29	28	19	19	18	25	25	25
IFS-AFL	8	8	14	26	26	27	2	2	3	7	8	6
MGA-AFL	36	36	33	15	15	15	21	21	20	21	21	21
aMGA-AFL	36	36	33	15	15	15	21	21	20	21	21	21
BFS/l.off-AFL	0	0	0	0	0	0	3	3	3	1	1	1
BFS/l.on-AFL	0	0	0	0	0	0	0	0	0	0	0	0
BFS/n.on-AFL	0	0	1	0	0	0	2	2	3	1	1	1
YAG-AFL	10	10	10	28	28	27	2	2	2	8	8	8
SAL.on	0	0	2	0	0	0	5	5	5	2	2	2
SAL.off	13	12	7	15	15	14	15	15	13	17	17	16
GA	0	0	1	0	0	0	5	5	4	3	3	3

Table 5
TNLLP: calculation times (10 runs average) [s]

Network	N14						N28					
n	4			5			4			5		
k	4	5	6	4	5	6	4	5	6	4	5	6
IFS	0	0	0	0	0	0	0	1	1	0	0	1
MGA	0	0	0	0	0	0	0	0	0	1	0	0
aMGA	0	0	0	0	0	0	0	0	1	0	0	0
BFS/l.off	0	1	0	0	0	0	5	5	5	5	5	5
BFS/l.on	0	0	0	1	0	0	5	5	5	4	4	4
BFS/n.on	0	0	0	0	0	0	5	5	5	4	4	4
BFS/b.on	1	1	1	1	1	1	43	43	43	23	22	23
YAG	0	0	0	0	0	0	1	1	1	0	1	1
IFS-AFL	9	9	8	6	6	6	131	134	122	104	104	105
MGA-AFL	1	1	1	1	1	1	4	4	4	5	5	5
aMGA-AFL	1	1	1	1	1	1	4	4	3	4	4	4
BFS/l.off-AFL	41	40	35	41	42	41	708	708	725	736	732	737
BFS/l.on-AFL	45	45	39	42	42	42	762	785	915	760	762	767
BFS/n.on-AFL	48	49	36	37	37	37	956	934	751	952	968	946
YAG-AFL	6	6	6	4	4	4	87	87	87	71	71	71
SAL.on	9	9	9	9	9	9	94	94	94	117	118	117
SAL.off	9	9	9	8	8	8	84	84	92	88	88	88
GA	44	48	80	48	48	48	872	828	951	855	840	831

Table 6
LLP: results (single run) [cost unit 10^6]

Network	N7				N14				N28			
n	0	2	3	6	0	2	3	6	0	2	3	6
IFS	**3.76**	3.95	5.22	1077.71	**26.34**	27.32	33.16	2286.59	151.24	154.81	175.40	5433.15
MGA	**3.76**	**3.94**	5.25	488.57	**26.34**	27.26	32.74	1294.16	151.24	154.74	176.48	2623.15
aMGA	**3.76**	3.95	5.25	488.57	**26.34**	27.33	32.96	1382.16	151.24	154.84	177.31	3281.58
BFS	**3.76**	**3.94**	5.12	503.71	**26.34**	27.12	31.44	1119.19	151.23	154.17	168.58	3035.84
YAG	**3.76**	3.95	5.22	1077.71	**26.34**	27.36	33.37	2358.59	151.24	154.87	176.04	5433.15
IFS-AFL	**3.76**	**3.94**	5.10	**460.69**	**26.34**	27.13	30.58	1261.56	151.24	154.48	167.97	3018.90
MGA-AFL	**3.76**	**3.94**	5.25	488.57	**26.34**	27.26	32.74	1210.20	151.24	154.71	175.88	2458.68
aMGA-AFL	**3.76**	**3.94**	5.25	488.57	**26.34**	27.26	32.74	1138.20	151.24	154.70	176.02	2491.64
BFS-AFL	**3.76**	**3.94**	5.03	**460.69**	**26.34**	27.05	**30.08**	**1062.01**	**151.23**	**153.91**	**165.09**	**2371.12**
YAG-AFL	**3.76**	**3.94**	5.10	548.44	**26.34**	27.36	31.41	1435.27	151.24	154.87	175.57	3114.65
SAL	**3.76**	**3.94**	5.16	**460.69**	**26.34**	27.14	31.47	1172.97	151.23	154.26	169.43	2504.49
EX	**3.76**	**3.94**	**5.00**	**460.69**	**26.34**	**27.02**	**30.08**					

Table 7
LLP: calculation times (single run) [s]

Network	N7				N14				N28			
n	0	2	3	6	0	2	3	6	0	2	3	6
IFS	0	0	0	0	0	0	0	0	1	1	1	1
MGA	0	0	0	0	0	0	0	0	1	1	1	1
aMGA	0	0	0	0	0	0	0	0	1	0	1	0
BFS	0	1	0	0	0	0	0	0	3	4	7	3
YAG	0	0	0	0	0	0	0	0	1	1	0	0
IFS-AFL	1	1	1	1	25	27	27	23	370	383	378	340
MGA-AFL	0	0	0	0	8	10	6	4	129	157	69	51
aMGA-AFL	1	0	1	0	6	4	2	2	39	23	14	11
BFS-AFL	2	2	3	2	52	71	71	71	966	1033	1051	1203
YAG-AFL	0	1	0	0	4	4	5	3	54	55	57	54
SAL	1	0	0	1	4	3	4	6	43	69	55	71

compared to the simple methods, has been achieved while keeping the calculation times pretty low. BFS-AFL approach provided clearly the best results of all the algorithms, being at the same time the most time consuming. Executing the BFS-AFL with fewer AFL iterations, resulted in proportionally reduced calculation times and slightly worse solutions (comparable with methods such as SAL).

As already mentioned in Section 3.3, the algorithms perform multiple AFL runs for different combinations of initial threshold parameter and number of steps parameter. This approach allows for better exploration of the solution space, but results in proportionally longer computations. In this numerical study each of the algorithms performed 40 AFL iterations for TNLLP and 120 for LLP.

The simple implementation of the SAL algorithm also proved to be a very good performer. It provided results at least as good as those cited in [1]. However attempts to further improve the solutions produced by SAL by tuning of the algorithms parameters, even at the cost of significant extensions of the computation time, did not produce any substantial gain.

The performance of GA should be regarded as promising, given the fact that the implemented version of the algorithm requires further tuning and study. However, it should be kept in mind that genetic algorithms are usually quite slow.

4.3. LLP results

The discussed algorithms were tested also on LLP network examples. A single run of all of the methods was performed. The results and computation times are presented in Tables 6 and 7, respectively.

The relative performance of the algorithms as well as their basic characteristics are comparable to those observed for TNLLP and commented in the previous subsection. Also in this case, the BFS-AFL provided the best results. MGA and especially aMGA worked much better with LLP than with TNLLP.

4.4. Concave link cost functions

Network example N9 has been used in order to present the capability of the studied algorithms to deal with problems characterised by concave functions. Link cost function of the following form was assumed (with $A = 0.05$, $B = 300$ and $C = 0.003$):

$$f(y_e) = (-\exp(-Cy_e) + 1) \cdot (Ay_e + B)\,. \qquad (6)$$

The results are summarised in Table 8. The average values are derived from ten runs of each of the algorithms.

Table 8
TNLLP: N9 – concave function example (10 runs data)

Algorithm	Average cost	Best cost	Avg.-best /best [%]	Time [s]
IFS	14733.97	14313.07	5.2	0
MGA	15571.60	14507.29	11.1	0
AMGA	14398.66	14114.22	2.8	0
BFS/l.off	14258.21	14060.45	1.8	0
BFS/l.on	14180.96	14065.07	1.2	1
BFS/n.on	14619.52	**14010.17**	4.3	0
BFS/b.on	14031.60	**14010.17**	0.2	1
YAG	14733.97	14313.07	5.2	0
IFS-AFL	14221.04	14037.11	1.5	5
MGA-AFL	15553.69	14507.29	11.0	1
aMGA-AFL	14368.22	14114.22	2.6	0
BFS/l.off-AFL	14021.05	**14010.17**	0.1	13
BFS/l.on-AFL	14104.12	**14010.17**	0.7	28
BFS/n.on-AFL	**14010.17**	**14010.17**	0.0	26
YAG-AFL	14270.68	14037.11	1.9	2
SAL.on	14046.07	**14010.17**	0.3	8
SAL.off	14082.38	**14010.17**	0.5	8
GA	14055.06	**14010.17**	0.3	33

As expected, all of the algorithms are able to deal with the concave link cost problem. Again the BFS-AFL, SAL and GA are the most effective methods. This time it is the BFS/n.on-AFL variant that provided the best result in all of the iterations. Pretty good performance of IFS-AFL and YAG-AFL can be observed, especially as compared to the simple versions of these methods.
Longest calculation times were experienced for BFS-AFL algorithms and for GA.

5. Conclusions

The generic topological network design problem has been discussed. The link-path formulation of the design task has been presented in order to express the considered problem in a formal manner. Two subproblems, namely the LLP and the TNLLP, have been distinguished. Since exact solution methods are applicable only to trivial networks, adequate heuristic methods capable of providing good suboptimal solutions in reasonable time are called for.
A wide range of specialised heuristic algorithms has been presented. The considered solution methods encompass algorithms known from literature, their modifications and enhancements as well as original ideas. Efficiency of the proposed methods has been demonstrated on a set of numerical examples of diverse complexity. Most of the implemented algorithms proved to be suitable for solution of the considered network examples.
The numerical study enabled a comparison of the methods and selection of the most effective algorithms. The usual time versus quality trade-off has been observed. Simple methods based on the BFS principle were able to provide decent solutions in very short time. Slightly more time consuming versions of BFS provided results comparable to SAL and GA. SAL delivered quite good solutions in a reasonable time. BFS-AFL outperformed all of the other algorithms in terms on solution quality, however required significantly longer computation times. GA implementation was also very time consuming. Generally speaking the AFL notion proved very successful. Depending on the imposed time and result quality requirements, an appropriate algorithm from the discussed range can be selected.
Analysis of relative efficiency or the presented methods gives valuable insight into advantages and drawbacks of various heuristic approaches to solution of the topological design task. Of course the proposed algorithms are not perfect and there is a lot of room for improvements and tuning. Derivation of a good lower bound of the optimal solution would be helpful for algorithm evaluation purposes and/or as a stop criterion. Implementation of more effective procedures, e.g. for shortest paths computation, should be also considered.

References

[1] M. Pióro, A. Jüttner, J. Harmatos, Á. Szentesi, P. Gajowniczek, and A. Mysłek, "Topological design of telecommunication networks. Nodes and links localization under demand constraints" in *Proc. 17th Int. Teletraf. Congr.*, 2001.

[2] M. Minoux, "Network synthesis and optimum network design problems: models, solution methods and application", *Networks*, vol. 19, pp. 313–360, 1989.

[3] B. Yaged Jr., "Minimum cost routing for static network models", *Networks*, vol. 1, pp. 139–172, 1972.

[4] P. Karaś, S. Kozdrowski, and M. Pióro, "Doubly iterative algorithm for multi-layer network design" in *6th PSRT*, Oficyna Wydawnicza Politechniki Wrocławskiej, 1999.

[5] M. Pióro and P. Gajowniczek, "Solving multicommodity integral flow problems by simulated allocation", *Telecommun. Syst.*, vol. 7, no. 1–3, 1997.

[6] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd ed., Springer, 1996.

[7] Network examples can be downloaded from, http://www.tele.pw.edu.pl/networks/TNLLP/

Piotr Karaś received his M.Sc. degree in telecommunication in the Institute of Telecommunications at Warsaw University of Technology in 1999. His M.Sc. thesis concerned concave design problems related to multi-layer telecommunication networks. Currently, he is following Ph.D. studies at Warsaw University of Technology under the supervision of Prof. Michał Pióro. His Ph.D. thesis concerns use of heuristic algorithms in difficult network design tasks. Since 1999 he is working with PTC mobile operator as a specialist in network dimensioning. His research interests include various aspects of network design, core network architecture, traffic analysis and prediction.

pkaras@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Paper

Optimization algorithm for reconfiguration process of the IP over optical networks

Nghia Le Hoang

Abstract — The IP over optical (IPO) network is becoming one of the most interesting among all proposed models of transport networks nowadays. In an IPO network, the reconfiguration capability of the network could be used in order to balance the load of its network elements (NEs). Reconfiguration operations (i.e., switching in OXC nodes and rerouting in IP routers) take place in real-time. Consequently, intensive changes in NEs settings might cause failures in the existing connections in the network. For that reason, changes in NEs settings should be coordinated in a reconfiguration process. In this paper, the author has proposed an optimization method for such reconfiguration process. The mathematical model of the method including computation results has been presented.

Keywords — network reconfiguration, network optimization, IP over optical network.

1. Introduction

IP networks are to transport traffic of various services and with different characteristics [1, 2, 9, 10]. IP traffic, in comparison with traditional telephony traffic, is characterized by greater dynamics and irregularity [13–15]. In order to cope with changing traffic conditions, either the capacity of NEs should be over-provisioned, or NEs should elastically adapt to the changes of traffic conditions. Considering the huge annual growth of IP traffic, the second solution seems to be more desirable. However, the reconfiguration capabilities of NEs are limited. For that reason, changes in NEs settings should be optimally coordinated in a reconfiguration process. A model of such reconfiguration process has been proposed in this paper. The mechanism of the reconfiguration process is based on the optimization method described in the further sections.

1.1. The architecture of an IPO network

An IPO network consists of two layers – the *optical layer* and the *IP layer*. Components of each layer are grouped into three generic classes: nodes, links and paths. In each layer, a link connects two nodes, whilst a path is defined as a sequence of links.

In the optical layer, an *optical node* represents an optical cross connect (OXC). An *optical link* is constructed of one or several parallel optical fiber trails terminated at its nodes. An *optical path* transports one or several optical signals (lambdas). The frequency (wave) of an optical signal should be the same along its path, unless some O/E/O conversions are used on its way. Each optical signal carries a bit-stream. Consequently, an optical path, after E/O-O/E adaptation, is seen by the IP layer as a bundle of electric bit-streams, which has certain rate.

In an IPO network, an IP router is connected to one or several OXCs by means of *inter-layer connectors*. An inter-layer connector could be implemented with an optical fiber or a copper pair. In the first case, the IP router should have E/O-O/E conversion capability. Otherwise, such a capability should be located in the OXC.

In the IP layer, an *IP node* represents an IP router. An *IP link* is constructed of the bit-streams transported by one or several optical paths. These optical paths should start at an optical node connected to an IP node of the IP link, and end at an optical node connected to the other IP node of the IP link. An *IP path* (i.e., a sequence of IP links) transports IP packets from its source node to its destination node.

A simplified illustration of the architecture of an IPO network is shown in Fig. 1.

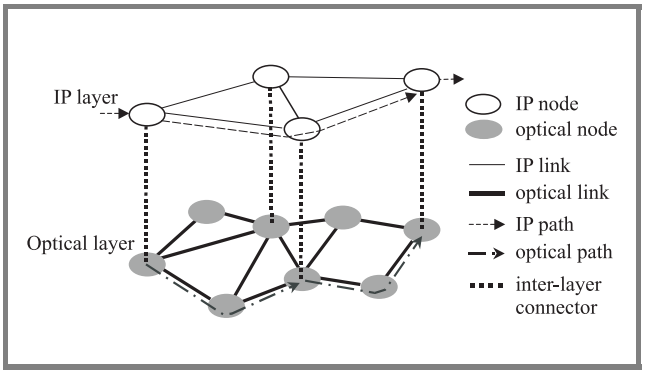


Fig. 1. The architecture of an IPO network.

The services offered by an IPO network are grouped into *service classes*. Between some pair of IP nodes, in each service class, a *demand* is defined in order to represent the offered traffic generated by the users connected to these nodes. The bandwidth required by a demand is called the volume of the demand, and is realized by means of one or several IP paths.

1.2. The description of a reconfiguration process

A *network configuration* corresponds to a feasible combination of NEs settings. The process, in which NEs settings evolve to adapt the network configuration to the traffic conditions, is called the *reconfiguration process* of the network (Fig. 2). The effect of changing traffic conditions is that the network configuration could not be the optimal one in a long time. The consequence of the non-optimal configuration is a low quality of service (QoS) level for connections in the network. In order to prevent such a situation, the reconfiguration process should be active each time when a fall in QoS level is forecasted. Each activation of the reconfiguration process is called a *reconfiguration procedure*. The result of the reconfiguration procedure should be an optimal configuration with regard to the estimated traffic conditions.
Because generally the network configuration cannot be changed to the optimal one at once, the reconfiguration procedure should be decomposed into a sequence of *reconfiguration operations*. Each reconfiguration operation represents an indivisible action that modifies the network configuration.

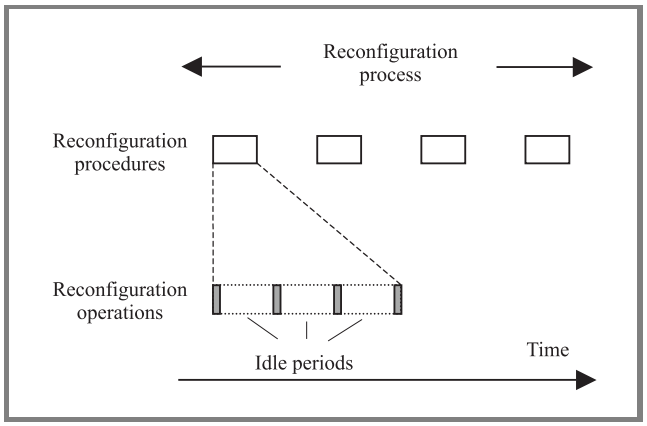


Fig. 2. The structure of a reconfiguration process.

When a reconfiguration operation takes place, some NEs settings are in an unstable state. It is one of the reasons why the next reconfiguration operation should not start immediately after the preceding reconfiguration operation ends. Furthermore, a period is necessery in order to propagate the information about changes of NEs settings throughout the network. Therefore, each reconfiguration operation should be followed by an idle period considerably larger than its duration.
The configuration between two consecutive reconfiguration operations is called an *intermediate configuration*. In a reconfiguration procedure, a series consisting of intermediate configurations and the optimal configuration determines how the network configuration should be changed. In the further part of this paper, such series is called the *configuration series* of a reconfiguration procedure.
The configuration series of a reconfiguration procedure would be computed by means of an optimization algorithm. Such an algorithm has been proposed in this paper. The principle of the algorithm is based on the *optimization problem of reconfiguration* (OPR) that expresses the relations among NEs and the suitability degree of a network configuration to traffic conditions.

2. The simplified description of OPR

The formulation of OPR consists of a group of constraints and an objective function. The constraints of OPR result from the relations among NEs, whilst the objective function of OPR corresponds to the suitability degree of a network configuration to traffic conditions.

2.1. The relations among the NEs – the constraints of OPR

The main constraints of OPR are below:

1) the conservation of flows – the Kirchhoff's law,

2) the budget constraint,

3) the network inertia constraint.

Other possible constraints, e.g. the continuity of an optical wave along its path [5], etc., are neglected in order to simplify the OPR algorithm. It is possible to introduce them into the model in future works.
The result of constraint (1) is that, among the links of a path, the amount of capacity reserved for the path in each link should be equal. Such amount is defined as the capacity of a path. In the IP layer, considering the possibility of packets loss, the Kirchhoff's law should only be a simplifying assumption.
Constraint (2) states that, the total capacity of the paths going across a link (or a node) is restricted by the capacity of the link (or the node).
Constraint (3) is a specific constraint of OPR. Network inertia results from various aspects. Firstly, the switching time in an OXC is finite (~ 50 ms). During the switching time, a stream of 10 Gbit/s rate suffers a loss of 0.5 Gbit. Such loss causes the number of simultaneous switching operations to be limited [3, 8, 16]. Furthermore, a node (an OXC or an IP router), in order to modify its paths, should negotiate with other nodes. The duration time of such procedures is in the order of seconds or dozens of seconds. With a large number of modified paths, commands between nodes have to be queued, and consequently, the unstable period might be minutes or tens of minutes [4].
The full model of network inertia including all its parameters might be quite complicated. In this paper, only the main parameters of network inertia are considered:

– the maximal feasible number of switching operations of a node, within a reconfiguration operation;

– the minimal feasible interval between two consecutive reconfiguration operations.

The network inertia constraint is used in computing an intermediate configuration. In contrary, the optimal configuration is computed by solving the OPR without this constraint.

2.2. The suitability degree of a network configuration to traffic conditions – the objective function of OPR

The suitability degree of a network configuration to traffic conditions is determined with the level of QoS in the network. The level of QoS for connections of a demand depends on the ratio of the bandwidth dedicated for the demand to the volume of the demand. Such ratio is called the *dedicated-requested ratio*. Because network resources are shared for all demands, the level of QoS in the network as a whole should be constructed according to a fairness principle [7, 11, 12]. In this paper, the simplest one has been considered. The level of QoS in the network is defined as the lower bound of the dedicated-requested ratio for each demand.
Because the optimization purpose is to increase the level of QoS in the network, maximization is optimizing the direction. In the basic version, the formulation of the objective function is:

$$u = \min_d \left(a_d / z_d\right),$$

where a_d denotes the bandwidth dedicated for demand d, and z_d denotes the volume of demand d.

2.3. A simplified scenario of a reconfiguration procedure

One of the constraints defined in OPR results from network inertia. Network inertia is the reason why a reconfiguration procedure should be decomposed into reconfiguration operations. In a reconfiguration procedure, a configuration series determines how the network configuration should be changed. Each element of such series could be computed by solving a mixed integer programming (MIP) problem with the objective and the constraints based on OPR.

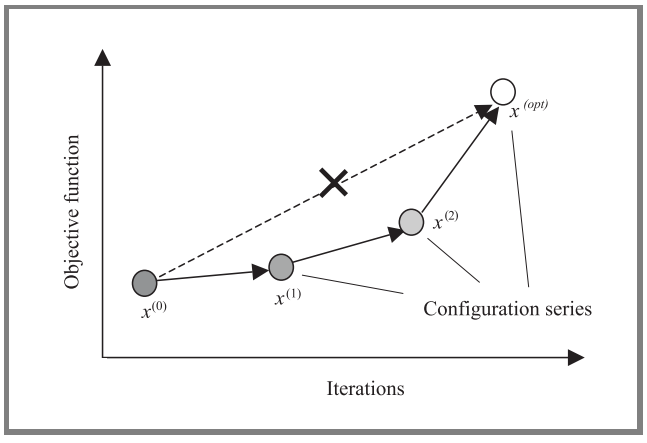


Fig. 3. A reconfiguration procedure.

A configuration series could be computed as follows.
Suppose the current configuration is $x^{(0)}$, whilst the optimal one is $x^{(\mathrm{opt})}$. Let us assume further that $x^{(\mathrm{opt})}$ is not directly reachable from $x^{(0)}$ due to the inertia constraints; while the best reachable configuration (constricted by the network inertia) is $x^{(1)}$ with an objective value much worse than $x^{(\mathrm{opt})}$'s. In this case, we could compute a configuration $x^{(2)}$ modified from $x^{(1)}$ and better than $x^{(1)}$. After a finite number of such iterations, we can achieve the optimal configuration (Fig. 3).
It was only a simplified illustration. Detailed problems (e.g. the shortest way to reach the optimal configuration, the degeneration effect, etc) have been discussed in further sections of this paper.

3. The mathematical model of OPR

3.1. The objects of an IPO network

An IPO network is modeled with generic objects (nodes, links, paths) of both layers and demands. In each configuration, the absence of such object as an optical path, an IP link or an IP path is denoted with a zero capacity of this object. Hence, a creation or a deletion of such object is expressed by a change of capacity from zero to a nonzero value or *vice versa*.
The IP layer is modeled as a directed graph $\boldsymbol{G}_{\mathrm{IP}} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{P})$, where $\boldsymbol{V} = \{v\}_{v=1}^{V}$, $\boldsymbol{E} = \{e\}_{e=1}^{E}$ and $\boldsymbol{P} = \{p\}_{p=1}^{P}$ denote the set of IP nodes, the set of IP links and the set of IP paths, respectively.
Similarly, the optical layer is modeled as a directed graph $\boldsymbol{G}_{\mathrm{opt}} = (\boldsymbol{W}, \boldsymbol{F}, \boldsymbol{Q})$, where $\boldsymbol{W} = \{w\}_{w=1}^{W}$, ,$\boldsymbol{F} = \{f\}_{f=1}^{F}$ and $\boldsymbol{Q} = \{q\}_{q=1}^{Q}$ denote the set of optical nodes, the set of optical links and the set of optical paths, respectively. An optical path serves an IP link by transporting its bit-streams. The set of optical paths that serve IP link $e \in \boldsymbol{E}$ is denoted by $\boldsymbol{Q}(e)$.
Symbols $\boldsymbol{S} = \{s\}_{s=1}^{S}$ and $\boldsymbol{D} = \{d\}_{d=1}^{D}$ denote the set of service classes and the set of demands, respectively. The service class that demand $d \in \boldsymbol{D}$ belongs to is denoted with $s(d)$. The set of IP paths that serve demand $d \in \boldsymbol{D}$ is denoted by $\boldsymbol{P}(d)$.
Each object has some attributes representing e.g. capacity, switching capability etc. The value of such attribute is a scalar quantity. In each object set, each attribute could be denoted with a vector in the $\mathfrak{I}^{+N}$ or $\mathfrak{R}^{+N}$ space (N denotes the number of members of the set). Such characteristic vectors have been described in the next subsections.

3.2. The characteristic vectors of IPO network objects

The capacity of an IP link or an IP path is measured in bit/s. Symbols $b = (b_e)_{e \in E} \in \mathfrak{R}^{+E}$ and $\xi = (\xi_p)_{p \in P} \in \mathfrak{R}^{+P}$ denote the capacity vectors of IP links and IP paths, respectively. The relation between vectors b and ξ, the budget constraint, is given by

$$\forall e \in \boldsymbol{E} : \sum_{p: p \in P \wedge p \ni e} \xi_p \le b_e , \qquad (1)$$

where $p \ni e$ denotes that path p goes across link e.

The capacity of an optical link or an optical path is measured in the number of lambdas. Symbols $c = (c_f)_{f\in F} \in \mathfrak{I}^{+F}$ and $\eta = (\eta_q)_{q\in Q} \in \mathfrak{I}^{+Q}$ denote the capacity vectors of optical links and optical paths, respectively. The relation between vectors c and η, the budget constraint, is presented below:

$$\forall f \in \boldsymbol{F}: \sum_{q:q\in Q\wedge q\ni f} \eta_q \le c_f. \tag{2}$$

In an IPO network, an IP link is constructed by the bitstreams of one or several optical paths. Hence, the value of vector b could be directly derived by the value of vector η by the following formula:

$$\forall e \in \boldsymbol{E}: b_e = C_\lambda \cdot \sum_{q\in Q(e)} \eta_q. \tag{3}$$

Here C_λ is a constant that denotes the bandwidth of the bitstream transported by an optical signal. Besides the link capacity and path capacity, the node capacity should also be considered. The limited capacity of an optical node results from the bounded space in the switching matrix of an OXC and causes a limited number of its input and output ports. In each optical node, the number of input ports and the number of output ports are usually equal. Thus, the capacity vector of optical nodes is denoted consistently with $n = (n_w)_{w\in W} \in \mathfrak{I}^{+W}$. The ports of an OXC are used in connecting the OXC to other OXCs and IP routers. The number of connections in the first type (OXC–OXCs) is equal the total capacity of optical links ended in the OXC. The number of connections in the last type (OXC–IP routers) is equal the total capacity of optical paths ended in the OXC. Consequently, the number of busy ports in an optical node could be accounted by the total capacity of optical links and optical paths ended at the node. In particular, the number of busy input ports of node $w \in W$ is

$$\sum_{f:eg(f)=w} c_f + \sum_{q:ing(q)=w} \eta_q,$$

and its number of busy output ports is

$$\sum_{f:ing(f)=w} c_f + \sum_{q:eg(q)=w} \eta_q.$$

The budget constraint is formulated in this case as follows:

$$\forall w \in \boldsymbol{W}: \sum_{f:eg(f)=w} c_f + \sum_{q:ing(q)=w} \eta_q \le n_w \tag{4}$$

$$\forall w \in \boldsymbol{W}: \sum_{f:ing(f)=w} c_f + \sum_{q:eg(q)=w} \eta_q \le n_w, \tag{5}$$

where $ing(f)$ and $eg(f)$ denote, respectively, the ingress node and the egress node of link f. Similarly, $ing(q)$ and $eg(q)$ denote the ingress node and the egress node, respectively, of path 'q.

The limited capacity of an IP node results from the bounded computational capability of an IP router. The capacity of an IP node $v \in \boldsymbol{V}$ represents the maximal total capacity of IP links ended at the node. The capacity vector of IP nodes is denoted by $m = (m_v)_{v\in V} \in \mathfrak{R}^{+V}$. The budget constraint is formulated in this case as follows:

$$\forall v \in \boldsymbol{V}: \sum_{eg(e)=v} b_e \le m_v \tag{6}$$

$$\forall v \in \boldsymbol{V}: \sum_{ing(e)=v} b_e \le m_v. \tag{7}$$

Vectors m, n and c are parameters that do not depend on NEs settings. By contrast, vectors ξ and η depend on them. On the other hand, the pairs (ξ, η) from different network configurations should not be equal. Hence, a configuration of an IPO network could be defined with a pair (ξ, η). In this paper, the pair (ξ, η) has sometimes been denoted in a convenient form as a mixed vector $x = (\xi, \eta) \in \mathfrak{R}^{+P} \times \mathfrak{I}^{+Q}$.

In an IPO network with known m, n and c, vector $x \in \mathfrak{R}^{+P} \times \mathfrak{I}^{+Q}$ represents a network configuration if and only if it satisfies the conditions from (1) to (7). The set of all such vectors is called the set of feasible configurations, and denoted by $\boldsymbol{K}_{mnc}$.

3.3. The formulation of network inertia

Network inertia causes some configurations not to be directly reachable (i.e., reachable within a reconfiguration operation) from the current configuration. Besides, it prevents reconfiguration operations from taking place too frequently.

Such aspects of the network inertia are represented by following parameters:

- the maximal feasible number of switching operations of a node, within a reconfiguration operation;
- the minimal feasible interval between two consecutive reconfiguration operations.

The switching capability of an IP node is represented by the maximal feasible number of switching operations of the node within a reconfiguration operation. The switching capability vector of IP nodes is denoted with $\alpha = (\alpha_v)_{v\in V} \in \mathfrak{I}^{+V}$. Each switching operation causes a change in the capacity of an IP path. In particular, such change could be a growth from zero or a fall to zero that practically means a creation or a deletion of an IP path. Thus, in each node, the limited switching capability causes the bounded number of changes in the capacity of IP paths that go across the node. Denoting by ξ and ξ' the path capacity vectors of the current and the next configuration, respectively, the following inequality holds:

$$\forall v \in \boldsymbol{V}: \sum_{p:p\ni v} \left|\mathrm{sgn}(\xi'_p - \xi_p)\right| \le \alpha_v, \tag{8}$$

where sgn denotes the sign-num function (sgn(r) returns the sign of real number r). The inequality (8) could be

transformed into a mixed integer form by using a binary indicator $\delta = (\delta_p)_{v \in V} \in \{0,1\}^P$:

$$\forall p \in \boldsymbol{P} : -\delta_p \leq M \cdot \left(\xi'_p - \xi_p\right) \leq \delta_p \tag{9}$$

$$\forall v \in \boldsymbol{V} : \sum_{p: p \ni v} \delta_p \leq \alpha_v . \tag{10}$$

In the inequality (9), M stands for a big constant (e.g. 10^6). Because of the large value of $M, \delta_p = 0$ only if $\xi_p = \xi'_p$, so δ_p could represent the value of $\left|\mathrm{sgn}(\xi'_p - \xi_p)\right|$.
The switching capability of an optical node is represented by the maximal feasible number of switching operations of the node within a reconfiguration operation. The switching capability vector of optical nodes is denoted with $\beta = (\beta_w)_{w \in W} \in \mathfrak{I}^{+W}$. The effect of a switching operation is an increment or a decrement of one optical wave in an optical path. Thus, in the optical node, the limited switching capability causes the bounded sum of capacity change of optical paths that go across the node. Denoting by η and η' the path capacity vectors of the current and the next configuration, respectively, we have the following inequality:

$$\forall w \in \boldsymbol{W} : \sum_{q: q \ni w} \left|\eta_q - \eta'_q\right| \leq \beta_w . \tag{11}$$

The reader could notice some dissimilarity between the inequalities (8) and (11). The dissimilarity results from the different ways to change the path capacity in the layers of an IPO network. A change in the capacity of an optical path requires a number of physical switching operations. The more a capacity changes, the larger number of such operations is needed. In the contrary, the number of operations for a change in the capacity of an IP path generally does not depend on the amount of capacity to be changed. Hence, the number of operations executed by an IP router to modify capacity of some IP paths depends on the number of changed paths (detected by the signnum function), not on the amount of capacity to be changed.

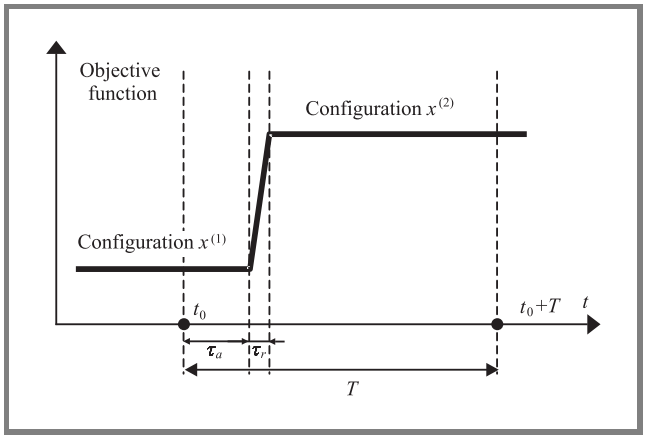


Fig. 4. Time parameters.

In an IPO network of known parameters m, n, c, α and β, a configuration $x' = (\xi', \eta')$ belong to $\boldsymbol{K}_{mnc}$ is directly reachable from the current configuration $x = (\xi, \eta)$ if and only if the inequalities (8) and (11) hold. The set of such configurations is denoted by $\boldsymbol{H}_{mnc\alpha\beta}(x)$, and called the neighborhood of configuration x. In further part of this paper, assuming the parameters m, n, c, α and β to be known and constant, the notations $\boldsymbol{K}_{mnc}$ and $\boldsymbol{H}_{mnc\alpha\beta}(x)$ have been replaced by shortened forms $\boldsymbol{K}$ and $\boldsymbol{H}(x)$.
Another effect of the network inertia is the need of an idle period between consecutive reconfiguration operations. When a reconfiguration operation takes place, some NEs are in an unstable state. Furthermore, a period of time is needed to propagate the information about changes of NEs settings throughout the network. Consequently, each reconfiguration operation should be followed by an idle period considerably larger (e.g. 10^3 times) than its duration. The duration τ_r of a reconfiguration operation is in the order of 100 ms, so interval between reconfiguration operations T should be at least tens of seconds.
Besides the effect of network inertia, another factor delaying a reconfiguration procedure is the duration of traffic analysis and optimization computing. Such duration, denoted by τ_a, should not exceed 10% of T.
The relation among τ_r, τ_a and T is illustrated in Fig. 4. Furthermore, this figure shows how the objective function changes when a reconfiguration operation takes place.

3.4. The demand models

In the preceding subsections, the model of an IPO network has been described. The volume of a demand changes in time and is a quantity independent of the network configuration. Furthermore, it is generally difficult to determine the volume of a demand exactly at each moment. In this paper, two demand models have been proposed:

1) the deterministic model, based on the effective bandwidth;

2) the probabilistic model.

In the first model, the volume of a demand is represented by the effective bandwidth of the demand. The effective bandwidth of a demand is defined as the amount of bandwidth enough to serve the demand with an acceptable QoS. By this way, traffic conditions in the network are represented by the effective bandwidth vector $z = (z_d)_{d \in D} \in \mathfrak{R}^{+D}$.
In the second model, traffic conditions in the network are represented by random vector $Y = (Y_d)_{d \in D}$ of known distribution function $F_Y : \mathfrak{R}^{+D} \to [0 \ldots 1]$. Each element of the random vector Y represents the probabilistic behavior of the volume of a demand.
Because of the dynamics and irregularity of IP traffic, an effective bandwidth defined for a long period (e.g. for all day) causes to be useful. Such a quantity might be too large in comparison with the mean used bandwidth. Similarly, a random vector Y defined for such a long period might have too wide variation. Since the NEs are able to adapt to the changes of traffic conditions, it is not necessary to define z or Y for such a long period. This period should

be chosen in the order of the duration of a reconfiguration process.

3.5. Optimization objective models

The formulation of an optimization objective function depends on the demand model. For the deterministic demand model, the objective function, denoted by $u_1(z,x)$, has the basic formulation as follows:

$$u_1(z,x) = \min_{d \in D}(a_d / z_d)\,. \tag{12}$$

In this formulation, $a_d = \sum_{p:p \in P(d)} \xi_p$ denotes the bandwidth dedicated for demand $d \in D$, which equals the total capacity of the IP paths serving this demand.
The advanced formulation of the objective function, which includes the service class weights, is described below:

$$u_1(z,x) = \min_{d \in D}(\rho_{s(d)} \cdot a_d / z_d)\,. \tag{13}$$

The weights of service classes are denoted by vector $\rho = (\rho_s)_{s \in S} \in \mathfrak{R}^{+S}$.
The objective function for the probabilistic demand model is defined with the help of function u_1. In this case, the value of function $u_1(y,x)$ represents the QoS level of configuration x for an instance y of the random vector Y. Thus, the statistical QoS level of configuration x, which is denoted by $u_2(Y,x)$, could be derived with a Lebesgue's integral:

$$u_2(Y,x) = \int_{\mathfrak{R}^{+D}} u_1(y,x)\,dF_Y(y)\,. \tag{14}$$

Further in this paper, the objective functions $u_1(z,x)$ and $u_2(Y,x)$ have sometimes been denoted by common form $u(x)$ in such situation when the formulations do not depend on the demand model.

4. The optimization algorithm proposed for reconfiguration procedures

Having the model of OPR, we could now determine how a reconfiguration procedure should take place. The configuration series of a reconfiguration procedure should satisfy the following requirements:

- The number of reconfiguration operations in the reconfiguration procedure should be minimized.
- The value of the objective function of a configuration should not be worse than its preceding configuration.
- The last element in the configuration series should be the optimal configuration.

In this paper, three algorithms for determining the configuration series have been proposed. The first exactly obeys the requirements mentioned above. The others are some heuristic ones.

4.1. Algorithm 1 – by defining a master problem

Algorithm 1 is to solve the master problem defined below: Minimize N, subjects to:

$$\forall k = 1 \ldots N : x^{(k)} \in \boldsymbol{H}\left(x^{(k-1)}\right) \tag{15}$$

$$\forall k = 1 \ldots N : u\left(x^{(k)}\right) \geq u\left(x^{(k-1)}\right) \tag{16}$$

$$u\left(x^{(N)}\right) = \max_{x \in K} u(x)\,, \tag{17}$$

where $x^{(0)}$ represents the current configuration; $x^{(1)} \ldots x^{(N-1)}$ the intermediate configurations and $x^{(N)}$ the optimal configuration. Given is $x^{(0)}$, whilst $x^{(1)} \ldots x^{(N)}$ are variables to find.
The procedure for resolving the master problem is following:

1. Let $N \leftarrow 1$.
2. Find $x^{(1)} \ldots x^{(N)}$ satisfying (15)–(17). If there exist such vectors, then **finish**, the series $\left[x^{(1)} \ldots x^{(N)}\right]$ found is the solution we look for.
3. Otherwise, let $N \leftarrow N+1$, and **go to** Step 2.

In each time at Step 2, a MIP problem should be resolved. The size of such MIP problem increases in proportion with N. Therefor, at present; the algorithm could be used only for small networks. In future works, the efficiency of the algorithm could be improved by using a decomposition technique.

4.2. Algorithm 2 – by finding the local optimum

The idea of Algorithm 2 has been described in the simplified illustration in Section 1. This algorithm is based on repeating the local optimization computing in order to reach the global optimal configuration. The description of this algorithm is below:

1. Compute $u_{\text{opt}} = \max_{x \in K} u(x)$.
2. Let $N \leftarrow 1$.
3. Compute such configuration $x^{(N)} \in \boldsymbol{H}\left(x^{(N-1)}\right)$ that $u\left(x^{(N)}\right) = \max_{x \in H\left(x^{(N-1)}\right)} u(x)$. If $u\left(x^{(N)}\right) = u_{\text{opt}}$, then **finish**, the series $\left[x^{(1)} \ldots x^{(N)}\right]$ found is the solution we look for.
4. Otherwise, let $N \leftarrow N+1$, and **go to** Step 3.

Algorithm 2 has a critical drawback. The consequence of the drawback might occur when $u\left(x^{(N)}\right) = u\left(x^{(N-1)}\right)$ in Step 3. In such a situation, we have no guarantee that the next configuration must not be the same as some earlier one. Then, the computation procedure might fall into a perpetual loop, and never reach the optimal configuration. The phenomenon is known as a degeneration effect. In order to repair the drawback, some anti-degeneration techniques should be used, e.g. lexicographic ordering of the configurations.

4.3. Algorithm 3 – by finding the shortest distance to the global optimum

Algorithm 3 is an improved version of Algorithm 2. The description of this algorithm is below:

1. Compute the global optimal configuration $x^{(\mathrm{opt})}$, so $u\left(x^{(\mathrm{opt})}\right) = \max_{x \in K} u(x)$.
2. Let $N \leftarrow 1$.
3. Find such configuration $x^{(N)} \in \boldsymbol{H}\left(x^{(N-1)}\right)$ that $u\left(x^{(N)}\right) = \max_{x \in H\left(x^{(N-1)}\right)} u(x)$ and $\sum_{k=1}^{P+Q} \left|x_k^{(N)} - x_k^{(\mathrm{opt})}\right|$ to be minimized. If $u\left(x^{(N)}\right) = u_{\mathrm{opt}}$, then **finish**, the series $\left[x^{(1)} \ldots x^{(N)}\right]$ found is the solution we look for.
4. Otherwise, let $N \leftarrow N + 1$, and **go to** Step 3.

In comparison with Algorithm 2, in Step 3, the configuration $x^{(N)}$ should not only be the best in the neighborhood of the configuration $x^{(N-1)}$, but should also be the nearest to the optimal configuration $x^{(\mathrm{opt})}$. The introduction of this condition avoids the degeneration effect in most practical cases.

5. Computation examples

The method has been implemented as a C program with the use of the CPLEX optimization package [6]. Two examples have been considered. In Example 1, a small size network is used for illustrating the method. In Example 2, a medium size network is used for testing the convergence of the method. In these examples, the deterministic demand model has been used.

5.1. Example 1 – a small size network for illustrating the method

The topology of the network in this example is shown in Fig. 5. In this figure, an edge (without an arrow) represents a pair of links in both directions between two nodes.

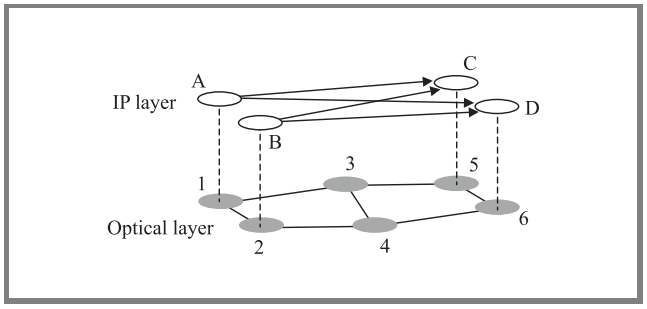


Fig. 5. The topology of the network in Example 1.

The optical layer consists of 6 nodes and 14 links. The capacity of an optical node equals 10 lambdas. The capacity of an optical link equals 5 lambdas. Each lambda carries a bit-stream with 1 Gbit/s. Four optical nodes are connected to IP nodes.

In order to simplify the illustration, only the optical layer is considered, while the IP layer is reduced as much as possible.

The IP layer consists of 4 nodes and 4 links. Each demand is served by only one IP path consisting of only one IP link. The demands are AC, AD, BC and BD. The traffic conditions in the network are represented with effective bandwidth vector *z*.

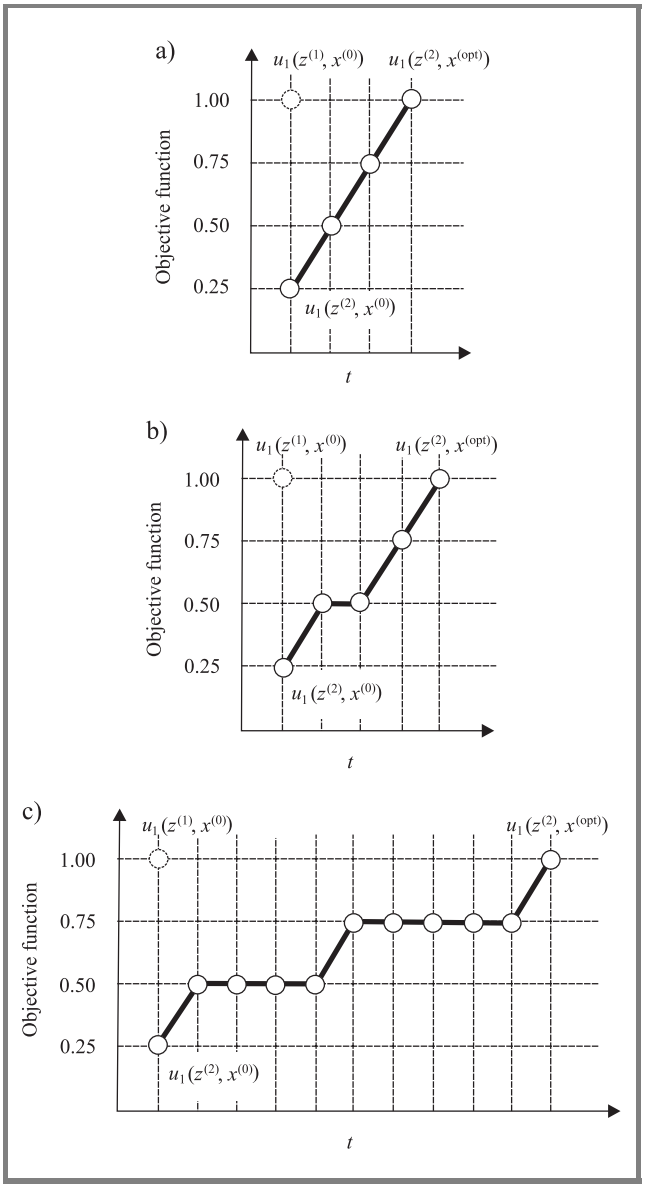


Fig. 6. The configuration series of the reconfiguration procedure in different cases: (a) $\beta_0 = 3$; (b) $\beta_0 = 2$; (c) $\beta_0 = 1$.

Network inertia is represented by the maximal feasible number of switching operations of an optical node within a reconfiguration operation. Such number is denoted by β_0. As regards IP nodes, the network inertia constraint is neglected.

We suppose vector z changes from $z^{(1)} = (1,4,1,4)$ to $z^{(2)} = (4,1,4,1)$. As vector z changes from $z^{(1)}$ to $z^{(2)}$, the objective function falls from 1 to 0.25. We consider three cases: when β_0 equals 3, 2 or 1. Figure 6 shows how the reconfiguration procedure should take place in each case.

When $\beta_0 = 3$, only 3 reconfiguration operations are needed. The objective function successively grows from 0.25 to 1. In such case, both Algorithms 2 and 3 could be used.

When $\beta_0 = 2$, a problem occurs in the second reconfiguration operation: the objective function of $x^{(1)}$ and $x^{(2)}$ are equal. In this case, Algorithm 2 cannot be used because of the degeneration effect. However, Algorithm 3, having an anti-degeneration technique, copes with the problem and succeeds. In this case, 4 reconfiguration operations are needed.

When $\beta_0 = 1$, the network inertia constraint becomes too severe and causes the reconfiguration procedure to take place in 10 reconfiguration operations. Algorithm 3 is the only one that could be used.

In this example, the relation between the reconfiguration capability and the duration of a reconfiguration procedure has been observed. It seems that for a not too severe network inertia constraint, the length of configuration series is acceptable. In the next example, a medium size network is used for testing the convergence of the method.

5.2. Example 2 – a medium size network for testing the convergence of the method

The topology of the network of this example is shown in Fig. 7. In this figure, an edge represents a pair of links in both directions between two nodes.

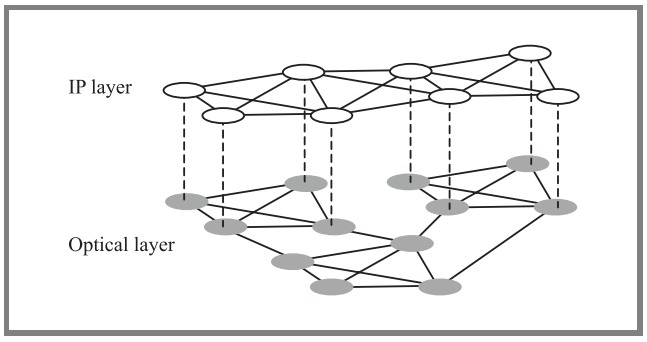


Fig. 7. The topology of the network in Example 2.

The optical layer consists of 12 nodes and 44 links. The capacity of an optical node equals 100 lambdas. The capacity of an optical link equals 10 lambdas. Each lambda carries a bit-stream with 1 Gbit/s. Eight optical nodes are connected to IP nodes. Between each pair of such nodes, because of the enormous number of possible paths, only 4 optical paths are chosen as the admissible paths.

The IP layer consists of 8 nodes and 24 links. The capacity of an IP node equals 100 Gbit/s. Between each pair of IP nodes, 3 IP paths are chosen as admissible paths. The traffic conditions in the network are represented with effective bandwidth vector z. Each element of vector z should be a quantity from 0 to 1 Gbit/s.

Network inertia is represented by the maximal feasible number of switching operations of an optical node within a reconfiguration operation. Such number is chosen to be 4 for each optical node. As regards IP nodes, the network inertia constraint is neglected.

In order to test the method, a random series $\left[z^{(1)} \ldots z^{(K)}\right]$ has been generated. Each pair $\left(z^{(k)}, z^{(k+1)}\right)$ represents a change of traffic conditions in the network. Each time when traffic conditions change, a reconfiguration procedure should take place. In order to determine how the network configuration changes in a reconfiguration procedure, a configuration series is computed. Algorithm 3 has been used.

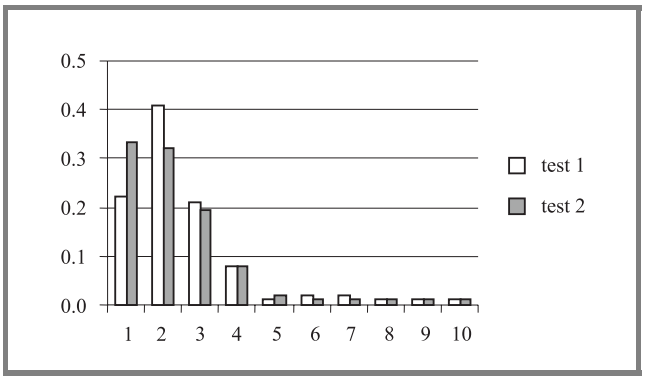


Fig. 8. The test results.

Two tests have been realized. In the first test, the elements of random series $\left[z^{(1)} \ldots z^{(K)}\right]$ are probabilistically independent. In the second test, the elements of random series $\left[z^{(1)} \ldots z^{(K)}\right]$ are correlated by the following rule: for a demand, the probability, that the change of its effective bandwidth is above 50%, is 20%. This rule is denoted as follows:

$$\forall k = 1 \ldots K-1,\ \forall d \in \boldsymbol{D}: \Pr\left\{ \left| z_d^{(k+1)} - z_d^{(k)} \right| \geq 0.5 \cdot z_d^{(k)} \right\} = 0.2.$$

Figure 8 presents the distribution of the number of reconfiguration operations in a reconfiguration procedure. We see that, in this example, the number varies just from 1 to 6.

References

[1] B. T. Doshi, "Future WAN architecture driven by services, traffic volume, and technology trends", *Bell Labs Tech. J.*, vol. 6, no. 1, 2001.

[2] H. Epstein, "Managing optical networks", *Bell Labs Tech. J.*, vol. 6, no. 1, 2001.

[3] D. Banerjee, "Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study", *IEEE/ACM Trans. Netw.*, vol. 7, 2000.

[4] B. T. Doshi, "Optical network design and restoration", *Bell Labs Tech. J.*, vol. 4, no. 1, 1999.

[5] R. Dutta, "A survey of virtual topology design algorithms for wavelength routed optical networks", *Opt. Netw. Mag.*, vol. 10, 2000.

[6] ILOG, "User's Manual of ILOG CPLEX 6.5", 1999.

[7] W. Ogryczak, "Multiple criteria optimization and decisions under risk", *Control Cybern.*, vol. 31, 2002.

[8] J. F. P. Labourdette, "Branch-exchange sequences for reconfiguration of lightwave networks", *IEEE Commun.*, vol. 37, 1994.

[9] B. Rajagopalan, "IP over optical networks: a framework", Internet Draft, draft-ietf-ipo-framework-00.txt

[10] S. Seetharamna, "IP over optical networks: a summary of issues", Internet Draft, draft-osu-ipo-mpls-issue-02.txt

[11] W. Ogryczak and T. Śliwiński, "On solving liner programs with the ordered weighted averaging objective", *Eur. J. Oper. Res.*, vol. 148, pp. 80–91, 2003.

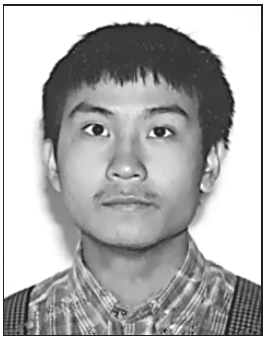
[12] A. Tomaszewski, "A polynomial algorithm for max-min fairness problem", in *Second Polish-German Teletraffic Symp.*, Gdańsk, Poland, 2002.

[13] Z. Papir, *Ruch telekomunikacyjny i przeciążenia sieci pakietowych*. Warszawa: WKŁ, 2001.

[14] A. Adas, "Traffic models in broadband networks", *IEEE Commun. Mag.*, July 1997.

[15] V. S. Frost, "Traffic modeling for telecommunications networks", *IEEE Commun. Mag.*, March 1994.

[16] G. N. Rouskas and M. H. Ammar, "Dynamic reconfiguration in multihop WDM networks", *J. High Speed Netw.*, vol. 4, no. 3, 1995.



Nghia Le Hoang was born in Vietnam in 1974. He received his M.Sc. degree in telecommunications from Warsaw University of Technology, in 1998. He is now preparing his Ph.D. theses, concerning reconfiguration methods for telecommunications networks, at Warsaw University of Technology, in Institute of Telecommunications.

e-mail: nhoang@tele.pw.edu.pl
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Paper

Contextual probability

Hui Wang

Abstract — In this paper we present a new probability function G that generalizes the classical probability function. A *mass function* is an assignment of basic probability to some *context* (events, propositions). It represents the strength of support for some contexts in a domain. A *context* is a subset of the basic elements of interest in a domain – the *frame of discernment*. It is a medium to carry the "probabilistic" knowledge about a domain. The G function is defined in terms of a mass function under various contexts. G is shown to be a probability function satisfying the *axioms of probability*. Therefore G has all the properties attributed to a probability function. If the mass function is obtained from probability function by normalization, then G is shown to be a linear function of probability distribution and a linear function of probability. With this relationship we can estimate probability distribution from probabilistic knowledge carried in some contexts without any model assumption.

Keywords — mathematical foundations, knowledge representation, machine learning, uncertainty, data mining.

1. Introduction

Probability theory is the body of knowledge that enables us to reason formally about uncertain events or propositions. There are different approaches to probability theory, most notably the *frequentist* and *Bayesian* approaches [1, 4].

In the frequentist point of view, the probability of an event is taken to be equal to the limit of the relative frequency of the chosen event with respect to all possible events as the number of trials goes to infinity. The appeal of the frequentist approach for scientists lies in the apparent objectivity of its treatment of data.

On the other hand, the Bayesian approach extends the interpretation of probability to include degrees of belief or knowledge in propositions. We pass from the probability of events (frequentist) to the probability of propositions (Bayesian). Nevertheless the axioms used to define the mathematical properties of probability remain unchanged. Consequently many of the statistical procedures of the two approaches are identical.

Here we focus on the mathematical properties of probability. In particular we take probability to be defined in terms of probability distribution. Let Ω be a set consisting of the basic elements of interest in a domain. A probability distribution function is $p : \Omega \to [0,1]$ such that $\sum_{x\in\Omega} p(x) = 1$. A (classical) probability function is $P : 2^{\Omega} \to [0,1]$ such that, for any $E \subseteq \Omega$

$$P(E) = \sum_{x\in E} p(x). \tag{1}$$

The function $P(E)$ is the probability that an arbitrary element $x \in \Omega$ belongs to a well-defined subset $E \subseteq \Omega$.

It can be shown that the classical probability function defined above satisfies the *axioms of probability*: for any *event* $E \subseteq \Omega$:

- $P(E) \geq 0$.
- $P(\Omega) = 1$.
- If $E_1 \cap E_2 = \emptyset$ then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$.

It is recognized that any function satisfying the axioms of probability, however defined, is a probability function [1].

If we know the probability distribution (in the case of finite sets) or density (in the case of infinite sets) we can calculate probability for any events – in a sense probability distribution provides us with complete information about a domain ([3], p. 273). Therefore probability distribution estimation – estimating the probability distribution from known probabilities for some events – is very important. This is in a sense a way of extending or generalizing our knowledge (represented by probabilities on some events) to all possible events of interest.

There are two general classes of distribution models: parametric and nonparametric. Parametric models assume a particular functional form for the distribution function, such as a uniform distribution, a normal distribution, and so on [3]. Parametric models are often characterized by a relatively small number parameters. Parametric models have the advantage of simplicity (easy to estimate and interpret) but may have relatively high bias because real data may not obey the assumed functional form.

In nonparametric models the distribution estimate is data-driven and relatively few assumptions are made *a priori* about the functional form. Histogram, kernel models and *k*-nearest-neighbors are examples. Histogram is a relatively primitive version of kernel method, and *k-NN* is a special case of kernel [5]. Kernel methods are based on the assumption that a function is constant locally, but the extent of "locality" is parameter to be given, which has a critical bearing on the performance of the methods.

In this paper we present a theory that generalizes the classical probability theory. It can estimate probability distribution without any model assumption.

2. Contextual probability function

Let Ω be a **finite** set called *frame of discernment*. $E \subseteq \Omega$ is called a *context* or *event*. A *mass function* is $m : 2^{\Omega} \to [0,1]$ such that

$$\sum_{X\subseteq\Omega} m(X) = 1. \tag{2}$$

The mass function is interpreted as a *representation of (probabilistic) knowledge* about Ω.

Our objective is to extend our knowledge to those contexts that we have no explicit knowledge about in m. Therefore we define a new function $G: 2^{\Omega} \to [0,1]$ such that for any $E \subseteq \Omega$

$$G(E) = \sum_{X \subseteq \Omega} m(X) \frac{|E \cap X|}{|X|}. \tag{3}$$

The interpretation of the above definition is as follows. Context E may not be known explicitly in the representation of our knowledge, but we know explicitly some contexts X that are related to it (i.e., E overlaps with X or $E \cap X \neq \emptyset$). Part of the knowledge about X ($m(X)$) should then be attributed to E. Since we do not know how this knowledge about X is distributed among the components in X, we can assume it is evenly distributed. So the part of this knowledge attributable to E is $m(X) \times |E \cap X|/|X|$.

Theorem 1. *G is a probability function on Ω. That is to say:*

1. *For any $E \subseteq \Omega$, $G(E) \geq 0$.*
2. *$G(\Omega) = 1$.*
3. *For $E_1, E_2 \in \Omega$, $G(E_1 \cup E_2) = G(E_1) + G(E_2)$ if $E_1 \cap E_2 = \emptyset$.*

Proof. The first claim is true following the fact that $m(X) \geq 0$ for any $X \subseteq \Omega$. The equation holds when $E = \emptyset$.

The second claim is true since $G(\Omega) = \sum_{X \subseteq \Omega} m(X)$.

Let's now consider the third claim. $X \cap (E_1 \cup E_2) = (X \cap E_1) \cup (X \cap E_2)$. If $E_1 \cap E_2 = \emptyset$ then $|X \cap (E_1 \cup E_2)| = |X \cap E_1| + |X \cap E_2|$. As a result we have

$$\begin{aligned} G(E_1 \cup E_2) &= \sum_{X \subseteq \Omega} m(X) \frac{|X \cap (E_1 \cup E_2)|}{|X|} = \\ &= \sum_{X \subseteq \Omega} m(X) \frac{|X \cap E_1| + |X \cap E_2|}{|X|} = \\ &= \sum_{X \subseteq \Omega} m(X) \frac{|X \cap E_1|}{|X|} + \sum_{X \subseteq \Omega} m(X) \frac{|X \cap E_2|}{|X|} = \\ &= G(E_1) + G(E_2). \end{aligned}$$

□

We therefore call G a *contextual probability function*, and the formalism about this function is termed *contextual probability theory* (CPT for short).

As a probability function G has the following properties, the proofs of which are left to the readers.

- $G(\bar{E}) = 1 - G(E)$.
- $G(\emptyset) = 0$.
- If $E_1 \subseteq E_2$, then $G(E_1) \leq G(E_2)$.
- $G(E) \leq 1$, for any context E.
- If $E_1, E_2, \cdots, E_n$ are contexts such that $E_i \cap E_j = \emptyset$ for all pairs i, j, then

$$G(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} G(E_i).$$

- $G(E_1 \cup E_2) = G(E_1) + G(E_2) - G(E_1 \cap E_2)$.

By the above theorem we have

$$1 = G(\Omega) = \sum_{x \in \Omega} G(x).$$

As a result, if G is restricted to the singleton sets (i.e., the elements in Ω) it is a (calculated) probability distribution. This is in contrast to p, which can be interpreted as *a priori* probability distribution.

For simplicity, if E is a singleton set, e.g., $E = \{a\}$, we write $G(a)$ for $G(\{a\})$.

Now we look at an example before we move on.

Example 1. Let $\Omega = \{a,b,c,d,e,f\}$, and the mass function m be as follows:

$$\begin{aligned} m(\{a,b\}) &= 0.3 \\ m(\{a,b,c\}) &= 0.4 \\ m(\{a,b,c,d\}) &= 0.1 \\ m(\{a,b,c,d,e,f\}) &= 0.2 \end{aligned}$$

Suppose that we are interested in the probabilities of the contexts: $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{b,c\}, \{a,b,d\}$. According to the definition of G function, we have

$$\begin{aligned} G(a) &= m(\{a,b\}) \times \frac{|\{a\}|}{|\{a,b\}|} + m(\{a,b,c\}) \times \frac{|\{a\}|}{|\{a,b,c\}|} + \\ &\quad + m(\{a,b,c,d\}) \times \frac{|\{a\}|}{|\{a,b,c,d\}|} + \\ &\quad + m(\{a,b,c,d,e,f\}) \times \frac{|\{a\}|}{|\{a,b,c,d,e,f\}|} = \\ &= 0.3 \times 1/2 + 0.4 \times 1/3 + 0.1 \times 1/4 + 0.2 \times 1/6 = \\ &= 41/120 \end{aligned}$$

$$G(b) = G(a)$$

$$\begin{aligned} G(c) &= m(\{a,b,c\}) \times \frac{|\{c\}|}{|\{a,b,c\}|} + \\ &\quad + m(\{a,b,c,d\}) \times \frac{|\{c\}|}{|\{a,b,c,d\}|} + \\ &\quad + m(\{a,b,c,d,e,f\}) \times \frac{|\{c\}|}{|\{a,b,c,d,e,f\}|} = \\ &= 0.4 \times 1/3 + 0.1 \times 1/4 + 0.2 \times 1/6 = \\ &= 23/120 \end{aligned}$$

$$\begin{aligned} G(d) &= m(\{a,b,c,d\}) \times \frac{|\{d\}|}{|\{a,b,c,d\}|} + \\ &\quad + m(\{a,b,c,d,e,f\}) \times \frac{|\{d\}|}{|\{a,b,c,d,e,f\}|} = \\ &= 0.1 \times 1/4 + 0.2 \times 1/6 = 7/120 \end{aligned}$$

$$G(e) = m(\{a,b,c,d,e,f\}) \times \frac{|\{e\}|}{|\{a,b,c,d,e,f\}|} =$$
$$= 0.2 \times 1/6 = 4/120$$
$$G(f) = G(e).$$

Clearly $G(a)+G(b)+G(c)+G(d)+G(e)+G(f)=1$. Further on, we have

$$G(\{b,c\}) = m(\{a,b\}) \times \frac{|\{b\}|}{|\{a,b\}|} +$$
$$+ m(\{a,b,c\}) \times \frac{|\{b,c\}|}{|\{a,b,c\}|} +$$
$$+ m(\{a,b,c,d\}) \times \frac{|\{b,c\}|}{|\{a,b,c,d\}|} +$$
$$+ m(\{a,b,c,d,e,f\}) \times \frac{|\{b,c\}|}{|\{a,b,c,d,e,f\}|} =$$
$$= 0.3 \times 1/2 + 0.4 \times 2/3 + 0.1 \times 2/4 + 0.2 \times 2/6 =$$
$$= 64/120 = G(b) + G(c)$$
$$G(\{a,b,d\}) = m(\{a,b\}) \times \frac{|\{a,b\}|}{|\{a,b\}|} +$$
$$+ m(\{a,b,c\}) \times \frac{|\{a,b\}|}{|\{a,b,c\}|} +$$
$$+ m(\{a,b,c,d\}) \times \frac{|\{a,b,d\}|}{|\{a,b,c,d\}|} +$$
$$+ m(\{a,b,c,d,e,f\}) \times \frac{|\{a,b,d\}|}{|\{a,b,c,d,e,f\}|} =$$
$$= 0.3 + 0.4 \times 2/3 + 0.1 \times 3/4 + 0.2 \times 3/6 =$$
$$= 89/120 = G(a) + G(b) + G(d).$$

3. CPT versus probability theory and Dempster-Shafer theory

Contextual probability theory generalizes classical probability theory in the sense that the probability distribution p changes to the mass function m and the probability function P changes to contextual probability function G. The probability distribution p is defined on Ω while m is defined on 2^Ω; P and G are both defined on 2^Ω and they are both probability functions. Once the mass function is restricted to singletons the G function becomes the probability function.

Dempster-Shafer (D-S for short) theory [6] is also a generalization of probability theory, which has evolved from a theory of upper and lower probabilities. It starts by assuming a set Ω and a *mass function* m, based on which the *belief* function *bel* and *plausibility* function *pls* are defined. Formally the mass function is $m : 2^\Omega \to [0,1]$ where $m(\emptyset) = 0$ and $\sum_{X \subseteq \Omega} m(X) = 1$. Belief function is $bel : 2^\Omega \to [0,1]$ such that, for $E \subseteq \Omega$, $bel(E) = \sum_{X \subseteq \Omega, X \subseteq E} m(X)$. Plausibility function is $pls : 2^\Omega \to [0,1]$ such that $pls(E) = 1 - bel(E')$, where E' is the complement of E in Ω.

The set Ω is a set of mutually exclusive alternatives. For any $E \subseteq \Omega$, $m(E)$ represents the strength of some evidence supporting E; $bel(E)$ summarizes all reasons to believe E, and $pls(E)$ expresses how much we should believe in E if all currently unknown facts were to support E. Thus the true belief in E will be somewhere in the interval $[bel(E), pls(E)]$.

When the mass function is restricted to singleton elements $x \in \Omega$, the belief and plausibility functions become the same and they are also the same as the probability function. Therefore D-S theory is regarded as a generalization of probability theory [2].

However the belief function satisfies the first two axioms of probability theory, but for the third axiom the equation is changed to "$\geq$" [2]. Therefore the belief function is not probability function.

Although both CPT and D-S theory can be understood as generalizations of probability theory, there are differences between the two:

- CPT uses a single function to represent "uncertainty" while D-S theory uses two functions.
- The G function is a probability function, therefore all of the properties of probability theory are still valid. For example, with the additive property we do not need to calculate G for every $E \subseteq \Omega$; instead we only need to do so for singletons $x \in \Omega$ and $G(E) = \sum_{x \in E} G(x)$. The belief function is, however, not a probability function. So we have to calculate *bel* for every $E \subseteq \Omega$.

4. Relationship between G and P

Now that G is a probability function, we may ask the question: what is the relationship between G and P? To answer this question we need to base m on p so that G can be connected to P. Here we interpret mass function as a measure of the occurrence of elements in a set. Therefore the larger a set is the more likely the set as an event occurs.

Specifically we assume that the mass function be defined in terms of probability as follows, letting $K \stackrel{\text{def}}{=} \sum_{X \subseteq \Omega} P(X)$

$$m(E) = \frac{P(E)}{\sum_{X \subseteq \Omega} P(X)} = \frac{P(E)}{K}. \tag{4}$$

According to this interpretation of the mass function the following lemma follows from the fact that $P(E_1) \leq P(E_2)$ when $E_1 \subseteq E_2$.

Lemma 1. *If* $E_1 \subseteq E_2$ *then* $m(E_1) \leq m(E_2)$.

Let $\binom{N}{n}$ be the combinatorial number representing the number of ways of picking n unordered outcomes from N possibilities. From combinatorics we know that $\binom{N}{n} = \frac{N!}{(N-n)!n!}$. With these assumptions we have the following results.

Lemma 2. *Let* $N = |\Omega|$. *Then* $K = \sum_{i=1}^{N} \binom{N-1}{i-1} = 2^{N-1}$.

Proof.

$$K = \sum_{X\subseteq\Omega} P(X) = \sum_{i=1}^{N} \sum_{X\subseteq\Omega,|X|=i} P(X) =$$
$$= \sum_{i=1}^{N} \sum_{x\in\Omega} \binom{N-1}{i-1} p(x) = \sum_{i=1}^{N} \binom{N-1}{i-1} \sum_{x\in\Omega} p(x) =$$
$$= \sum_{i=1}^{N} \binom{N-1}{i-1} = 2^{N-1}.$$

□

Theorem 2. *Let* $\alpha \stackrel{\text{def}}{=} \frac{1}{K}\sum_{i=1}^{N} \frac{\binom{N-2}{i-1}}{i}$, *and* $\beta \stackrel{\text{def}}{=} \frac{1}{K}\sum_{i=1}^{N} \frac{\binom{N-2}{i-2}}{i}$. *Then* $G(x) = \alpha p(x) + \beta$ *for* $x \in \Omega$.

Proof.

$$G(x) = \sum_{Y\subseteq\Omega} \frac{x\cap Y}{|Y|} m(Y) = \sum_{Y\subseteq\Omega, x\in Y} \frac{1}{|Y|}\frac{P(Y)}{K} =$$
$$= \frac{1}{K} \sum_{Y\subseteq\Omega, x\in Y} \frac{P(Y)}{|Y|} = \frac{1}{K} \sum_{Y\subseteq\Omega, x\in Y} \frac{\sum_{z\in Y} p(z)}{|Y|} =$$
$$= \frac{1}{K} \sum_{i=1}^{N} \frac{1}{i} \sum_{Y\subseteq\Omega,|Y|=i, x\in Y} \sum_{z\in Y} p(z) =$$
$$= \frac{1}{K} \sum_{i=1}^{N} \frac{1}{i} \left(\binom{N-1}{i-1} p(x) + \binom{N-2}{i-2} \sum_{z\neq x} p(z) \right) =$$
$$= \frac{1}{K} \sum_{i=1}^{N} \frac{1}{i} \left(\binom{N-1}{i-1} p(x) + \binom{N-2}{i-2} (1-p(x)) \right) =$$
$$= \frac{1}{K} \sum_{i=1}^{N} \frac{1}{i} \left(\binom{N-2}{i-1} p(x) + \binom{N-2}{i-2} \right) =$$
$$= \alpha p(x) + \beta.$$

□

The claim then follows.

Since both P and G are probability functions we have $\sum_{x\in\Omega} P(x) = 1$ and $\sum_{x\in\Omega} G(x) = 1$. According to Theorem 2 we then have:

Corollary 1. $\alpha + |\Omega| \times \beta = 1$.

As a result we only need to calculate either of α and β, and the other can be determined according to the corollary.

Since both P and G are probability functions they satisfy the additive axiom. In other words for $E \subseteq \Omega$, $P(E) = \sum_{x\in E} p(x)$ and $G(E) = \sum_{x\in E} G(x)$. Following Theorem 2 we then have:

Corollary 2. $G(E) = \alpha P(E) + \beta |E|$.

Theorem 2 and Corollary 2 establish the relationship between G and probability distribution, and G and probability respectively. If we have full knowledge about the distribution we can calculate probability, which can further be used to calculate G. On the contrary, if we have full knowledge about G then we can calculate distribution and probability *precisely*. The interesting question is, if we have only incomplete or partial knowledge about G then we can get an approximation to the probability (and probability distribution). Therefore CPT can be used as a method for probability distribution estimation.

Example 2. Consider a set $\{a,b,c,d\}$, whose probability distribution is assumed to be $\{0.1, 0.3, 0.4, 0.2\}$. Following definition, the P, m and G values can be calculated for all the subsets (contexts) and are shown in Table 1.

Table 1
The set is $\Omega = \{a,b,c,d\}$. The probability distribution is assumed to be $\Omega = \{a:0.1, b:0.3, c:0.4, d:0.2\}$. Note that $p(x) = P(\{x\})$ for $x \in \Omega$

E	∅	{a}	{b}	{c}
P	0	0.1	0.3	0.4
m	0	1/80	3/80	4/80
G	0	198/960	254/960	282/960
E	{d}	{a,b}	{a,c}	{a,d}
P	0.2	0.4	0.5	0.3
m	2/80	4/80	5/80	3/80
G	226/960	452/960	480/960	424/960
E	{b,c}	{b,d}	{c,d}	{a,b,c}
P	0.7	0.5	0.6	0.8
m	7/80	5/80	6/80	8/80
G	536/960	480/960	508/960	734/960
E	{a,b,d}	{a,c,d}	{b,c,d}	{a,b,c,d}
P	0.6	0.7	0.9	1.0
m	6/80	7/80	9/80	10/80
G	678/960	706/960	762/960	1

Clearly the G values for singleton subsets are slightly different from the those P values given in probability distribution.

Let's now illustrate the relationship between P and G with respect to Theorem 2.

Here Ω has four elements so $N = 4$. Then according to Lemma 2, $K = 2^{N-1} = 2^3 = 8$. Other components in the theorem are calculated as follows:

$$\alpha = \frac{1}{K}\sum_{i=1}^{N} \frac{\binom{N-2}{i-1}}{i} =$$
$$= \frac{1}{8}\left(\binom{2}{0} + \frac{\binom{2}{1}}{2} + \frac{\binom{2}{2}}{3} + \frac{\binom{2}{3}}{4} \right) = \frac{1}{8}\left(1 + 1 + \frac{1}{3}\right) = \frac{28}{96}$$

$$\beta = \frac{1}{K}\sum_{i=1}^{N} \frac{\binom{N-2}{i-2}}{i} =$$
$$= \frac{1}{8}\left(\binom{2}{-1} + \frac{\binom{2}{0}}{2} + \frac{\binom{2}{1}}{3} + \frac{\binom{2}{2}}{4} \right) = \frac{1}{8}\left(\frac{1}{2} + \frac{2}{3} + \frac{1}{4}\right) = \frac{17}{96}.$$

Now we show that $P(x) = [G(x) - \beta]/\alpha$ for every elements in the set.

$$[G(a) - \beta]/\alpha = \left(\frac{198}{960} - \frac{17}{96}\right) \times \frac{96}{28} = \frac{28}{960} \times \frac{96}{28} = 0.1$$

$$[G(b) - \beta]/\alpha = \left(\frac{254}{960} - \frac{17}{96}\right) \times \frac{96}{28} = \frac{84}{960} \times \frac{96}{28} = 0.3$$

$$[G(c) - \beta]/\alpha = \left(\frac{282}{960} - \frac{17}{96}\right) \times \frac{96}{28} = \frac{112}{960} \times \frac{96}{28} = 0.4$$

$$[G(d) - \beta]/\alpha = \left(\frac{226}{960} - \frac{17}{96}\right) \times \frac{96}{28} = \frac{56}{960} \times \frac{96}{28} = 0.2$$

Clearly the equation holds.
To illustrate Corollary 2, consider context $E = \{a,b,c\}$. By the additive property of the G function we have

$$G(E) = G(a) + G(b) + G(c) = 734/960.$$

By the definition of G we have

$$\begin{aligned} G_1(E) &= m(a) + m(b) + m(c) + m(\{a,b\}) + m(\{a,c\}) + \\ &+ m(\{b,c\}) + m(\{a,b,c\}) + \frac{3}{4} \times m(\{a,b,c,d\}) = \\ &= \frac{1}{80} + \frac{3}{80} + \frac{4}{80} + \frac{4}{80} + \frac{5}{80} + \frac{7}{80} + \frac{8}{80} + \frac{3 \times 10}{4 \times 80} = \\ &= \frac{158}{320} \end{aligned}$$

$$\begin{aligned} G_2(E) &= \frac{1}{2} m(\{a,d\}) + \frac{1}{2} m(\{b,d\}) + \frac{1}{2} m(\{c,d\}) = \\ &= \frac{3}{2 \times 80} + \frac{5}{2 \times 80} + \frac{6}{2 \times 80} = \frac{14}{160} \end{aligned}$$

$$\begin{aligned} G_3(E) &= \frac{2}{3} m(\{a,b,d\}) + \frac{2}{3} m(\{a,c,d\}) + \frac{2}{3} m(\{b,c,d\}) = \\ &= \frac{2 \times 6}{3 \times 80} + \frac{2 \times 7}{3 \times 80} + \frac{2 \times 9}{3 \times 80} = \frac{44}{240} \end{aligned}$$

$$G(E) = G_1(E) + G_2(E) + G_3(E) = \frac{734}{960}.$$

The probability function $P(E)$ is calculated according the additive property as follows:

$$P(E) = P(\{a\}) + P(\{b\}) + P(\{c\}) = 0.8.$$

Using the α and β values above we have

$$P(E) \times \alpha + |E| \times \beta = 0.8 \times \frac{28}{96} + 3 \times \frac{17}{96} = \frac{734}{960} = G(E)$$

Example 3. Now let's look at another example with the same Ω and the same underlying probability distribution. We assume that we do not know the distribution explicitly, but we know the probability values for some subsets. These values are normalized to give a mass function. The known probability and mass values are shown in Table 2.

Table 2
The probability and mass values for Example 3

E	{a,b}	{a,c}	{b,d}	{c,d}
P	0.4	0.5	0.5	0.6
m	4/50	5/50	5/50	6/50
E	{a,b,c}	{a,b,d}	{a,c,d}	{b,c,d}
P	0.8	0.6	0.7	0.9
m	8/50	6/50	7/50	9/50

By definition we have

$$\begin{aligned} G(a) &= \frac{1}{2}[m(\{a,b\}) + m(\{a,c\})] + \\ &+ \frac{1}{3}[m(\{a,b,c\}) + m(\{a,b,d\}) + m(\{a,c,d\})] = \\ &= \frac{1}{2} \times \frac{4+5}{50} + \frac{1}{3} \times \frac{8+6+7}{50} = \frac{69}{300} = 0.230 \end{aligned}$$

$$\begin{aligned} G(b) &= \frac{1}{2}[m(\{a,b\}) + m(\{b,d\})] + \\ &+ \frac{1}{3}[m(\{a,b,c\}) + m(\{a,b,d\}) + m(\{b,c,d\})] = \\ &= \frac{1}{2} \times \frac{4+5}{50} + \frac{1}{3} \times \frac{8+6+9}{50} = \frac{73}{300} = 0.243 \end{aligned}$$

$$\begin{aligned} G(c) &= \frac{1}{2}[m(\{a,c\}) + m(\{c,d\})] + \\ &+ \frac{1}{3}[m(\{a,b,c\}) + m(\{a,c,d\}) + m(\{b,c,d\})] = \\ &= \frac{1}{2} \times \frac{5+6}{50} + \frac{1}{3} \times \frac{8+7+9}{50} = \frac{81}{300} = 0.270 \end{aligned}$$

$$\begin{aligned} G(d) &= \frac{1}{2}[m(\{b,d\}) + m(\{c,d\})] + \\ &+ \frac{1}{3}[m(\{a,b,d\}) + m(\{a,c,d\}) + m(\{b,c,d\})] = \\ &= \frac{1}{2} \times \frac{5+6}{50} + \frac{1}{3} \times \frac{6+7+9}{50} = \frac{77}{300} = 0.257 \end{aligned}$$

To calculate P values from respective G values we need K, α and β, which are functions of N. From Example 2 we know that $K = 8$, $\alpha = 28/96 = 0.292$ and $\beta = 17/96 = = 0.177$. $P(x)$ can be calculated by $P(x) = [G(x) - \beta]/\alpha$ for every elements as follows:

$$P(\{a\}) = (0.230 - 0.177)/0.292 = 0.18$$

$$P(\{b\}) = (0.243 - 0.177)/0.292 = 0.23$$

$$P(\{c\}) = (0.270 - 0.177)/0.292 = 0.32$$

$$P(\{d\}) = (0.257 - 0.177)/0.292 = 0.27$$

With these values we can calculate probability for any other subsets.

If we apply Dempster-Shafer theory we can calculate the belief and plausibility values for any contexts. For example

$$bel(\{x\}) = 0 \text{ for any } x \in \Omega.$$

The plausibility value is calculated as follows:

$$pls(\{a\}) = m(\{a,b\}) + m(\{a,c\}) + m(\{a,b,c\}) + \\ + m(\{a,b,d\}) + m(\{a,c,d\}) = 30/50$$

$$pls(\{b\}) = m(\{a,b\}) + m(\{b,d\}) + m(\{a,b,c\}) + \\ + m(\{a,b,d\}) + m(\{b,c,d\}) = 32/50$$

$$pls(\{c\}) = m(\{a,c\}) + m(\{c,d\}) + m(\{a,b,c\}) + \\ + m(\{a,c,d\}) + m(\{b,c,d\}) = 35/50$$

$$pls(\{d\}) = m(\{b,d\}) + m(\{c,d\}) + m(\{a,b,d\}) + \\ + m(\{a,c,d\}) + m(\{b,c,d\}) = 33/50.$$

5. Summary and conclusion

In this paper we have presented a new probability function G – contextual probability function, which is defined in terms of a basic probability assignment – mass function. Therefore G has all the properties of the classical probability function, which satisfies the three *axioms of probability*. The mass function has similar meaning as that in the D-S theory. Thus CPT enjoys the flexibility and other properties attributed to the D-S theory. The key difference between the two, however, is the fact that G is a probability function whereas the belief and plausibility functions are not. One consequence is that, due to the additive property, only the G values for singleton elements in Ω need to be calculated and the G values for any other subsets of Ω can be obtained from the G values for singletons. This is a big save in time. Since the belief and plausibility functions in the D-S theory are not additive we have to calculate belief and plausibility values for all subsets of Ω.

Though G is a probability function, CPT can be viewed as a generalization of the classical probability theory in the sense that if the mass function is defined only for elements in Ω, the G becomes the P function. The D-S theory is also regarded as a generalization of probability theory, but the belief and plausibility functions are not probability functions; they become probability functions when the mass function is defined only for elements in Ω.

The mass function can be interpreted in different ways for different purposes. We have shown that if it is interpreted as normalized (summing up to 1) probability function, G is a linear function of the P function. This connection makes it possible to estimate probability distribution from the probability values of some known events.

Future work should include interpreting the mass function in other ways for other purposes and applying the CPT to some real world problems.

References

[1] W. Feller, *An Introduction to Probability Theory and its Applications*. Wiley, 1968.

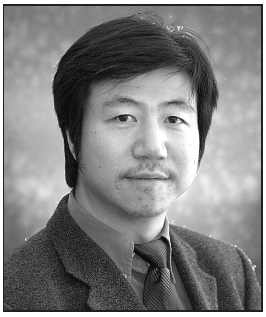
[2] J. W. Guan and D. A. Bell, "Generalization of the Dempster-Shafer theory" in *Proc. IJCAI-93*, 1993, pp. 592–597.

[3] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. The MIT Press, 2001.

[4] E. T. Jaynes, "Probability theory: the logic of science", http://bayes.wustl.edu

[5] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[6] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, New Jersey: Princeton University Press, 1976.



Hui Wang is a senior lecturer at the University of Ulster, Northern Ireland, UK. He received his B.Sc. and M.Sc. from Jilin University (China) and Ph.D. from the University of Ulster. His research interests include machine learning, data mining, text classification, probability theory, qualitative reasoning, neural networks, and Internet technologies.

e-mail: H.Wang@ulster.ac.uk
School of Computing and Mathematics
Faculty of Informatics
University of Ulster
Newtownabbey, Northern Ireland
BT 37 0QB, United Kingdom

Paper

Decision algorithms and flow graphs; a rough set approach

Zdzisław Pawlak

Abstract — This paper concerns some relationship between Bayes' theorem and rough sets. It is revealed that any decision algorithm satisfies Bayes' theorem, without referring to either *prior* or *posterior* probabilities inherently associated with classical Bayesian methodology. This leads to a new simple form of this theorem, which results in new algorithms and applications. Besides, it is shown that with every decision algorithm a flow graph can be associated. Bayes' theorem can be viewed as a flow conservation rule of information flow in the graph. Moreover, to every flow graph the Euclidean space can be assigned. Points of the space represent decisions specified by the decision algorithm, and distance between points depicts distance between decisions in the decision algorithm.

Keywords — rough sets, decision algorithms, flow graphs, data mining.

1. Introduction

Decision algorithm is a finite set of "*if .. then*" decision rules. With every decision rule three coefficients are associated: the *strength*, the *certainty* and the *coverage factors* of the rule. The coefficients can be computed from the data or can be a subjective assessment. It is shown that these coefficients satisfy Bayes' formula.

Bayesian inference methodology consists in updating *prior* probabilities by means of data to *posterior* probabilities, which express updated knowledge when data become available. The strength, certainty and coverage factors can be interpreted either as probabilities (objective), or as a degree of truth, along the line proposed by Łukasiewicz [5]. Moreover, they can be also interpreted as a deterministic flow distribution in flow graphs associated with decision algorithms. This leads to a new look on Bayes' theorem and its applications in reasoning from data, without referring to its probabilistic character.

In this context it is worthwhile to mention that in spite of great power of statistical Bayesian methodology of inference methods, the theorem raised wide criticism. E.g., "The technical result at the heart of the essay is what we now know as *Bayes' theorem*. However, from a purely formal perspective there is no obvious reason why this essentially trivial probability result should continue to excite interest" [1]. "Opinion as to the values of Bayes' theorem as a basic for statistical inference has swung between acceptance and rejection since its publication on 1763" [2]. In the proposed setting Bayes' theorem has been set free from its mystic flavor. With every decision algorithm a flow graph can be associated. It is revealed that the throughflow in the graph is ruled by Bayes' theorem. The flow graphs considered in this paper are different to flow networks introduced by Ford and Fulkerson [4], which are intended to model the flow in transportation network – in contrast to flow graphs, which are meant to be used as a model for decision analysis in decision algorithms. Besides, with every decision algorithm the Euclidian decision space is associated. The decision space is intended to be used to depict differences between decisions of a decision algorithm in a geometrical way.

2. Decision algorithms

A *decision rule* is an expression in the form $\Phi \to \Psi$, read "*if* Φ *then* Ψ", where Φ and Ψ are logical formulas called *condition* and *decision* of the rule, respectively [8].

Let $|\Phi|$ denote the set of all objects from the universe U, having the property Φ.

If $\Phi \to \Psi$ is a decision rule then $supp(\Phi,\Psi) = card(|\Phi \wedge \Psi|)$ will be called the *support* of the decision rule and

$$\sigma(\Phi,\Psi) = \frac{supp(\Phi,\Psi)}{card(U)}$$

will be referred to as the *strength* of the decision rule.

With every decision rule $\Phi \to \Psi$ we associate a *certainty factor*

$$cer(\Phi,\Psi) = \frac{supp(\Phi,\Psi)}{card(|\Phi|)}$$

and a *coverage factor*

$$cov(\Phi,\Psi) = \frac{supp(\Phi,\Psi)}{card(|\Psi|)}.$$

Remark. These coefficients for a long time have been used in data bases and machine learning [9, 10], but first they have been introduced by Łukasiewicz [5] in connection with his study of logic and probability.

If $cer(\Phi,\Psi) = 1$, then the decision rule $\Phi \to \Psi$ will be called *certain*, otherwise the decision rule will be referred to as *uncertain*.

A set of decision rules $Dec(\Phi,\Psi) = \{\Phi_i \to \Psi_i\}_{i-1}^{n}$ $n \geq 2$, will be called a *decision algorithm* if all its decision rules are:

- *admissible*, i.e., $supp(\Phi_i,\Psi)_i \neq \emptyset$ for every $1 \leq i \leq n$,
- *mutually exclusive* (*independent*), i.e., for every $\Phi_i \to \Psi_i$ and $\Phi_j \to \Psi_j$, $\Phi_i = \Psi_j$, or $\Phi_i \wedge \Psi_j = \emptyset$ and $\Psi_i = \Psi_j$ or $\Psi_i \wedge \Phi_j = \emptyset$,
- *cover* U, i.e., $\left|\bigvee_{i-1}^{n} \Phi_i\right| = \left|\bigvee_{i-1}^{n} \Psi_i\right| = U$.

If $Dec(\Phi,\Psi) = \{\Phi_i \to \Psi_i\}_{i=1}^n$ is a decision algorithm, then $Dec(\Psi,\Phi) = \{\Psi_i \to \Phi_i\}_{i=1}^n$ is also a decision algorithm and will be called an *inverse decision algorithm* of $Dec(\Phi,\Psi)$.
$Dec(\Psi,\Phi)$ gives *reasons* (*explanations*) for decisions of the algorithm $Dec(\Phi,\Psi)$.

3. Properties of decision algorithms

Let $Dec(\Phi,\Psi)$ be a decision algorithm and let $\Phi \to \Psi$ be a decision rule in the decision algorithm. By $D(\Phi)$ and $C(\Psi)$ we denote the set of all decisions of Φ and the set of all conditions of Ψ in $Dec(\Phi,\Psi)$, respectively [8].
It can be shown that every decision algorithm has the following probabilistic properties:

$$\sum_{\Phi' \in C(\Psi)} cov(\Phi',\Psi) = 1, \tag{1}$$

$$\sum_{\Psi' \in D(\Phi)} cer(\Phi,\Psi') = 1, \tag{2}$$

$$\pi(\Psi) = \sum_{\Phi' \in C(\Psi)} cer(\Phi',\Psi)\cdot\pi_S(\Phi') = \sum_{\Phi' \in C(\Psi)} \sigma(\Phi',\Psi), \tag{3}$$

$$\pi(\Phi) = \sum_{\Psi' \in D(\Phi)} cov(\Phi,\Psi')\cdot\pi_S(\Psi') = \sum_{\Psi' \in D(\Phi)} \sigma(\Phi,\Psi'), \tag{4}$$

$$cer(\Phi,\Psi) = \frac{cov(\Phi,\Psi)\cdot\pi(\Psi)}{\sum_{\Psi' \in D(\Phi)} \sigma(\Phi,\Psi')} = \frac{\sigma(\Psi,\Phi)}{\pi(\Phi)}, \tag{5}$$

$$cov(\Phi,\Psi) = \frac{cer(\Phi,\Psi)\cdot\pi(\Phi)}{\sum_{\Phi' \in C(\Psi)} \sigma(\Phi',\Psi)} = \frac{\sigma(\Phi,\Psi)}{\pi(\Psi)}, \tag{6}$$

where $\pi(\Psi) = \frac{card(|\Psi|)}{card(U)}$ and $\pi(\Phi) = \frac{card(|\Phi|)}{card(U)}$.

Let us observe that formally formulas (1)–(6) have probabilistic favor. In particular, formulas (3) and (4) can be understood as total probability theorems, whereas formulas (5) and (6) have the form of Bayes' theorem. As mentioned before, if we interpret strength, certainty and coverage factors as probabilities (frequencies) then these formulas describe some relationships between data in the decision algorithm. However, we can also interpret these factors in a deterministic way, as a degree of truth. In this case the coverage factor $cov(\Phi,\Psi)$ expresses to which degree the conclusion Ψ of a decision rule $\Phi \to \Psi$ is true if the condition Φ of the rule is true to the degree $cer(\Phi,\Psi)$, whereas $\sigma(\Phi,\Psi)$ can be regarded as a truth value of the decision rule (implication) $\Phi \to \Psi$.

The idea to replace probability by truth values is due to Łukasiewicz [5], but we will not discuss this issue here.

4. Flow graphs

With every decision algorithm we associate a directed, acyclic, connected graph defined in the following way: to every condition and decision of the decision rule in the decision algorithm we associate a *node* of the graph. To every decision rule $\Phi \to \Psi$ we assign a directed *branch* connecting the *input node* Φ and the *output node* Ψ. Strength of the decision rule represents the *throughflow* of the corresponding branch. More about flow graphs and decision algorithms can be found in [7].
The throughflow of the graph is governed by formulas (1)–(6), and can be considered as a *flow conservation equation* similar to that of Ford and Fulkerson [4].
Consequently, the flow graphs can be regarded as a third model of Bayes' theorem, in which the theorem describes flow distribution in a flow graph.

5. Decision space

With every decision algorithm with n-valued decisions we can associate n-dimensional Euclidean space, where values of decisions determine n axis of the space and condition attribute values (equivalence classes) determine point of the space. Strengths of decision rules are to be understood as coordinates of corresponding points.
Distance $\delta(x,y)$ between point x and y in an n-dimensional decision space is defined as

$$\delta(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},$$

where $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$ are vectors of strengths of corresponding decision rules.

6. An example

For the sake of illustration let us consider a very simple decision algorithm describing vote distribution for two political parties X_1, and X_2 from three mutually disjoint sample group of voters Y_1, Y_2 and Y_3:

1. $Y_1 \to X_1$ (400)
2. $Y_1 \to X_2$ (200)
3. $Y_2 \to X_1$ (250)
4. $Y_2 \to X_2$ (50)
5. $Y_3 \to X_1$ (90)
6. $Y_3 \to X_2$ (10).

Number given at the end of each rule is the support of the rule, i.e., the number of voters from group X_i voting for party Y_j.

The *strength, certainty* and *coverage* factors for each decision rule are given in Table 1.

Table 1
Parameters of the decision rules

Rule	Strength	Certainty	Coverage
1	0.40	0.67	0.54
2	0.20	0.33	0.77
3	0.25	0.83	0.34
4	0.05	0.17	0.19
5	0.09	0.90	0.12
6	0.01	0.10	0.04

The corresponding flow graph is shown in Fig. 1.

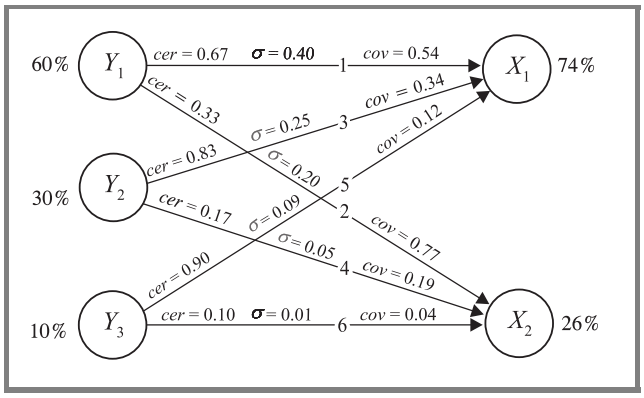


Fig. 1. The flow graph corresponding to the example.

Thus from the decision algorithm follows, for example, that 83% voters from group Y_2 voted for party X_1 and 17% voters voted for party X_2. From the inverse decision algorithm we get, for example, that for party X_1 voted 54% voters of group Y_1, 34% – of group Y_2, and 12% – of group Y_3. The corresponding distance space is shown in Fig. 2. Distances between voters are presented in Table 2.

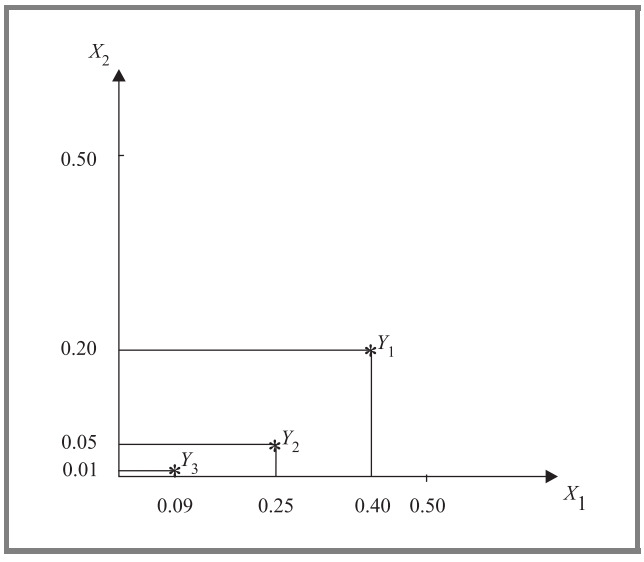


Fig. 2. The distance space corresponding to the example.

Table 2
Distances between voters

	Y_1	Y_2	Y_3
Y_1			
Y_2	0.20		
Y_3	0.37	0.22	

7. Summary

In this paper a relationship between decision algorithms, flow graphs and Bayes' theorem are defined and briefly analyzed. It is shown that decision algorithms satisfy Bayes' theorem, and that the theorem can be also interpreted without referring to its probabilistic connotation – in a purely deterministic way. This property leads to a new look on Bayes' theorem and new applications of Bayes' rule in data analysis.

References

[1] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. Chichester [etc.]: Wiley, 1994 (Wiley series in probability and mathematical statistics).

[2] G. E. P. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*. New York [etc.]: Wiley, 1992.

[3] M. Berthold and D. J. Hand, *Intelligent Data Analysis – an Introduction*. Berlin: Springer, 1999.

[4] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton [etc.]: Princeton University Press, 1962.

[5] J. Łukasiewicz, "Die logishen Grundlagen der Wahrscheinilchkeitsrechnung" (Kraków, 1913) in *Jan Łukasiewicz – Selected Works*, L. Borkowski, Ed. Amsterdam [etc.]: North Holland Publ., Warsaw: Polish Scientific Publ., 1970.

[6] S. Greco, Z. Pawlak, and R. Słowiński, "Generalized decision algorithms, rough inference rules and flow graphs" in *Rough Sets and Current Trends in Computing*, J. J. Alpigini, J. F. Peters, A. Skowron, and N. Zhong, Eds., *Lecture Notes in Artificial Intelligence*. Berlin: Springer, 2002, vol. 2475, pp. 93–104.

[7] Z. Pawlak, "In pursuit of patterns in data reasoning from data – the rough set way" in *Rough Sets and Current Trends in Computing*, J. J. Alpigini, J. F. Peters, A. Skowron, and N. Zhong, Eds., *Lecture Notes in Artificial Intelligence*. Berlin: Springer, 2002, vol. 2475, pp. 1–9.

[8] Z. Pawlak, "Rough sets, decision algorithms and Bayes' theorem", *Eur. J. Oper. Res.*, vol. 136, pp. 181–189, 2002.

[9] S. Tsumoto and H. Tanaka, "Discovery of functional components of proteins based on PRIMEROSE and domain knowledge hierarchy" in *Proc. Worksh. Rough Sets & Soft Comput. RSSC-94*, T. Y. Lin and A. M. Wildberger, Eds., *Soft Computing*, SCS, 1995, pp. 280–285.

[10] S. K. M. Wong and W. Ziarko, "Algorithm for inductive learning", *Bull. Polish Acad. Sci.*, vol. 34, no. 5-6, pp. 271–276, 1986.

Zdzisław Pawlak was born in Łódź (Poland), in 1926. He obtained his M.Sc. in 1951 in electronics from Warsaw University of Technology, Ph.D. in 1958 and D.Sc. in 1963 in the theory of computation from the Polish Academy of Sciences. He is a Professor of the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences and the University of Information Technology and Management and Member of the Polish Academy of Sciences. His current research interests include intelligent systems and cognitive sciences, in particular, decision support systems, knowledge representation, reasoning about knowledge, machine learning, inductive reasoning, vagueness, uncertainty and decision support. He is an author of a new mathematical tool, called rough set theory, intended to deal with vagueness and uncertainty. About two thousand papers have been published by now on rough sets and their applications world wide. Several international workshops and conferences on rough sets have been heldin recent years. He is a recipient of many awards among others the State Award in Computer Science in 1978, the Hugo Steinhaus Award for achievements in applied mathematics in 1989. Doctor honoris causa of Poznań University of Technology in 2002. Member of editorial boards of several dozens international journals. Deputy Editor-in-Chief of the Bulletin of the Polish Academy of Sciences. Program committee member of many international conferences on computer science. Over forty visiting university appointments in Europe, USA and Canada, about fifty invited international conference talks, and over one hundred seminar talks given in about fifty universities in Europe, USA, Canada, China, India, Japan, Korea, Taiwan, Australia and Israel. About two hundred articles in international journals and several books on various aspects on computer science and application of mathematics. Supervisor of thirty Ph.D. theses in computer science and applied mathematics.

e-mail: zip@mimuw.edu.pl
University of Information Technology and Management
Newelska st 6
01-447 Warsaw, Poland

Paper

A world according to artificial neural networks

Alfons Schuster

Abstract — This paper presents results from a preliminary study in the field of artificial neural networks (ANN). The overall aim of our work relates to the field of cognitive science. In this wider framework we try to investigate, reason about, and model cognitive processes in order to obtain a better understanding of the major processing device involved – the human brain. In terms of content this paper presents a novel ANN learning approach. Note that throughout the paper we assume supervised learning. In contrast to the classical ANN learning approach where an ANN algorithm alters an initial random weight assignment until a reasonable solution to a problem is obtained this approach does not alter the initial random weight assignment at all, but provides a solution to the problem by transforming the actual input data. The approach is applied to perceptrons and adalines and its quality is demonstrated on simple classification problems.

Keywords — artificial neural networks, cognitive science, input space transformation.

1. Introduction

A few examples are chosen in order to provide a quick introduction and also to illustrate the motivation behind this work.

Example 1. Imagine a student looking for material supporting an assignment. The student probably browses several books in a library and maybe selects a few of them for more detailed study. Important in this example is (a) the information required by the student is external, in library books etc., (b) the information is represented in different formats, i.e. different books may cover the same topics but the styles etc. may be different in each book, (c) after a while the student may have forgotten some of the studied material, i.e. the information may no longer be available in the brain of the student. In order to re-acquire the information the student may need to go back to the library. On the other hand some of the information may not be considered (studied) on purpose. For example, in case the information is false or obsolete.

Example 2. Imagine a person having a dog and also having a picture of the same dog. Further imagine a second person pointing to the real dog asking the question: *Who's dog is this?* Then the dog owner will answer: *This is my dog*. Imagine now the second person pointing to the picture of the dog asking the same question: *Who's dog is this?* Then the person will answer again: *This is my dog*. The interesting aspects here are (a) there are two completely different representations of the dog, namely the dog itself and the picture of the dog, (b) for the purpose of answering the question however both representations are sufficient, (c) the two representations are external to the processing device – the brain of the dog owner, and (d) the original data, the physical dog, has been transformed into a different representation, the picture.

Example 3. Philosophers have been thinking about similar problems in the past. An example involving the philosopher Immanuel Kant can be summarised as follows [1]. Imagine a person wearing green glasses since the day of his/her birth. The person then would see the world in a shade of green. This may not necessarily have a major influence on the person's life. If the person ever took of the glasses then the person would see the world differently. The points here are (a) how can we be sure to experience the world with our senses as the world really is, i.e. how can we be sure we are not wearing some sort of glasses? and (b) the data transformation (here a green shift) may not necessarily have an impact on our ability to do things or on aspects of our reasoning. For example, the ability to swim is independent of the color of the water.

Example 4. This example particularly focuses on artificial neural networks. ANN learning usually starts with a random weight assignment followed by a training process using a training set [2]. The training process alters the initial random weight configuration and establishes a final weight configuration. This final configuration remains static and specifies the ANN for a particular application scenario. From the viewpoint of ANNs as being a model of the human brain this approach presents a number of questions. For example, let N be the total number of neurons in the brain. Further, imagine a simple classification scenario that can be solved by the utilisation of an ANN. Let n be the total number of neurons in this particular ANN. Now imagine this ANN presenting a part or region in the brain. Since the weight configuration of this ANN remains fixed or static after the training process this part of the brain (the ANN) can only be applied for the particular task it is designed for. This however could lead to the possible conclusion that the number of neurons in the brain that remains available for other tasks is now reduced to $N-n$. Although the number of neurons in the brain is quite substantial this means that sooner or later the brain runs out of neurons. This example may be a bit naive, but it helps to illustrate a major difference between the brain and ANNs, namely the high dynamic and flexibility of the brain opposed to the static inflexibility of ANNs.

Example 5. This is not really an example but rather a consideration. If there are different representations of an entity then a learning device such as an ANN, for example, needs

to be constructed for every representation. The previous example mentioned that this may require a separate set of neurons for the processing of each representation. From an energy point of view all these processes require energy. Nature usually looks for a maximum in efficiency through a minimum of effort (energy). The processing of each representation individually may contradict this particular drive. A summary of these examples may be:

- The same information can be represented in different formats, shapes, representations.
- The different representations of the information are usually external to a learner or learning device (student, brain, ANN).
- Some information may be false or obsolete and therefore may not be needed to be acquired (studied, learned).
- Some of the information learned may get lost or forgotten.
- In case there are different representations for the same information then it is ineffective for a learning device in terms of energy to construct a model for each representation.

Based on these examples and observations we here consider the following approach:

- Instead of producing a model for the learning of different representations of the same information use a single model that is able to learn the different representations.

Very generally speaking this paper has a focus on this issue and investigates a new approach to this problem. The major difference of this approach lies in the fact that instead of modifying the weight configuration of an ANN until the weight configuration suits the problem the approach proposed here transforms the actual data representing the problem until the data fits the initial configuration of the learning device (ANN). The initial weigh configuration assigned to the ANN at the start of the training process remains unchanged throughout this process. The transformed data can be viewed as a different representation of the problem and the examples given before indicate that it is possible to arrive at meaningful conclusions using different representations of the same information (dog/picture). One of the interesting consequences of this approach is that a single, random weight configuration can be used for many different scenarios.

The reminder of the paper is organised as follows. Section 2 presents the basic idea behind this paper, namely ANN training algorithms that are based on the transformation of input data rather than on the modification of weight values. Section 3 introduces two such algorithms. Section 4 summarises the results obtained from an application of the approach on simple classification tasks. Section 5 ends the paper with a summary.

2. Classification through data input space transformation

This section outlines the basic idea behind this paper. Although the paper deals with two well-known ANN training algorithms, namely the perceptron training algorithm and the Adaline training algorithm this section only refers to the perceptron training algorithm [3, 4]. This is basically due to the fact that the situation for both algorithms is very similar. The paper uses a simple classification task as a run-through example for illustration purposes. The task is illustrated in Fig. 1 and involves the correct classification of a particular number of different objects into one of two classes, *Class 1* or *Class 2*.

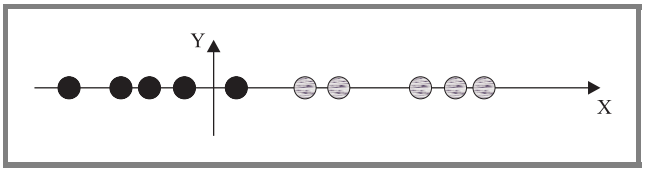


Fig. 1. A simple, one-dimensional, linearly separable classification task.

Imagine, for example, that Fig. 1 illustrates an arbitrary real valued *x-y* co-ordinate system in which the classification scenario takes place. Let the black dots in Fig. 1 represent objects of *Class 1* and the lined circles objects of *Class 2*. From the viewpoint of a classification task Fig. 1 illustrates a simple, one-dimensional, linearly separable task. Such a task can be solved by a perceptron, for example.

2.1. Perceptron classification

At this stage it is not necessary to know the exact details of a perceptron, the details are going to be explained at a later stage. The motivation in this section is to establish an understanding for the basic strategy behind perceptron classification. The perceptron training algorithm starts with a random weight assignment to the perceptron. In the context of the one-dimensional task at hand such a random weight assignment represents an arbitrary division point on the *x*-axes. For example, let the diamond in Fig. 2a be such an initial, random division point.

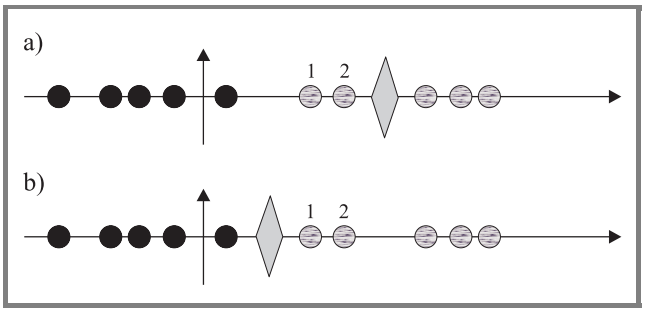


Fig. 2. (a) Possible start of perceptron learning process; (b) possible end of perceptron learning process.

Note that the diamond in Fig. 2a does not separate all objects into their correct classes. The two objects indexed 1 and 2 in Fig. 2a are misclassified. In order to achieve the correct classification of all objects the perceptron training algorithm alters the initial weight assignment in a number of successive, defined steps. This process usually continues until either all objects are classified correctly, or until a predefined number of iterations is reached. In illustrative terms the process of continuously altering the weight configuration of the perceptron is equivalent to an organised movement of the diamond along the *x*-axis in Fig. 2a. For example, after the training process the diamond might end up in the position illustrated in Fig. 2b. This position actually represents a solution to the problem, because all samples of *Class 1* are now on one side of the diamond and all objects of *Class 2* are on the other side. The actual details of the perceptron training algorithm are not so important at the moment, they will be discussed later. At the moment this study is interested in the question whether there is an alternative solution to the classification task given in Fig. 1.

2.2. An alternative solution to the classification problem

Figure 3 indicates an alternative solution to the problem at hand. In this particular case this alternative solution shall be referred to as alternative classification algorithm, or simply ACA.

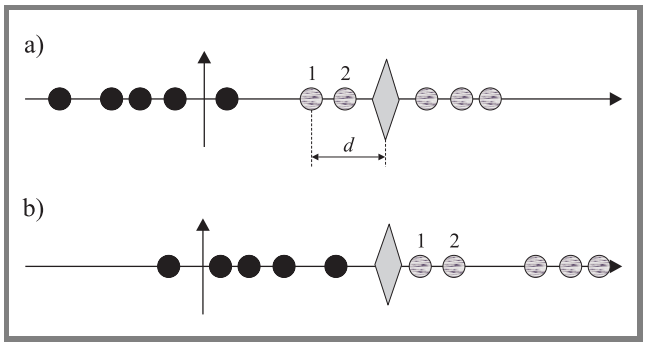


Fig. 3. (a) Original classification scenario; (b) alternative solution to the classification task through a transformation of the actual input data.

Figure 3 illustrates that the ACA is basically an input data transformation process. The ACA also starts with an initial random weight assignment. But then, instead of altering the weight values, which would be equivalent to moving the initial diamond around, the ACA transforms or shifts the actual input data in the *x*-*y* co-ordinate system until the initial location of the diamond presents a solution to the classification task. For example, Fig. 3a represents the original scenario. Note again that the diamond in Fig. 3a does not separate all objects into their correct classes. Figure 3b illustrates a possible scenario after the ACA has finished a training session. Figure 3b indicates that the location of the diamond remains unchanged throughout this training session, but also that the input data has been transformed in such a way that the initial diamond now presents a solution to the classification task. The diamond now separates all objects of *Class 1* from objects of *Class 2*. This transformation of input data is the basic principle behind the algorithms presented in the forthcoming sections. Figure 3 provides one more piece of information. In order to come up with the correct classification of all objects the input data has to be transformed by a certain amount. The index d in the figure indicates that the magnitude (m) of this transformation or shift has to be $m \geq d$.

3. Modified algorithms for perceptron and Adaline

This section presents two new ANN training algorithms. The two algorithms are modifications of the well-known perceptron and Adaline training algorithm and therefore are referred to as modified perceptron algorithm (MPA) and modified Adaline algorithm (MAA). The characteristic feature of the MPA and the MAA is the implementation of the ideas presented throughout the previous sections, that is, classification achieved through the transformation of the data input space as opposed to the update of weight values.

3.1. Perceptron and modified perceptron training algorithm

The MPA algorithm is a derived modification of the perceptron training algorithm. The section therefore starts with a brief introduction to perceptrons. Figure 4 illustrates a typical perceptron.

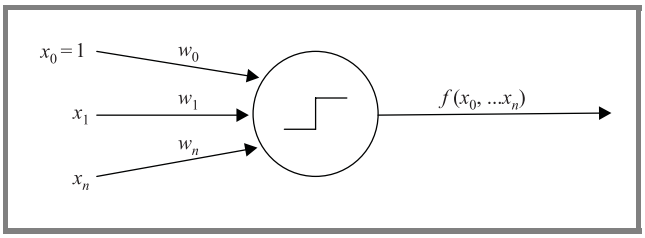


Fig. 4. A typical perceptron.

The perceptron in Fig. 4 has one output, and an undefined number of n inputs $(x_0, \ldots, x_n)$. The dummy input x_0, often called bias, has a constant value of one. Every input has a weight value $(w_0, \ldots, w_n)$ associated with it. The input values and the weight values are usually dealt with as vectors (e.g., $\boldsymbol{w}$ and $\boldsymbol{i}$). The output uses the step function $f(x_0, \ldots, x_n)$ in order to determine into which of two available classes an object belongs. The step function is applied to the weighted sum of the inputs to the perceptron and is often defined as follows:

$$f(x_0, \ldots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 & \text{otherwise} \end{cases}, \qquad (1)$$

where x_i and w_i are components of the input vector $\boldsymbol{i}$ and the weight vector $\boldsymbol{w}$, and the weighted sum $\sum_{i=0}^{n} w_i x_i$ is

given by the scalar product between the input vector and the weight vector as: $\boldsymbol{w}\cdot\boldsymbol{i} = w_0x_0 + \ldots + w_nx_n$. Note that solving the one-dimensional, linear separable classification task illustrated in Fig. 1 only requires a single perceptron with two inputs, x_0 and x_1.

In order to explain the perceptron training algorithm we make the following assumptions. For an object of *Class 1* the desired perceptron output shall be 1, and for an object of *Class 2* the desired output shall be -1. If an object is classified correctly then the perceptron remains unchanged. If the desired output is different from the actual output generated by the perceptron then the weight vector needs to be changed such that the error reduces. Theoretically this process is repeated until the desired output and the generated output are the same. In reality however linearly separable problems are not the norm and so the process usually runs for a predefined number of iterations.

Whenever an input vector is presented to a perceptron for classification two types of error can occur.

Case 1. The input vector $\boldsymbol{i}$ belongs to *Class 1* for which the desired perceptron output is 1, but $\boldsymbol{w}\cdot\boldsymbol{i} \leq 0$ (the actual perceptron output is -1).

Case 2. The input vector $\boldsymbol{i}$ belongs to *Class 2* for which the desired perceptron output is -1, but $\boldsymbol{w}\cdot\boldsymbol{i} > 0$ (the actual perceptron output is 1).

A perceptron strives to overcome both types of error through a defined weight vector update. This update establishes a new weight vector $\boldsymbol{w'}$ from a previous weight vector $\boldsymbol{w}$ according to: $\boldsymbol{w'} = \boldsymbol{w} + \Delta\boldsymbol{w}$. In the first case the update has to achieve that $\boldsymbol{w'}\cdot\boldsymbol{i} = (\boldsymbol{w} + \Delta\boldsymbol{w})\boldsymbol{i} > \boldsymbol{w}\cdot\boldsymbol{i}$. In the second case the aim is $\boldsymbol{w'}\cdot\boldsymbol{i} = (\boldsymbol{w} + \Delta\boldsymbol{w})\boldsymbol{i} < \boldsymbol{w}\cdot\boldsymbol{i}$. This behaviour can be achieved by letting $\Delta\boldsymbol{w} = \pm\eta\boldsymbol{i}$, where η is a positive constant called the learning rate. These concepts established define a weight update in the first case as:

$$\boldsymbol{w'}\cdot\boldsymbol{i} = (\boldsymbol{w} + \Delta\boldsymbol{w})\boldsymbol{i} = (\boldsymbol{w} + \eta\boldsymbol{i})\boldsymbol{i} = (\boldsymbol{w}\cdot\boldsymbol{i} + \eta\boldsymbol{i}\cdot\boldsymbol{i}) > \boldsymbol{w}\cdot\boldsymbol{i}. \quad (2)$$

Note in particular that η is a positive constant and that the scalar product $\boldsymbol{i}\cdot\boldsymbol{i} > 0$. In the second case the update looks like:

$$\boldsymbol{w'}\cdot\boldsymbol{i} = (\boldsymbol{w} + \Delta\boldsymbol{w})\boldsymbol{i} = (\boldsymbol{w} - \eta\boldsymbol{i})\boldsymbol{i} = (\boldsymbol{w}\cdot\boldsymbol{i} - \eta\boldsymbol{i}\cdot\boldsymbol{i}) < \boldsymbol{w}\cdot\boldsymbol{i}. \quad (3)$$

Equations 2 and 3 basically represent the core of the perceptron training algorithm. The task now is to design the MPA, the modified perceptron learning algorithm.

Modified perceptron training algorithm. Remember that the initial weight vector remains unchanged in the MPA training process. The MPA updates the input vector $\boldsymbol{i}$ instead. The MPA input vector update is defined as $\boldsymbol{i'} = \boldsymbol{i} + \Delta\boldsymbol{i}$ and so looks quite similar to a weight vector update described before.

The MPA uses this input vector update in order to deal with the two possible error scenarios mentioned before. For example, in the first case the input vector $\boldsymbol{i}$ belongs to *Class 1* for which the desired perceptron output is 1, but $\boldsymbol{w}\cdot\boldsymbol{i} \leq 0$ (the actual perceptron output is -1). In this situation the input vector is updated by the MPA such that $\boldsymbol{w}\cdot\boldsymbol{i'} = \boldsymbol{w}(\boldsymbol{i} + \Delta\boldsymbol{i}) > \boldsymbol{w}\cdot\boldsymbol{i}$. In the second case the input vector $\boldsymbol{i}$ belongs to *Class 2* for which the desired perceptron output is -1, but $\boldsymbol{w}\cdot\boldsymbol{i} > 0$ (the actual perceptron output is 1). Here the input vector is updated such that $\boldsymbol{w}\cdot\boldsymbol{i'} = \boldsymbol{w}(\boldsymbol{i} + \Delta\boldsymbol{i}) < \boldsymbol{w}\cdot\boldsymbol{i}$. With $\Delta\boldsymbol{i} = \pm\eta\boldsymbol{w}$ and the learning rate η it is possible to formulate an MPA update for the first case as follows:

$$\boldsymbol{w}\cdot\boldsymbol{i'} = \boldsymbol{w}(\boldsymbol{i} + \Delta\boldsymbol{i}) = \boldsymbol{w}(\boldsymbol{i} + \eta\boldsymbol{w}) = (\boldsymbol{w}\cdot\boldsymbol{i} + \eta\boldsymbol{w}\cdot\boldsymbol{w}) > \boldsymbol{w}\cdot\boldsymbol{i}. \quad (4)$$

Note, η is a positive constant and the scalar product $\boldsymbol{w}\cdot\boldsymbol{w} > 0$. In the second case the update appears as:

$$\boldsymbol{w}\cdot\boldsymbol{i'} = \boldsymbol{w}(\boldsymbol{i} + \Delta\boldsymbol{i}) = \boldsymbol{w}(\boldsymbol{i} - \eta\boldsymbol{w}) = (\boldsymbol{w}\cdot\boldsymbol{i} - \eta\boldsymbol{w}\cdot\boldsymbol{w}) < \boldsymbol{w}\cdot\boldsymbol{i}. \quad (5)$$

Although Eqs. 4 and 5 capture the essence of the MPA Fig. 5 may provide additional transparency to the whole process.

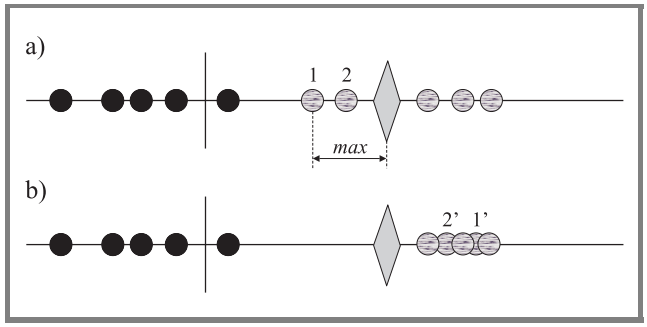


Fig. 5. A possible scenario for the MPA algorithm: (a) at the start; (b) at the end.

Figure 5a illustrates the familiar initial scenario with a random, but now static, weight assignment including the two misclassified objects carrying the labels 1 and 2. The MPA works similar to the perceptron training algorithm. If an object is classified correctly then the MPA does not interact. If however the desired output for an object is different from the actual output generated by the perceptron then the MPA alters the corresponding input vector until (in case of linear separability) the current object is classified correctly. Figure 5b aims to capture this process and illustrates that only the position of the two initially incorrect classified objects has changed. Figure 5b also indicates that the order of the position of such objects may change during the transformation process. For example, the new positions of objects 1 and 2 are now at 1' and 2'.

Further, the individual changes in the input vector of those objects that are actually transformed are recorded by the MPA. The MPA extracts the value *max* from this information (see Fig 5a). The value *max* determines the magnitude by which all objects need to be transformed along the *x*-axis in order to use the perceptron as a meaningful classifier. The value *max* is particularly important for the classification of unknown objects, that is objects that have

not been included in the training session. For such objects it is necessary to add an offset of magnitude *max* to their input vector.

Since results of an application of the MPA are presented and discussed at a later stage we here leave the perceptron training algorithm and its modified alternative and proceed with the investigation of the Adaline training algorithm.

3.2. Adaline and modified Adaline training algorithm

An Adaline is quite similar to a perceptron. For example, an Adaline also uses a step function in order to determine the class membership for different objects. For this reason and because both, perceptrons and Adalines, are well documented in the literature this section only elaborates on those issues that are necessary for the understanding of Adalines and the presented here alternative the modified Adaline training algorithm.

An Adaline is a system that like a perceptron aims for a reduction of the number of misclassifications through a defined weight update. The difference to the perceptron training algorithm is that the Adaline aims to achieve this task by minimising the mean square error $E = (d_j - net_j)^2$ of the system through the application of a gradient descent method, where d_j is the desired output for a particular input vector i_j, and net_j is the weighted sum generated by this input vector and a weight vector. Figure 6 illustrates a typical Adaline.

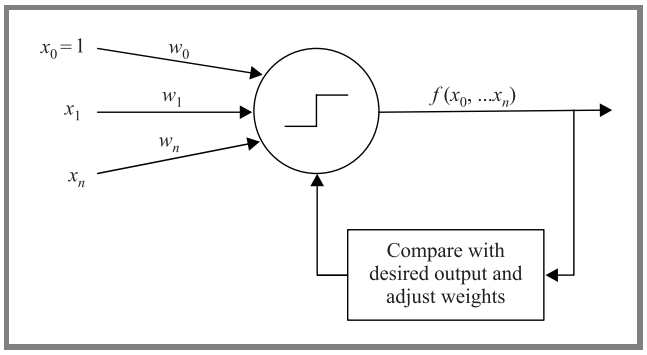


Fig. 6. An typical Adaline at work.

The basic formulas responsible for the learning process in an Adaline are given by Eqs. 6, 7, and 8. These formulas basically indicate how an updated weight value $\boldsymbol{w}$**'** is generated from a previous weight value $\boldsymbol{w}$:

$$\boldsymbol{w}' = \boldsymbol{w} + \Delta\boldsymbol{w}, \tag{6}$$

where

$$\Delta\boldsymbol{w} = \eta(d_j - net_j)i_j, \tag{7}$$

where η is the positive learning rate again, d_j is the desired output for a particular input vector i_j, and net_j is the weighted sum $\sum_{i=0}^{n} w_i x_{i,j}$ generated by the scalar product between the input vector i_j and the current weight vector $\boldsymbol{w}$. From Eqs. 6 and 7 the updated weight vector $\boldsymbol{w}$**'** is produced by:

$$\boldsymbol{w}' = \boldsymbol{w} + \eta\left(d_j - \sum_{i=0}^{n} w_i x_{i,j}\right) i_j. \tag{8}$$

A final difference between the perceptron and the Adaline needs to be mentioned. The perceptron training algorithm as well as its modification the MPA updates a weight vector/input vector only if the output generated by the perceptron is different from the desired output. On the other hand, the Adaline training algorithm and its modified alternative apply an update at every presentation of a sample to the system.

Modified Adaline training algorithm. The task now is similar as before and contains modifying Eq. 8 in such a way that instead of the weights the actual input data is transformed, without loosing the quality of the system as a classifier. The three equations below provide a summary of the mathematical procedures involved. They are very similar to Eqs. 6, 7, and 8 and form the basis for the modified Adaline algorithm MAA:

$$\boldsymbol{i}' = \boldsymbol{i} + \Delta\boldsymbol{i}, \tag{9}$$

$$\Delta\boldsymbol{i} = \eta(d_j - net_j)w_j, \tag{10}$$

$$\boldsymbol{i}' = \boldsymbol{i} + \eta\left(d_j - \sum_{i=0}^{n} w_i x_{i,j}\right) w_j, \tag{11}$$

where $\boldsymbol{i}$**'** is the resulting, updated input vector. A closer look at the different equations reveals again that the main difference is basically a substitution between the weight vector $\boldsymbol{w}$ and the input vector $\boldsymbol{i}$. So much for the mathematical background of the different algorithms. The next section provides a summary of the results obtained from an application of the different algorithms.

4. Results

All algorithms (the original perceptron and Adaline training algorithm as well as their modifications MPA and MAA) were applied and evaluated on simple classification tasks. The problem to solve was always a one-dimensional, linearly separable classification problem, similar to the problem illustrated in Fig. 1. The total number of objects involved in a classification task was variable, but every class contained the same number of objects. The *x*-axis in Fig. 1 was given by the interval [0, 1]. The position of the objects in this interval was generated by the random number procedure included in the Delphi5 programming tool that was utilised for the programming of the algorithms. The position of the initial division point (the diamond in Fig. 1) was variable and so it was possible to generate a particular number of misclassified objects.

Very generally speaking it can be said that the modified algorithms MPA and MAA performed more or less equally well as their original counterparts perceptron and Adaline did. Since the problems posed did not contain any outliers MPA and MAA were always able to solve the classification tasks given to them, provided the number of iterations that was sufficiently high. For a particular task the number of iterations needed by MPA and MAA was about in the same range as the number of iterations needed by the perceptron and Adaline algorithm. In case of outliers MPA and MAA face similar problems as the original perceptron and Adaline algorithm do. The similarity of the approaches may allow the assumption that approaches to tackle the issue of outliers in MPA and MAA are similar to approaches proposed and known for perceptrons and Adalines [2].
The overall conclusion for the undertaken study is that all the different algorithms did perform about equally well on the problems given to them, and so MPA and MAA do not stand back behind the traditional perceptron and Adaline algorithms.

5. Summary and future work

The paper presents two new classification algorithms. To some extent these algorithms are derived from the classical perceptron and Adaline training algorithm. The major difference between the new algorithms and the classical algorithms is in the data manipulation that occurs during the learning process. The classical systems are based on the manipulation of weight vectors, whereas the proposed algorithms manipulate or transform the actual input data entering a system.
The quality of the proposed algorithms was investigated and tested on simple one-dimensional, linearly separable classification tasks. The proposed algorithms performed well on these. Their performance actually did match the quality of the classical approaches.
The approach has a number of interesting aspects. For example, the proposed algorithms allow a single system with an initial, random configuration to learn a variety of similar, but in context completely different, problems without changing the actual system configuration at all. This makes the approach flexible and this is one aspect mentioned at the outset of this study. This aspect might be particularly interesting in a neuroinformatics context. At this stage nobody really knows how the brain really works, and therefore there is a need for new directions and proposals even if they are a bit naive at first sight.

Future work. There are a few directions into which this study can be continued. For example, an interesting route relates to the question whether it is possible to apply the approach to more complex network structures, for instance networks with input, output and hidden layers. It is also possible to invest the approach and potential consequences from a mere cognitive science and maybe philosophical position. For example, how does a system interpret an environment when the information about this environment is transformed in the learning process. We hope to engage into some of these questions in forthcoming studies.

References

[1] P. H. Koesters, *Deutschland Deine Denker*. Gruner+Jahr AG & Co, 1981.

[2] K. Mehrotra, C. K. Monan, and S. Ranka, *Elements of Artificial Neural Networks*. The MIT Press, 1997.

[3] F. Rosenblatt, "The perceptron, a probabilistic model of information storage and organization in the brain", *Psych. Rev.*, vol. 62, pp. 386–408, 1958.

[4] B. Widrow, "Generalization and information storage in networks of Adaline neurons", *Self-org. Syst.*, vol. 10, pp. 435–461, 1962.

Alfons Schuster holds a Ph.D. in computer science from the University of Ulster in Northern Ireland, and a B.Sc. in applied physics from the University of Applied Sciences Munich in Germany. His current research interests include artificial intelligence, robotics, and DNA computing. Dr Schuster has several years working experience in industry and research. He is currently employed as a lecturer at the Faculty of Informatics of the University of Ulster at Jordanstown in Northern Ireland.
e-mail: a.schuster@ulster.ac.uk
Faculty of Informatics
School of Computing and Mathematics
University of Ulster
Shore Road, Newtownabbey, Co. Antrim, BT37 0QB
Northern Ireland

Paper

The role of time in influence diagrams

Wiesław Traczyk

Abstract — An influence diagram is a compact representation emphasizing the qualitative features of decision problem under uncertainty. Classical influence diagram has parameters stable in time, determined order of suggested decisions and generally is independent of time. Here we have shown some possible methods of construction of time dependent influence diagrams: with decision ordering, time-sliced segments and time consuming nodes. Such gathering of methods can help in selection of a proper solution.

Keywords — uncertainty, belief networks, influence diagrams, ordering in time.

1. Introduction

Graphical modelling for decision support systems under uncertainty is getting more and more widespread. It is an appealing way to think of and communicate on the underlying structure of the domain in question. Graphical models are potentially powerful because they translate a complex decision problem into an easily understood, qualitative form. Quantitative, numerical solution of the problem presented in such a form is usually much more complicated, but for most typical cases there are available not only precise algorithms but also commercial systems, computing needed results.

Probabilistic graphical models are graphs in which nodes represent random variables, and arcs (or lack of them) represent conditional independence assumptions. Undirected graphical models are used to depict *Markov networks*, but directed models, enhanced with additional nodes, can describe different sceneries of decision systems.

- If some random variables from the set C describe *state of the world*, with given prior probabilities of possible values, and these variables influence some other *chance variables* from the set C' (what is noted down as $C \to C'$), the graph can be understood as the model of simple *Bayesian network* or *believe network* [1]. For chance variables we must specify the conditional probability distribution at each node. In a more complicated case chance variables can influence another chance variables: $(C,C') \to C'$. The most common task we wish to solve using Bayesian networks is probabilistic inference.

- Sometimes belief networks are controlled by external interventions, described by decision variables (from the set D). A *decision variable* is a variable whose instances correspond to possible actions among which the intervening person can choose. The model $(C,D) \to C'$ is known as *causal graph*, chance variables are called *consequences* and appropriate graphs are used for causal reasoning [2].

- Believe networks and causal graphs may help in preparing a new kind of decisions (from the set D'), which control external actions ($(C,C',D) \to (C',D')$) or influence chance variables ($(C,C',D,D') \to (C',D')$). In many cases the differences between variables from sets C and C' or from D and D' are not important, thus putting $C \cup C' = \mathcal{C}$ (called *observations*) and $D \cup D' = \mathcal{D}$ one can describe this model by $(\mathcal{C},\mathcal{D}) \to (\mathcal{C},\mathcal{D})$. This is a simple version of an *influence diagram* [3].

- Each decision support system attempts to find the best possible decision, so in a graph model we need one more type of nodes: *utility nodes* (from the set $\mathcal{U}$), that represent the usefulness of the consequences of decisions and observations, measured on a numerical scale called *utility*. The model $(\mathcal{C},\mathcal{D}) \to (\mathcal{C},\mathcal{D},\mathcal{U})$ illustrates a full version of *influence diagram* (ID), used as an analysis tool and a communication tool for decision support.

We normally assume that the model structure and the parameters of influence diagrams do not change, i.e. the model is time-invariant. However, in many cases ID is used to describe a proces containing the sequence of events, and time should be taken into account. In such situations we can add extra nodes to represent the current "regime", or we can repeat the basic diagram to represent time-slices [4].

Classical IDs require a linear temporal ordering of the decisions, and this is often felt as an unnecessary constraint. In reality some decisions can be taken independently of each other, and their identification (and order modification) can simplify system implementation [5], because the solution of a decision problem modeled by an ID is a sequence of decisions that maximizes the expected utility.

An ID specifies also a certain order of observations and decisions through its structure. This order is reflected in the corresponding methodology of solving ID [6].

The problems specified above confirm the significant role of time in ID. Some of these problems will be considered below in detail, but for that more precise definitions are needed.

2. Influence diagrams

An influence diagram is a directed acyclic graph $I = (\mathcal{V}, \mathcal{E})$, where the nodes (vertices) $\mathcal{V}$ can be partitioned into three disjoint subsets: *chance nodes* $\mathcal{C}$, *decision nodes* $\mathcal{D}$ and *utility* (*value*) *nodes* $\mathcal{U}$, thus $\mathcal{V} = \mathcal{C} \cup \mathcal{D} \cup \mathcal{U}$. It is a common practise to term nodes and variables in ID by the same name and use them interchangeably.

The chance nodes (drawn as circles or ovals) correspond to *chance variables*, and represent events which are not under the direct control of the decision maker. The decision nodes (drawn as squares) correspond to *decision variables* and represent actions under the direct control of the decision maker. The utility nodes (drawn as diamonds) define *utility functions*, indicating the local utility for a given configuration of variables in their domain. The total utility is the sum or the product of the local utilities.

The arcs (links) in an ID (the pairs of nodes (x,y) from the set $\mathcal{E}$) can be partitioned into three disjoint subsets, corresponding to the type of node they go into. Arcs into utility nodes represent functional dependencies by indicating the domain of associated utility function. Arcs into chance nodes, denoted *dependency arcs*, represent probabilistic dependencies. Arcs into decision nodes, denoted *informational arcs*, imply information precedens: if there is an arc from a node x to a decision node d then the state of x is known when decision d is made.

If there is a directed link from x to y $(x,y \in \mathcal{V})$, then x is called a *parent* of y, and y a *child* of x. The sets of parents and children of x are denoted $pa(x)$ and $ch(x)$, respectively. For each utility node u the set $ch(u)$ is empty. In an ID we usually assume "no forgetting", which means that if there is a link from x to d we need not have a link from x to elements of $ch(d)$.

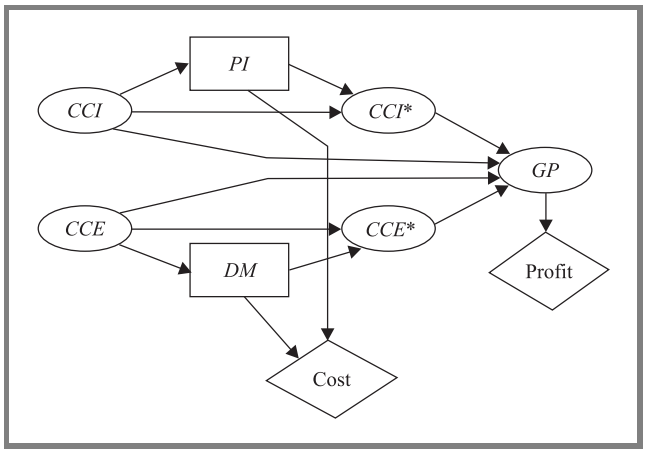


Fig. 1. Influence diagram of call center.

Simplified example of influence diagram for call center (CC) is depicted in Fig. 1. The task of this call center is to promote voice mail delivered by a telecommunication company. Call center intensity *CCI* and efficiency *CCE*, defined by known probability distribution of informative variables, influence the global performance *GP* of CC. This performance can be enlarged by external intervention, changing intensity and efficiency, namely personel increase *PI* and/or special algorithm of data mining *DM*.

Utility nodes state the *costs* of interventions (negative value) and the *profit* given by enlarged global performance (positive value). Optimal decision is determined on the basis of the sum of these utilities.

With each chance variable and decision variable x we associate a *state space* W_x which denotes the set of possible outcomes/decision alternatives for x [5]. For a set X of variables we define the state space as $W_X = \mathrm{X}\{W_x | x \in X\}$.

The uncertainty associated with each chance variable r is represented by a *conditional probability function* $P(r|pa(r)) : W_{\{r\} \cup pa(r)} \to [0,1]$.

When evaluating an ID we identify a *strategy* for the decision variables; a strategy can be seen as a prescription of responses to earlier observations and decisions. A strategy is then a set of functions $\Delta = \{\delta_d | d \in \mathcal{D}\}$, where δ_d is a decision function given by:

$$\delta_d : W_{pa(d)} \to W_d .$$

The evaluation is usually performed according to the *maximum expected utility principle*, which states that we should always choose an alternative that maximizes the expected utility. A strategy that maximizes the expected utility is termed an *optimal strategy*. It strongly depends on temporal ordering of variables and therefore time considerations are so important.

3. Time in influence diagrams

There are at least three particular cases when the value of time or ordering in time play important role.

Time is not explicitly declared. If the decision problem, modeled by an influence diagram, has not periods of time clearly stated, a diagram is constructed sequentially: at first chance variables and dependency arcs between them are introduced, then decision variables with information links are added, and next utility nodes are defined and connected with other nodes. The ID is ready, but before using it we have to term its *realization*: an attachment of functions to the appropriate variables. This means that the chance nodes and variables are associated with conditional probability functions (or prior probabilities for nodes without parents) and the utility nodes and variables are associated with utility functions. Decision nodes correspond to actions taken by external agents; ID defines information needed for each decision (by information arcs) and, sometimes, the order of decisions. An order depends on the structure of the ID and its interpretation.

A directed path $\pi = \langle x_1, x_2, \ldots, x_k \rangle$ in ID is an ordered sequence of distinct nodes such that $x_i \in pa(x_{i+1})$. The set $an(y)$, called *ancestors of y*, contains all nodes x such

that there exist a path $\langle x,\ldots,y\rangle$. For ordering in time we will use notation $x \prec y$ (x before y).
It will be assumed that:

1. If there is the path from a node x to a node y then a variable x and all elements of this path are relevant for a variable y, i.e. values or decisions of y are functions of all values and decisions from the path realization. This is the simplifying assumption because in reality:
   - some values of the probability distribution can block the influence of some other values from the path,
   - some values from the path are not required for an optimal strategy, i.e. are not elements of the maximum expected utility function [5].

   Nevertheless this assumption is sensible because it helps to order in time events defined by ID in more general case, when probability distributions and utility functions are modified during the project preparation.
2. Actions represented by decision nodes cannot be taken simultaneously and therefore different nodes d_i should be related to different times t_i.
3. All parameters are stable in time.

If in the ID one can find a path containing all decision nodes then an order of decisions is forced by this path, and each earlier decision influences future decisions. If decision nodes are located in distinct paths–their ordering in time can be changed and we call them *incompatible* nodes.

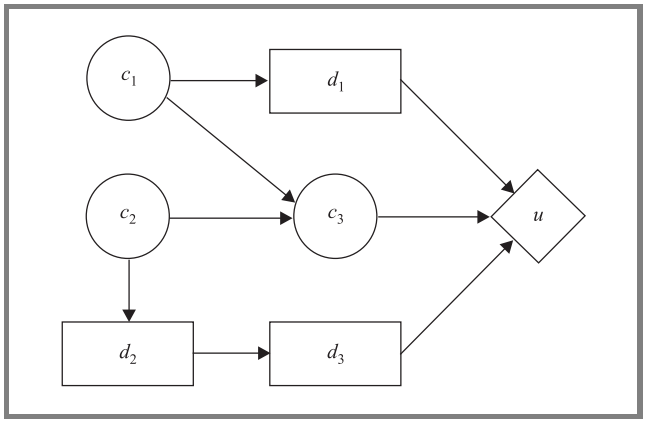


Fig. 2. Influence diagram with incompatible nodes.

Let two decision nodes belong to the set of parents of some utility nodes: $\{d', d''\} \in pa(u)$, i.e. nodes d' and d'' are incompatible. If $d' \prec d''$ then utility function of u may be partially or totally satisfied by first decision d', what influence a future decision d''; thus order is important. Since all nodes from $an(d')$ influence d'' (assumption 1), they also influence d''. In the case of two sequences with the paths of chance nodes α, β: $\langle d', \alpha, u\rangle$ and $\langle d'', \beta, u\rangle$ situation is the same if $\alpha = \beta$, but if $\alpha \neq \beta$ then real order of decisions influences total utility and depends on a delay introduced by α and β. This case will be discussed below.

The conclusion from these considerations is as follows: if $\{d', d''\} \in an(u)$ and $d' \prec d''$ and delays can be neglected then d' and all its ancestors have an impact on d''.

In the example from Fig. 2 the path $\langle c_2, d_2, d_3, u\rangle$ means that $d_2 \prec d_3$, and distinct path $\langle c_1, d_1\rangle$ means that node d_1 is incompatible with nodes d_2, d_3. If $d_3 \prec d_1$ then the nodes c_1, c_2, c_3, d_2, d_3 influence decision d_1, but if $d_1 \prec d_2$ then c_1, d_1 have an impact on remaining nodes.

Diagrams are time-sliced. The definition of ID and known solution algorithms assume that all parameters of ID have a static nature. If we want to use an influence diagram for modelling a system with uncertain states which alter in time, we must repeat basic structure of ID and relate each instantiation with distinct moment of time. From the basic structure $\mathcal{I}$ of ID one can construct a chain $\langle(\mathcal{I}_0, t_0), (\mathcal{I}_1, t_1), \ldots, (\mathcal{I}_k, t_k),\rangle$ of links $\mathcal{I}_i$, with similar structures. Usually each node $x_{j,0}$ from $\mathcal{I}_0$ has similar nodes $x_{j,1}, x_{j,2}, \ldots$ in remaining segments of ID, but some parameters of these nodes are different, simulating parameter changes in discrete time, with a characteristic $\langle x_{j,0}, x_{j,1}, \ldots, x_{j,k}\rangle$. Segments of such time-sliced diagram are connected by arcs (*temporal links*) which define how the distribution of time slice i depends conditionally on the distribution of the variables of time slice $i-1$. The time slices of ID are assumed to be chosen such that the ID obeys the Markov property: the future is conditionally independent of the past given the present.

Figure 3 depicts very simple structure of bit-sliced ID, with segments corresponding to weeks (in a month) or to quarters (in a year).

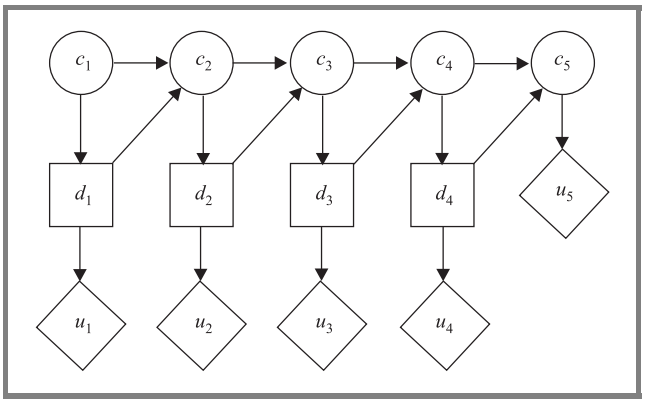


Fig. 3. Bit-sliced influence diagram.

Chance nodes may refer to the states of production (business, health etc.) during a considered period, decision nodes show external interventions improving appropriate state at the end of the period, and utility nodes valuate the results (periodic and final).

One of the segments (for example $\mathcal{I}_x$) of a time-sliced ID can refer to the current time t_x. If all parameters up to t_x are observed then inference of parameters from $\mathcal{I}_{x+1}, \mathcal{I}_{x+2}, \ldots$ is known as *prediction*. On the basis of information from $\mathcal{I}_x$ one can also compute some parameters from the past (*hindsight*).

Time-sliced ID helps in choosing the best moment for decision taking; comparing effects of the same decision in different slices one can select the most suitable time.

Time is related to nodes. Chance nodes may describe states or events with substantial length of time. Decision nodes are also related to time consuming actions, and results of decisions are sometimes very late towards the moment of decision. If in a modeled problem time is critical, all these delays should be taken into account.

Let $\tau(x)$ describe the time needed by a node x and $\tau(\pi)$ – the time necessary for a path π. When $\tau\langle d', \alpha\rangle > \tau\langle d'', \beta\rangle$ and both paths give the same result with equal costs, the decision d'' should be preferred. All other cases can be discussed easily.

As an example of these considerations we will discuss the case of the call center from the Section 1. In this case the global performance of CC can be enlarged by personel increase or the new algorithm of data mining, or both. If the time of performance improvement is critical, the total utility function depends on the times of two actions (personel recruiting and education versus algorithm preparation and implementation).

There are some other methods for introducing an impact of time in ID.

- When a value of time is uncertain we can use a special chance node to represent it, and utilize for further inference.
- If delay of action related with d decreases utility u, we can model losses with time by the linear or exponential form of utility functions [7]:

$$u(d,t) = u(d,t_0) - at\,, \qquad u(d,t) = u(d,t_0)e^{bt}\,.$$

4. Summary

We described the assessment and use of time dependent influence diagrams. It has been shown that there are many opportunities to introduce time: by ordering nodes, time slices, time dependent variables and functions. Unfortunately different approaches are devoid of common methodology helping in an ID construction. It seems that further investigations should be directed to the integration of existing methods.

References

[1] F. V. Jensen, *An Introduction to Bayesian Networks*. UCL Press, 1998.

[2] D. Heckerman and R. Shachter, "Decision-theoretic foundations for causal reasoning", *J. Artif. Intell. Res.*, no. 3, 1995.

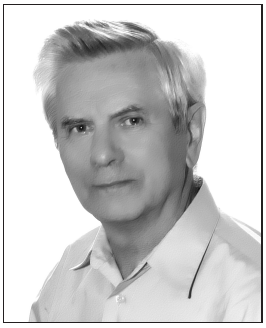
[3] R. A. Howard and J. E. Matheson, "Influence diagrams" in *The Principles and Applications of Decision Analisis*, R. A. Howard and J. E. Matheson, Eds., SDG, 1984, vol. II.

[4] S. L. Lauritzen and D. Nilsson, "Representing and solving decision problems with limited information", *Manag. Sci.*, vol. 47, no. 9, 2001.

[5] T. D. Nielsen and F. V. Jensen, "Welldefined decision scenerios" in *Proc. 15th Conf. Uncert. Artif. Intell.*, Stockholm, Sweden, 1999.

[6] S. L. Dittmer and F. V. Jensen, "Myopic value of information in influence diagrams" in *Proc. 13th Conf. Uncert. Artif. Intell.*, Providence, USA, 1997.

[7] E. Horvitz and G. Rutledge, "Time-dependent utility and actions under uncertainty" in *Proc. 7th Conf. Uncert. Artif. Intell.*, Los Angeles, USA, 1991.



Wiesław Traczyk is a Professor of the National Institute of Telecommunications and also of Warsaw University of Technology, Institute of Control and Computation Engineering. His research interests include expert systems, approximate reasoning, failures in computer networks and data mining.

e-mail: W.Traczyk@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

e-mail: traczyk@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Paper

Briefly on the GUHA method of data mining

Petr Hájek

Abstract — The paper gives brief, user-oriented, information on the GUHA method.

Keywords — GUHA method, data mining, exploratory data analysis.

GUHA is a method of computerized generation of hypotheses based on given data. GUHA stands for General Unary Hypotheses Automaton. Its origin goes to mid-sixties of XX century, original authors are P. Hájek, I. Havel and M. Chytil [7]. The basic monograph on GUHA and its theory is P. Hájek and T. Havránek [9]. With time the method has undergone continuous development, both in its theory and implementations.

The main principle of GUHA, formulated in [7] is to describe all the possible assertions which may be hypotheses, to generate them in some optimal manner, to verify each such assertion and to output those found interesting ("interesting" meaning: supported by data and sufficiently "strong"). "*The function of GUHA is to offer hypotheses, not to verify previously formulated hypotheses.*"

This makes GUHA a method of *exploratory data analysis* (as opposed to confirmatory data analysis); since the 90-ties of the 20th century the term "*data mining*" has been in use for such methods (mainly if they deal with very large data sets, see [1, 2]). From the present point of view, GUHA is a very early (and still developing) method of data mining, which, unfortunately, has been rather unknown. Here we describe GUHA very briefly, giving reference to the literature and implementations.

The *data* to be processed can be represented as a rectangular matrix **M** whose rows correspond to some objects and columns to some attributes. (For example, objects may be patients and attributes are symptoms and diseases; or objects are bank transactions and attributes are various items describing them as kind of loan etc. – note that this terminology – transactions and items – is standard in the methods of Agrawal *et al.*). The value in the ith row and jth column is the value of jth attribute for the ith object. The value may be binary (yes-no, coded by 1 and 0), integers or reals, the "classical" case being the first. Missing values can be handled.

Logic is used to code hypotheses. Give each attributes (matrix column) a name (e.g. SEX, AGE, ...). For any subset X of the domain of an attribute P, $P:X$ is the property saying "the value of P is in X, e.g.

$$TEMPERATURE : (\geq 38^{\circ}C).$$

If P is binary (e.g. HAS−A−CAR) then P stands for $P =$ YES and $\neg P$ (negation) for $P =$ NO. The formula $P:X$ is called a *literal* (atomic formula). You may form conjunctions of literals, e.g. $P_1 : X_1 \,\&\, P_3 : X_3 \,\&\, P_7 : X_7$ is a conjunction of three literals. The ith object satisfies this conjunction if its value (in the ith row of the data matrix) of P_1 is in X_1, value of P_3 is in X_3 and value P_7 is in X_7. Such conjunctions describe *composed properties* of our objects.

Hypotheses in GUHA (more precisely in its particular version called GUHA-assoc) have, roughly, the form "properties φ, ψ are associated" (think, for example, of smoking and cancer). We write $\varphi \sim \psi$ and here $\sim$ denotes some notion of association (logically speaking, a generalized quantifier). φ is called the *antecedent* and ψ the *succedent* of the sentence (hypothesis) $\varphi \sim \psi$.

The pair φ, ψ of properties and the given data matrix determine four frequencies:

$a =$ the number of objects satisfying both φ and ψ,
$b =$ the number of objects satisfying φ but not ψ,
$c =$ the number of objects satisfying ψ but not φ,
$d =$ the number of objects satisfying neither φ nor ψ.

This can be presented as a *four-fold table*

a	b	r
c	d	s
k	l	m

where $r = a+b$, $s = c+d$, $k = a+c$, $l = b+d$ (marginal sums) and $m = a+b+c+d = r+s = k+l$ is the number of objects in our data.

A *quantifier* (notion of association) $\sim$ is given by a function $Tr_{\sim}(a,b,c,d)$ associated with each four-fold table a,b,c,d either 1 (yes) or 0 (no); the formula $\varphi \sim \psi$ is *true* in the data if and only if for the four-fold table (a,b,c,d) of φ, ψ we get $Tr_{sim})(a,b,c,d) = 1$.

We shall give *four examples of quantifiers* used in GUHA (those are not all quantifiers GUHA uses).

(1) Implicational ($A \Rightarrow B$ says "Many A's are B's")
FIMPL (founded almost-implication): It has value 1 when

$$a \geq BASE \text{ and } \frac{a}{a+b} \geq p$$

($BASE, p$ are parameters)

LIMPL (lower critical almost-implication): Parameters $BASE, p, \alpha$. It has value 1 when $a \geq BASE$ and

$$\sum_{i=a}^{r} \binom{r}{i} \cdot p^i \cdot (1-p)^{r-i} \leq \alpha.$$

(2) Comparative associational ($A \sim B$ says "B is rather more frequent among A's than among $(\neg A)$'s" (or, if you want, "A makes B more plausible".

SIMPLE (simple deviation): Parameters: *BASE* (integer), $K \geq 1$. It has value 1 when $a \geq BASE$ and $a \cdot d > K \cdot b \cdot c$. (Note that the second formula is equivalent to

$$\frac{a}{a+Kb} > \frac{c}{c+d}).$$

Fisher quantifier: Parameters $BASE, \alpha$. It has value 1 when $a \geq BASE$, $a \cdot d > b \cdot c$ and

$$\sum_{i=0}^{\min(b,c)} \frac{(a+b)!(a+c)!(b+d)!(c+d)!}{m!(a+i)!(b-i)!(c-i)!(d+i)!} \leq \alpha.$$

In both groups, the first quantifier (FIMPL, SIMPLE) just expresses an observation on frequencies in the data; the second, more complicated, is a test of a statistical hypothesis concerning probabilities in the unknown universe from which our data are a sample. (High conditional probability in the case of LIMPL, positive dependence in the case of FISHER).

Note that the GUHA theory has a notion of associational and implicational quantifiers; FIMPL and LIMPL are examples of implicational quantifiers whereas SIMPLE and FISHER are examples of associational quantifiers that are not implicational (conversely, each implicational quantifier is associational). We do not go into any details; see [9] for this theory.

Let us also mention that FIMPL is almost the same notion of association as what Agrawal and his school calls "association rule", reinventing in fact our FIMPL (the only difference is that our BASE gives a lower bound for a whereas his "support" gives a lower bound to a/m, where m is the number of objects). Also note that examples of other quantifiers are found in [9] and in the papers by Rauch [21–25].

Now we shortly describe a GUHA procedure ASSOC working with associational quantifiers. The application of the procedure takes place in three steps:

- preprocessing – input of the data matrix and parameters determining syntactic form of antecedents and succedents to be generated (e.g. maximal length of conjunctions, attributes allowed to occur only in antecedents, in the succedents; choice of the quantifier used and its parameters etc.); preparing the internal representation of the data matrix in a form suitable for quick generation and evaluation of hypotheses;
- kernel – generating and evaluating hypotheses; a system of "interesting" ones is produced;
- postprocessing – browsing in the hypotheses found and their interpretation; sorting, reordering etc.

There have been several implementations in the history of GUHA; but the reader will agree that implementations get obsolete very quickly. There is a working implementation for PC [15] and two present implementations under Windows, freely available: GUHA+- and 4ft-Miner [28, 29]. The references below contain a selection of works devoted to GUHA theory and selected applications. *Don't overlook* the fact that the basic monograph on GUHA theory, [9] is now freely available on the web [10].

The antiquity of GUHA is not the most important thing; more important is the fact that the theory developed for GUHA is also presently useful and inspirative for data mining. One can hope that it will become more broadly known in the future.

Acknowledgement

Partial support by the COST action 274 TARSKI is acknowledged.

References

[1] R. Agrawal, H. Manilla, R. Sukent, A. Toivonen, and A. Verkamo, "Fast discovery of association rules" in *Advance in Knowledge Discovery and Data Mining*. AAA Press, 1996, pp. 307–328.

[2] J. M. Adamo, *Data Mining for Associational Rules and Sequential Patterns, Sequential and Parallel Algorithms*. Springer, 2001.

[3] D. Coufal, "GUHA analysis of air pollution data" in *Proc. Int. Conf. Artificial Neural Nets and Genetic Algorithms ICANNGA'2001*, V. Kůrková, N. C. Steele, R. Neruda, and M. Kárný, Eds., Prague, Czech Rep., 2001. Springer, 2001, pp. 465–468.

[4] T. Feglar, "The GUHA architecture" in *Proc. Relmics 6*, Tilburg, The Netherlands, 2001, pp. 358–364.

[5] P. Hájek, "The GUHA method and mining association rules" in *Proc. CIMA'2001*, Bangor, Wales, 2001, pp. 533–539.

[6] P. Hájek, "The new version of the GUHA procedure ASSOC" in *Proc. COMPSTAT*, Physica-Verlag, 1984, pp. 360–365.

[7] P. Hájek, I. Havel, and M. Chytil, "The GUHA method of automatic hypotheses determination", *Computing*, no. 1, pp. 293–308, 1966.

[8] P. Hájek, K. Bendová, and Z. Renc, "The GUHA method and three-valued logic", *Kybernetika*, no. 7, pp. 421–431, 1971.

[9] P. Hájek and T. Havránek, *Mechanizing Hypothesis Formation (Mathematical Foundations for a General Theory)*. Springer, 1978.

[10] P. Hájek and T. Havránek, *Mechanizing Hypothesis Formation (Mathematical Foundations for a General Theory)*. Internet edition, http://www.cs.cas.cz/∼hajek/guhabook/

[11] P. Hájek and M. Holeňa, "Formal logics of discovery and hypothesis formation by machine", *Theor. Comput. Sci.*, vol. 292, pp. 345–357, 2003.

[12] P. Hájek, J. Rauch, T. Feglar, and D. Coufal, "The GUHA method, data preprocessing and mining", in *Proc. Database Technologies for Data Mining DTDM'02*, Praha, Czech Rep., 2002, pp. 29–36.

[13] P. Hájek, Guest Ed., *Int. J. Man-Mach. Stud.*, vol. 10, no. 1, 1977 (special issue on Guha). Introductory paper, P. Hájek and T. Havránek, "The GUHA method – its aims and techniques", *Int. J. Man-Mach. Stud.*, no. 10, pp. 3–22, 1977.

[14] P. Hájek, Guest Ed., *Int. J. Man-Mach. Stud.*, vol. 15, no. 3, 1981 (second special issue on GUHA).

[15] P. Hájek, A. Sochorová, and J. Zvárová, "GUHA for personal computers", *Comp. Stat. Data Anal.*, no. 19, pp. 149–153, 1995.

[16] J. Hálová and P. Žák, "Coping discovery challenge of mutagenes discovery with GUHA+/- for windows" in *Int. Workshop on KDD Challenge on Real-world Data*, Kyoto, Japan, 2000, pp. 55–60; *Pacific-Asia Conf. Knowledge Discovery and Data Mining*, Kyoto, Japan, 2000.

[17] T. Havránek, "The statistical modification and interpretation of GUHA method", *Kybernetika*, no. 7, pp. 13–21, 1971.

[18] M. Holeňa, "Fuzzy hypotheses for GUHA implications", *Fuzzy Sets Syst.*, vol. 98, pp. 101–125, 1998.

[19] M. Holeňa, "Exploratory data processing using a fuzzy generalization of the GUHA approach" in *Fuzzy Logic*, Baldwin *et al.*, Eds., New York: Willey, 1996, pp. 213–229.

[20] L. Pecen, E. Pelikán, H. Beran, and D. Pivka, "Short-term fx market analysis and prediction" in *Neural Networks in Financial Engeneering*, A. P. Refenes, J. Moody, and Y. Abu-Mustafa, Eds., *World Scientific*, 1995, pp. 189–196.

[21] J. Rauch, "GUHA as a data mining tool, practical aspects of knowledge management". Schweizer Informatiker Gesellshaft Basel, 1996.

[22] J. Rauch, "Logical calculi for knowledge discovery" in *Principles of Data Mining and Knowledge Discovery*, J. Komorowski and J. Zytkow, Eds. Berlin: Springer, 1997, pp. 47–57.

[23] J. Rauch, "Logical problems of statistical data analysis in databases" in *Proc. Eleventh Int. Sem. Database Manag. Syst.*, 1988, pp. 53–63.

[24] J. Rauch and M. Šimůnek, "Mining for 4ft association rules" in *Proc. Discovery Science*, 2000, Kyoto. Springer, 2000, pp. 268–272.

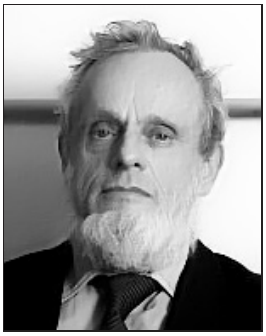
[25] J. Rauch and M. Šimůnek, "Mining for statistical association rules" in *Proc. PAKDD*, Hong Kong, 2001, pp. 149–158.

[26] V. Šebesta V and L. Straka, "Determination of suitable markers by the GUHA method for the prediction of bleeding at patients with chronic lymphoblastic leukemia" in *Mediter. Conf. Med. Biolog. Eng. Comput. Medicon 98*, Lemesos, Cyprus, 1998.

[27] J. Zvárová, J. Preiss, and A. Sochorová, "Analysis of data about epileptic patients using GUHA method" in *EuroMISE 95: Information, Health and Education*, J. Zvárová and I. Malá, Eds., Prague, Czech Rep., 1995; *TEMPUS Int. Conf.*, Prague, Czech Rep., 1995.

[28] "GUHA+– project web site", http://www.cs.cas.cz/ics/software.html

[29] LISp-Miner overview, http://lispminer.vse.cz/overview/4ftminer.html



Petr Hájek graduated in mathematics in Charles University in Prague. He obtained Ph.D. and D.Sc. degrees in mathematical logic in the Czechoslovak Academy of Sciences. In 1998 he obtained professorship in mathematics (awarded by the president of the Czech Republic). He worked in Mathematical Institute of the Czechoslovak Academy of Sciences (now Academy of Sciences of the Czech Republic) and then became a director of the Institute of Computer Science of AS CR, where he is now a senior scientist. He is an author and co-author of 6 monographs and more than 200 scientific papers in the field of mathematical logic. Member of the Association for Symbolic Logic and of editorial boards of *Archive for Mathematical Logic*, *Fundamenta Informaticae*, *Soft Computing* and *Czechoslovak Mathematical Journal*.

e-mail: hajek@cs.cas.cz
Institute of Computer Science
Academy of Sciences
Pod vodarenskou vezi 2
182 07 Prague, Czech Republic

Paper

Data mining and complex telecommunications problems modeling

Janusz Granat

Abstract — The telecommunications operators have to manage one of the most complex systems developed by human beings. Moreover, the new technological developments, the convergence of voice and data networks and the broad range of services still increase this complexity. Such complex object as telecommunication network requires advanced software tools for their planning and management. Telecommunications operators collect large volumes of the data in various databases. They realize that the knowledge in these huge databases might significantly improve various organizational strategic and operational decisions. However, this knowledge is not given explicitly, it is hidden in data. Advanced methods and algorithms are being developed for knowledge extracting. In this paper we will focus on using data mining for solving selected problems in telecommunication industry. We will provide a systematic overview of various telecommunications applications.

Keywords — decision support systems, telecommunications, dynamic information system, temporal data mining.

1. Introduction

The problems that are specified in the domain terms might be classified into three main levels of analysis (Fig. 1):

- *Business level* (e.g. better understanding and prediction of customer behavior, identification of customer needs, customer-oriented supply of new services, improvement of business processes). On this level we use a client oriented data.
- *Product or service level* (e.g. web mining). On this level we use service oriented data.
- *Network and information infrastructure analysis level* (e.g. fault detection, supporting network management, resource planning). On this level we use a network oriented data.

We can distinguish three main steps of describing data mining problems:

1. Problem formulation in the domain terms. This is usually textual description of the business requirements that have to be fulfilled by data mining.
2. The transformation of business requirements into a class of data mining problems like classification, prediction, associations etc. It is a bridge between business description and detailed model specification.
3. The detailed model specification. This is a model specification that is used by data mining modeler for a specific software tools.

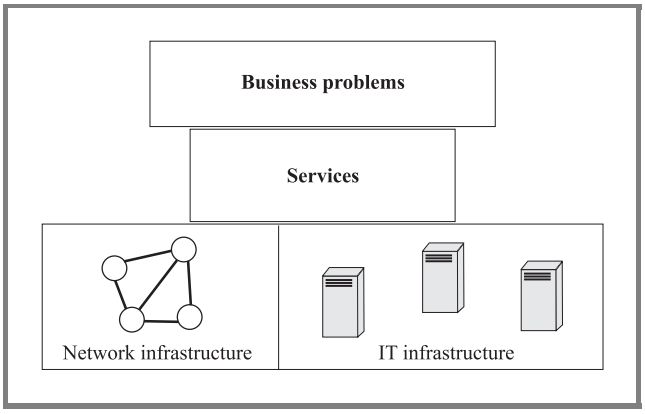


Fig. 1. Levels of problem analysis.

An overview of data mining problems in the context of business problems in telecommunication is given in [1, 5]. It can be observed that one of the main areas of applications of data mining on business level is a support for various task of the marketing departments. The data mining becomes a key part of analytical subsystem of customer relationship management systems. On business level of analysis there are many similarities to other industries. The applications of data mining for marketing can be found in [11]. The following main problems for marketing and sales departments of telecommunication operators can be distinguished:

– customer segmentation and profiling,

– churn prediction,

– cross selling and up-selling,

– live-time value,

– fraud detection,

– identifying the trends in customer behavior.

On product or service level there is a focus on analysis of incomes, quality of the service, grade of the service and others. There are formal agreements called service level agreements (SLA) [4] between providers of the service and the customers. Service level management (SLM) are becoming the prevailing business model for delivering

a products and services. Such approaches need advanced computerized tools.

On the level of infrastructure and network analysis we can distinguish the following problems:

– network planning,

– IT resources planning,

– fault detection, location and identification.

2. The formal description of a data mining process

The typical data mining process consist of the following steps:

– problem formulation,

– data preparation,

– model building,

– interpretation and evaluation of the results.

In the industry environment these steps as follows (in the brackets there is information about responsible persons):

– problem formulation (business users),

– developing programs for preprocessing the data (data mining analyst),

– building the model (data mining analyst, business users),

– prepare the processes of the use of the data mining models in the business (business users),

– repetitive running of the model (data mining analyst),

– running programs for loading and transformation of the data,

– running the data mining models – scoring,

– export the scoring results to the operational systems.

There are a lot of publications related to data mining but these publications are focusing on algorithms, description of problems etc. but there is no common formal description of data mining process in the context of enterprise application. In this section we will provide such a formal description of a data mining process. We will start with source data description by information system, then preprocessing of data in order to prepare input for data mining algorithms, and finally the results of the algorithms.

2.1. Source information systems

As the input for a data mining process there are various tables of the databases, text files etc. These source data might be described formally by *the information systems*. We define, following [8, 9] or [3], an *information system* as a 4-tuple:

$$S = (X, A, V, \rho), \tag{1}$$

where:

X – is the finite and nonempty set of objects or observations,

A – is finite and nonempty set of attributes,

$V = \bigcup_{a \in A} V_a$, V_a is a set of values of attribute $a \in A$, called the domain of a,

ρ – is an information function: $\rho : A \times X \to V$.

Information system S define a relation $R_s \subset V_{a_1} \times V_{a_2} \times \times \ldots \times V_{a_k}$, so that $R_s(v_{i_1}, v_{i_2}, \ldots, v_{i_k}) \Leftrightarrow (a_1, v_{i_1}), (a_2, v_{i_2}), \ldots, (a_k, v_{i_k})$ is nonempty information in S. The relational approach is often used in data processing, but in data mining we need more information that we have in information system. The links between information system and relations might be useful in data preprocessing.

The information system (1) describes the static nature of the system. In practical applications we have to deal with dynamics of the system. Orłowska [7] introduced the term *dynamic information system*:

$$D = (X, A, V, \rho, T, R), \tag{2}$$

where:

T – is a nonempty set whose elements are called moments of time,

R – is a order on the set T (here we assume linear order),

X – is the finite and nonempty set of objects or observations,

A – is finite and nonempty set of attributes,

$V = \bigcup_{a \in A} V_a$, V_a is a set values of attribute $a \in A$, called the domain of a,

ρ – is an information function: $\rho : A \times X \times T \to V$.

Orłowska in [7] have considered dynamic information system in context of a logic. In this paper, we wiil use this system as a base for formulation of the temporal data mining problems.

2.2. The preprocessing

The data sources of a data mining process might be described by the set of dynamics information systems:

$$\Sigma = \{D^1, D^2, \ldots, D^l\}. \tag{3}$$

The data sources have to be transformed into the input dynamic information system IT that is needed for data mining models:

$$IT = \mathcal{P}(\Sigma),$$

where:
IT – is an input dynamic information system,
Σ – is a set of source information systems,
$\mathcal{P}$ – is a process of preprocessing.

The IT is defined as:

$$IT = (X, F, V, \rho, T, R), \tag{4}$$

where:
T – is a nonempty set whose elements are called moments of time,
R – is a order on the set T (here we assume linear order),
X – is the finite and nonempty set of objects or observations,
F – is finite and nonempty set of features of the objects,
$V = \bigcup_{f \in F} V_f$, V_f is a set values of feature $f \in F$, called the domain of f,
ρ – is an information function: $\rho : F \times X \times T \to V$.
A process of preprocessing can be defined by the set of preprocessing steps (Fig. 2). The preprocessing step can be defined as:

$$N^i = (PN^i, SN^i, ID^i, ON^i, OD^i),$$

where:
N^i – ith preprocessing step,
PN^i – the set of steps that are the predecessors of the step N_i,
SN^i – the set of successors of the step N_i,
ID^i – the set of input dynamic information systems for the step N_i,
ON^i – the operator of the step,
OD^i – the set of output dynamic information systems of the step N_i.
The dynamic information system $IT \equiv OD^i_j$ for selected step N^i. ON^i belong to set of operators:

$$\mathcal{O} = (O_1, O_2, \ldots, O_k).$$

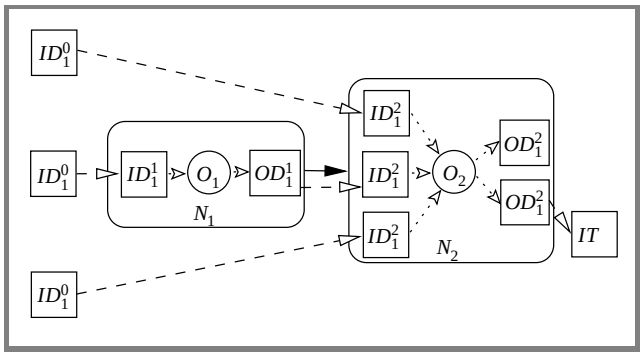


Fig. 2. A process of preprocessing – an example.

We might have the basic sets of operators on physical level like: projection, selection, etc. However, the preprocessing phase requires a broad knowledge about the data and methods of data transformation. In data mining we need advanced systems for preprocessing that will allow to store and reuse the knowledge about this phase. MiningMart [6] is an example of the system dedicated to preprocessing.

2.3. Modeling – the model building

After execution of the preprocessing step we have an dynamic information system IT that might be used for building a model. A model might have various forms. We can write that model $\mathcal{M}$ is build on the base of the dynamic information system IT:

$$IT \Rightarrow \mathcal{M}.$$

In this paper we restrict our models to feature based models. Feature base modeling assumes that objects are described by a set of features and the models find dependencies between features or predict unknown values.
We have to define the training, test, evaluation and scoring dynamic information systems. These information systems are equivalent to the sets defined in [2]. The training dynamic information system is used for preliminary model building. The test dynamic information system is used for refining the model. The performance of the model is tested by using evaluation dynamic information system. The model is applied to the score dynamic information system (Fig. 3).
The training, test, evaluation and scoring information systems are defined as follows:

$$IT^{\{id\}} = (X^{\{id\}}, F^{\{id\}}, V^{\{id\}}, \rho^{\{id\}}, T^{\{id\}}, R), \tag{5}$$

where:
id = Train – for a training dynamic information system,
id = Test – for a test dynamic information system,
id = Eval – for an evaluation set dynamic information system,
id = Score – for a scoring set dynamic information system,
$T^{\{id\}}$ – is a nonempty set whose elements are called moments of time,
R – is a order on the set T (here we assume linear order),
$X^{\{id\}} \subset X$ – is the finite and nonemty set of objects or observations,
$F^{\{id\}} = F$ – is finite and nonempty set of features of the objects,
$V^{\{id\}} = V = \bigcup_{f \in F} V_f$, V_f is a set values of feature $f \in F$, called the domain of f,
$\rho^{\{id\}}$ – is an information function:
$\rho^{\{id\}} : F \times X^{\{id\}} \times T \to V$.

The sets of objects fulfill the following condition:

$$X = X^{Train} \cup X^{Test} \cup X^{Eval} \cup X^{Score}.$$

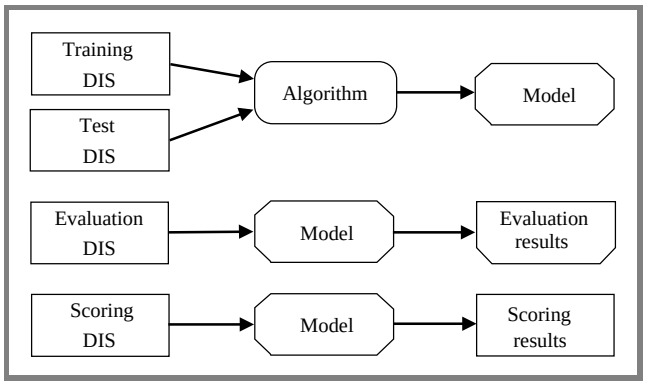


Fig. 3. The main components of a data mining process (DIS is a dynamic information system).

2.4. The data mining models

2.4.1. The classification models

There is a set of predefined m classes of the objects:

$$C = \{C_1, C_2, \ldots, C_m\}.$$

These classes divide the set X^{id} into m subsets:

$$X^{id} \mapsto \{X^{id}_{C_1}, X^{id}_{C_2}, \ldots, X^{id}_{C_m}\},$$

$$X^{id}_{C_i} \subset X^{id}, X^{id}_{C_i} \cap X^{id}_{C_j} = \emptyset \quad \text{for} \quad i \neq j, \quad \bigcup_i X^{id}_{C_i} = X^{id}.$$

The classification model assigns for each object its category. Let us consider the selected f_C feature of the object (where $C \in FC$, FC – is the index of a feature that identify the class), called class feature, and the subsets of input features f_I ($I \in FI$, $FI = F \setminus FC$, FI – is the index set of input features, F – is the index set of all object features).

The model is defined as follows:

$$\rho(f_C, x_i, t) = M_C(\rho(f_{k1}, x_i, t), \rho(f_{k2}, x_i, t), \ldots, \rho(f_{kk}, x_i, t)),$$

where:
x_i – is an object identifier,
$t \in T^{id}$ – is a moment of time,
$k1, k2, \ldots kk \in FI$.

2.4.2. The clustering based models

There is a set of objects X^{id} of a dynamic information system IT^{id} and the similarity measure between objects $x_i, x_j \in X^{id}$, $i \neq j$:

$$\varphi(x_i, x_j).$$

The clustering algorithms divide the set of objects into m subsets of similar objects (based on the similarity measure):

$$X^{id} \mapsto_{\varphi(x_i, x_j)} \{X^{id}_{S_1}, X^{id}_{S_2}, \ldots \cup X^{id}_{S_m}\},$$

$$X^{id}_{S_i} \subset X^{id}, X^{id}_{S_i} \cap X^{id}_{S_j} = \emptyset \quad \text{for} \quad i \neq j, \quad \bigcup_i X^{id}_{S_i} = X^{id}.$$

Each of the clusters has the corresponding identifier:

$$S = \{S_1, S_2, \ldots, S_m\}.$$

For a huge data set we have to find the clusters of objects for a training set and then we build a classification model that can be applied for a scoring set. Let us consider the selected f_S feature of the object (where $S \in FS$, FS – is the index of cluster feature), called cluster feature, and the subsets of input features f_I ($I \in FI$, $FI = F \setminus FC$, FI – is the index set of input features, F – is the index set of all object features).

The model is defined as follows:

$$\rho(f_S, x_i, t) = M_S(\rho(f_{k1}, x_i, t), \rho(f_{k2}, x_i, t), \ldots, \rho(f_{kk}, x_i, t)),$$

where:
x_i – is an object identifier,
$t \in T^{id}$ – is a moment of time,
$k1, k2, \ldots kk \in FI$.

2.4.3. The estimation models

The estimation model is used for finding the unknown values of the target feature that depend on some input data. Let us consider the set of objects X^{id} of a dynamic information system IT^{id}, the selected unknown feature of the object f_O (where $O \in FO$, FO – is the index of target (output) feature), called target feature, and the subsets of input features f_I ($I \in FI$, $FI = F \setminus FO$, FI – is the index set of input features, F – is the index set of all object features).

The model is defined as follows:

$$\rho(f_O, x_i, t) = M_E(\rho(f_{k1}, x_i, t), \rho(f_{k2}, x_i, t), \ldots, \rho(f_{kk}, x_i, t)),$$

where:
x_i – is an object identifier,
$t \in T^{id}$– is a moment of time,
$k1, k2, \ldots kk \in FI$.

2.4.4. The predictive models

The prediction model is used for finding the unknown values of the target that depend on some input historical data. The time is important in this model. Let us consider the set of objects X^{id} of a dynamic information system IT^{id}, the selected unknown feature of the object f_O (where $O \in FO$, FO – is the index of target (output) feature), called target feature, and the subsets of input features f_I ($I \in FI$, $FI = F \setminus FO$, FI – is the index set of input features, F – is the index set of all object features).

$$\rho(f_O, client\_id, t_{churn}) = M_P\big(\rho(f_1, client\_id, t_1), \rho(f_1, client\_id, t_2), \rho(f_1, client\_id, \dots), \rho(client\_id, f_1, t_T),$$
$$\rho(f_2, client\_id, t_1), \rho(f_2, client\_id, t_2), \rho(f_2, client\_id, \dots), \rho(f_2, client\_id, t_T),$$
$$\dots,$$
$$\rho(f_6, client\_id, t_1), \rho(f_6, client\_id, t_2), \rho(f_6, client\_id, \dots), \rho(f_6, client\_id, t_T),$$
$$\dots\big).$$

The model is defined as follows:

$$\rho(f_O, x_i, t_p) = M_E(\rho(f_{k1}, x_i, t_1), \rho(f_{k1}, x_i, t_2), \rho(f_{k1}, x_i, \dots),$$
$$\rho(f_{k2}, x_i, t_1), \rho(f_{k2}, x_i, t_2), \rho(f_{k2}, x_i, \dots),$$
$$\dots,$$
$$\rho(f_{kk}, x_i, t_1), \rho(f_{kk}, x_i, t_2), \rho(f_{kk}, x_i, \dots)),$$

where:
x_i – is an object identifier,
$T^{id} = \{t_1, t_2, \dots, t_T, t_p\}$,
$t_p = t_T + \zeta$, $\zeta > 0$,
t_p – is the prediction time,
$k1, k2, \dots kk \in FI$.

2.4.5. The association rules

Let us consider the set of objects X of a dynamic information system IT, the set of the identifiers of the rules $N = \{1, 2, \dots, m\}$, the selected subset of features of the object FP_i (where $FP_i \subset F$, $i \in N$, F – is the index set of all object features), and the subsets of features of object $FQ_i = F \setminus FP_i$.

The association rules are defined as follows:

$$P_i(\rho(f_{l1}, x_{l1}, t_{l1}), \rho(f_{l2}, x_{l2}, t_{l2}), \dots, \rho(f_{ll}, x_{ll}, t_{ll})) \Rightarrow$$
$$Q_i(\rho(f_{r1}, x_{r1}, t_{r1}), \rho(f_{r2}, x_{r2}, t_{r2}), \dots, \rho(f_{rr}, x_{rr}, t_{rr})),$$

where:
$i \in N$,
$f_{\dots}$ – is a feature of the object,
$x_{\dots}$ – is an object identifier,
$t_{\dots} \in T$– is a moment of time,
$l1, l2, \dots ll \in FP_i \quad \forall i \in N$,
$r1, r2, \dots rr \in FQ_i \quad \forall i \in N$.

3. An example of the model formulation

One of the main problems that have to be solved by marketing departments of telecommunications operator is a long-term relationship. They have found the way of convincing current clients to continue using the services. The methods that predicts the set of customers who are going to leave the operator might be a significant tool that improves the marketing campaigns [10, 12].
The telecommunication operator is storing a lot of information about the clients in the databases. At the detail level they have switch recordings in the form of call detail records (CDR). This information is useful for billing but can not be directly used for churn analysis. Therefore, this detailed information should be aggregated and additional data should be added. Table 1 shows a subset of the data for churn analysis.

Table 1
The features that describes the clients

Client id	f_1 t_1	f_2 t_1	f_3 t_1	f_4 t_1	f_5 t_1	f_6 t_1	...	churn t_c
1273	20	300	50	30	25	1	...	Y
2234	100	400	100	20	30	10	...	N
...	...	...	...	...	...	...	...	...

There are the following features of the clients in the Table 1:

- f_1 – remaining binding days,
- f_2 – total amount billed,
- f_3 – incoming calls,
- f_4 – outgoing calls within the same operator,
- f_5 – outgoing calls to other mobile operator,
- f_6 – international calls,
- and others.

The training information system is defined as follows:

$$IT^{Train} = (X^{Train}, F^{Train}, V^{Train}, \rho^{Train}, T^{Train}, R), \qquad (6)$$

where:
$T^{Train} = t_1, t_2, \dots, t_T, t_{churn}$, $t_{churn} = t_c = t_T + \zeta$, $\zeta > 0$,
X^{Train} – is the finite and nonemty set of clients,
F^{Train} – is finite and nonempty set of features of the objects,
$V^{Train} = \bigcup_{f \in F} V_f$, V_f, is a set values of feature $f \in F$, called the domain of f,
ρ^{Train} – is an information function:
$\rho : F^{Train} \times X^{Train} \times T^{train} \to V^{train}$.

A predictive model has been selected for a churn modeling. The "churn" feature has been selected as a target feature ($f_O = churn$), the indexes of the input features f_I belongs to the set $FI = \{1, 2, 3, 4, 5, \dots\}$.

The model is defined as follows – see the top of this page.

4. Conclusions

In this paper there is an overview of complex telecommunications problems modeling. We have applied the definiotion of the dynamic information system for a formal description of the preprocessing as well as model definition. We have stressed the importance of the preprocessing step in a data mining process. An example of churn model formulation has been provided. The presented approach might be stimulating for a development of various temporal data mining models.

References

[1] J.-L. Amat, "Using reporting and data mining techniques to improve knowledge of subscribers; applications to customer profiling and fraud management", *J. Telecommun. Inform. Technol.*, no. 3, pp. 11–16, 2002.

[2] M. J. Berry and G. S. Linoff, *Mastering Data Mining. The Art and Science of Customer Relationship Management*. Wiley, 2000.

[3] S. Greco, B. Matarazzo, and R. Słowiński, "Rough sets theory for multicriteria decision analysis", *Eur. J. Oper. Res.*, vol. 129, pp. 1–47, 2001.

[4] J. J. Lee and R. Ben-Natan, *Integrating Service Level Agreements. Optimizing Your OSS for SLA Delivery*. Indianapolis, Indiana: Wiley, 2002.

[5] R. Mattison, *Data Warehousing and Data Mining for Telecommunications*. Boston, London: Artech House, 1997.

[6] K. Morik and M. Scholz, "The MiningMart approach to knowledge discovery in databases" in *Handbook of Intelligent IT*, Ning Zhong and Jiming Liu, Eds. IOS Press, 2003

[7] E. Orłowska, "Dynamic information systems", *Ann. Soc. Math. Polon., Ser. IV: Fundam. Informat.*, vol. 5, no. 1, pp. 101–118, 1982.

[8] Z. Pawlak, "Rough sets", *Int. J. Inform. Comput. Sci.*, vol. 11, pp. 341–356, 1982.

[9] Z. Pawlak, *Systemy informacyjne. Podstawy teoretyczne*. Warszawa: WNT, 1983.

[10] Z. Pawlak, "Rough set theory and its applications", *J. Telecommun. Inform. Technol.*, no. 3, pp. 7–10, 2002.

[11] M. Shawa, C. Subramaniama, G. Tana, and M. Welgeb, "Knowledge management and data mining for marketing", *Decis. Supp. Syst.*, vol. 31, no. 1, pp. 127–137, 2001.

[12] C.-P. Wei and I.-T. Chiu, "Turning telecommunications call detail to churn prediction: a data mining approach", *Expert Syst. Appl.*, vol. 23, pp. 103–112, 2002.

Janusz Granat received his M.Sc. in control engineering (1996) and his Ph.D. (1997) in computer science from the Warsaw University of Technology. He holds a position as an Assistant Professor at the Warsaw University of Technology, and is the leader of a research group on applications of decision support systems at the National Institute of Telecommunications in Warsaw. He lectured decision support systems and various subjects in computer science. His scientific interests include data mining, modeling and decision support systems, information systems for IT management. Since 1988 he has been cooperating with IIASA. He contributed to the development of decision support systems of DIDAS family and the ISAAP module for specifying a user preferences. He has been involved in various projects related to data warehousing and data mining for telecommunication operators. He is involved in EU MiningMart project.
e-mail: J.Granat@itl.waw.pl.
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Institute of Control and Computation Engineering
Warsaw University Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Paper

The use of quantitative association rules in cellular network planning

Michał Okoniewski

Abstract — This paper describes the problem of planning cellular network base stations with optimization to traffic requirements. This research problem was a main incentive to add some development to the theory of association rules. The new form of quantitative and multi-dimensional association rules, unlike other approaches, does not require the discretization of real value attributes as a preprocessing step. They are discovered with data driven algorithm that gives precise and complete results and has polynomial complexity for a given dimensionality.

Keywords — data mining, quantitative association rules, knowledge discovery process, cellular network planning.

1. Problem statement

One of the most important areas for a young cellular telecom provider is network expansion. This creates a need for traffic prediction, i.e. we would like to estimate the number of calls made during a certain time span, on an area where we want to build a new base station. Such information is crucial for station equipment design – there must be enough transceivers to ensure that every subscriber in the GSM cell created by this station is able to place or receive call. On the other hand, there should not be too much available – and unused – radio channels, because this would mean unnecessary costs.

Traffic prediction is a complex task, as the number of subscribers present on a certain area may vary. After all GSM is an abbreviation of name Global System for Mobile Telecommunications, and GSM subscribers travel between cells, for example moving into city centers at day, and going to suburbs (where their homes are) in the evening. Similar effect can be observed also for longer time periods. So called vacation traffic analysis shows that in the summer average traffic generally increases in popular resort areas – like mountains, seashore etc. Fortunately our analysis showed that these variations are periodic, and predictable, at least for regions with well developed GSM coverage. We can therefore try to predict traffic for a certain characteristic time period – say, for vacation time, using measurements of existing network elements, and than interpolate obtained values.

In this particular experiment we were able to extract two types of information from ERA GSM network monitoring system. First was the traffic information. For each cell we obtained average of weekly traffic measured at busy hour (usually around midday). Other type of information has been extracted from company's geographical information system (GIS). For each cell, the types of terrain occupied have been established. The GIS database contained information about nine terrain types (landuses) that may occur in particular cell. These were:

1) fields,
2) forests,
3) water,
4) rocks, seashores, swamps,
5) roads, concrete, parks,
6) suburbs,
7) urban,
8) dense urban,
9) industrial.

For each cell the amount of ground pixels occupied by every landuse have been defined. One ground pixel width and length are approximately 5 seconds of arc. Our initial data about existing network have been collected and recorded in a table with following attributes:

– cell identification number,
– landuse type 1..9,
– number of pixels occupied by above landuse,
– cell number that allows to determine region in which the cell is situated,
– traffic – the average weekly value in Erlangs for a given cell.

The main goal of the project was to build a predictive model of network traffic that would allow analysts to plan the power of base station in newly created radio network cells. Various methods such as neural networks, clustering, regression or regressional clustering were used [1, 2].

However there were almost no results with classic association rules. It was due to the fact that the quality of data, already not perfect, was made even worse with discretization. It turned out that it's difficult to find a discretization that does not loose informational value of discovered rules. After the research project in thesis [7] there was introduced a new theory of quantitative rules that could be able to cope with the problem using new form of rules and data driven approach instead of discretization.

2. New form of quantitative rules

In [4] rules with single numeric attribute in both antecedent and consequent are presented. In this paper we consider their generalized forms. Thus, definitions included in this section are multi-dimensional extensions of definitions for "Quantitative to Quantitative" rule from [4].

Notations. Let D be a relational table with a set of quantitative attributes $E = \{I_1, I_2, \ldots, I_k, J\}$.
Letters $A, B, \ldots$ mean single attributes from E, while $X, Y, \ldots$ mean subsets of E. Table D may be viewed as a set of tuples $D = \{t_1, t_2, \ldots, t_n\}$. Notation $t_i.A$ indicates the value of attribute A for i tuple. A range (A, a, b) is defined by a single attribute $A \in E$ and two numbers $\{a, b\} \in domain(A) \subseteq \mathbf{R}, a \leq b$. A profile Pr_X over $X \subseteq E$ is defined as a common part of ranges $\bigcap_{i \in X}(a_i, b_i)$ – one range for each attribute in X. Notation $(A, a, b) \in Pr_X$ means that range (A, a, b) is one of the ranges that delimit Pr_X.
Basically, a profile may be simply viewed as a k-dimensional hyper-cuboid. $|Pr_X|$ is a number of tuples from D that have all corresponding attribute values within profile Pr_X. A statistical measure M is defined over distribution of attribute J values. $M(Pr_X)$ is a value of this measure for distribution of J for tuples that have all corresponding attribute values within Pr_X . In addition, $M(D)$ is the measure value for distribution of J attribute values for the whole D. As in [4], the measure M is usually the mean of J values.
The antecedent of the rule is a profile that defines subpopulation of tuples that is significantly different from the whole D with regard to the attribute J. It is assured by second condition (a difference condition) that holds if there is a minimal difference $mindif$ between the measure for D and for the Pr_X. In [4] standard methods for statistical hypothesis testing were then applied (e.g. a Z-test for the mean) to check the significance of the difference. Another condition is a standard support requirement for an association rule.
With the use of above notations we can build up a definition of generalized "Quantitative to Quantitative" rule.

Definition 2.1. Multi-dimensional (mean based) quantitative association rule is a rule of the form:

$$Pr_X \Rightarrow M(Pr_X) \ \ (M(D)),$$

where:

- $X \cap Y = \emptyset$,
- $|M(Pr_X) - MD| \geq mindif$,
- $|Pr_X| \geq minsup$.

Constants $mindif$ and $minsup$ are user-defined parameters. There is no confidence parameter of the rule. The rule has the difference parameter $dif = M(Pr_X) - M(D)$ instead, to indicate its strength. Let us here specify minimal M for a rule by $\mu = M(D) + mindif$. The dimensionality of the rule is equal to the number of attributes in its profile.

Remark. Definition 2.1 describes a rule that has the mean above average ($(M(Pr_X) > M(D))$. The work in this paper considers above-average rules that follow this definition. All this may be also applied by the simple analogy for below-average rules.
Examples of quantitative rules are:

$$cigarettes\_daily \in (10, 20) \wedge overweight \in (10, 20) \Rightarrow life\_expectancy = 58 (life\_expectancy = 72)$$

$$latitude \in (49N, 50N) \wedge longitude \in (19E, 21E) \Rightarrow avg\ temp.April = 3C (avg\ temp\ April\ Poland = 7C)$$

Important notions are irreducible and maximal rules. These are rules that are intuitively "interesting" to discover:

Definition 2.2 (irreducible rule). Rule $Pr_X \Rightarrow M_J(T_{Pr_X})$ is irreducible, if for every range $(A, a, b) \in Pr_X$ and every number c, $a < c < b$ the following is true: profiles $Pr_X 1$ and $Pr_X 2$ that are created by exchanging (A, a, b) in Pr_X respectively with ranges (A, a, c) and (A, b, c) result in rules $Pr_X 1 \Rightarrow M_J(T_{Pr_X 1})$ i $Pr_X 2 \Rightarrow M_J(T_{Pr_X 2})$ that fulfill at least the difference condition from Definition 2.1.

Definition 2.3 (maximal rule). Rule $Pr_X \Rightarrow M_J(T_{Pr_X})$ is a maximal rule , if for every range $(A, a, b) \in Pr_X$ and every c, $c > b$ ($c < a$) the rule which is created by exchanging range (A, a, b) in the input rule with range (A, a, c) ((A, c, b)) does not fulfill the difference condition from Definition 2.1 or is reducible.
Accordingly, irreducible rule profile may be divided by any hyperplane $A = c$ into two profiles, that maintain above-average difference condition. As it is pointed out in Section 5 in multiple dimensions irreducibility is not good enough to the intuitive connotation of a rule with homogeneous distribution of tuples. However, the irreducibility is a basic quality that makes the rule desired.
Maximal rule is one that cannot be extended into a single dimension. Nonetheless, it may be extended into two or more dimensions by enlarging more than one range from Pr_X . That is why definition of maximization is useful mainly for one-dimensional rules.
Let us present two theorems that describe properties of quantitative rules and are essential for discovering them.

Theorem 2.1. If the quantitative association rule $(A, a, b) \Rightarrow M_J(T_{Pr_X})$ is irreducible, then

$$\forall_{(A,a,b) \in Pr_X} \exists_{t_1, t_2 \in D}\ t_1.A = a \wedge t_2.A = b \wedge t_1.J \geq \mu \wedge t_2.J \geq \mu$$

Theorem 2.2. There are minimum 2, maximum $2k$ μ-tuples to define a profile of the irreducible rule, where k is the dimensionality of the profile.

For example, a profile in two dimensions is defined by 2,3 or 4 μ-tuples (Fig. 1).

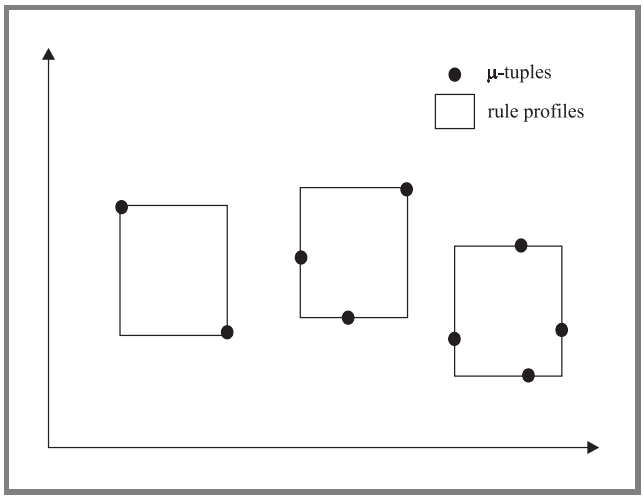


Fig. 1. The μ-tuples that define a profile.

3. Exploratory algorithm

The general outline of the mining algorithm that utilizes Theorems 2.1 and 2.2 is as follows. First, select all the μ-tuples. Then consider the μ-tuples in sets from minimum 2 to maximum $2k$ elements. For each set find minimal and maximal values for each attribute $I_1, \ldots, I_k$. In this way we obtain suspected profile boundaries. Then, check the irreducibility of the profile by incrementally checking divisions of the profile into two hyper-cuboids with all included tuples. All the hyper-cuboids have to be above average in terms of the mean J value. If only one of the checks fails, the profile does not form a rule and so may be rejected. The algorithm may be sketched in a recursive form that works on object variable MiTupleSet:

```
1   int CheckMiTuples(int level,int TupleNo)
2   {
3   if (level < 2*k)
4   {
5   for(int i=TupleNo+1;i<=MiTuplesQ,i++)
6   {
7   AddTuple(MiTupleSet,i);
8   if(level>1) CheckProfile(MiTupleSet);
9   CheckMiTuples(level+1,i);
10  }
11  }
12  }
13  FindQRules()
14  {
15  SelectAndSortMiTuples();
16  CheckMiTuples(1,0);
17  }
```

The computational complexity of above algorithm depends on the percentage p of μ-tuples in the database , and may be estimated [7] as $O(k(pn)^{2k})$. This assumes that the cost of selecting tuples inside a profile hyper-cuboid is small, because of effective indexing method for k attributes.
The complexity is polynomial with the number of tuples, but may be still considered high. It may be decreased with clustering strategies for μ-tuples.
The efectiveness of the algorithm and comparisons with classic mining algorithms for quantitative rules [8] was described in detail in [7].

4. Application

As described in section with problem statement, the space may be divided into regions, for example mobile telecom cells. For each region we can establish a number of numeric parameters (e.g. population or percentages of area types in the cell: forests, urban, water, etc.). For each region we obtain a tuple of attributes $I_1, \ldots, I_k$ that stand for parameters plus one analyzed attribute J. These tuples are the input to rule discovery algorithm. As a result we obtain a rule-based predictive model that may be used for classification of other regions in the space. Examples of such rules are

$$urban \in (10\%, 28\%) \wedge$$
$$\wedge roads \in (5\%, 11\%) \wedge forest \in (0\%, 8\%) \Rightarrow$$
$$traffic = 4.1erl (traffic = 1.2erl)$$

$$water \in (6\%, 25\%) \wedge forest \in (30\%, 80\%) \Rightarrow$$
$$traffic = 0.2erl (traffic = 1.2erl)$$

$$water \in (15\%, 35\%) \wedge suburbs \in (20\%, 40\%) \Rightarrow$$
$$traffic = 3.0erl (traffic = 1.2erl)$$

5. Other areas of application and future work

The new form of rules may be used also directly to raw spatial data. Such data may be sampled, even at random, and used as an input to rule mining algorithm. It is obvious that the frequency of sampling increases rule accuracy and consequently the algorithm running time. Antecedent attributes $I_1, \ldots, I_k$ are coordinates of points in 2D, 3D or even higher dimensionality space. Decisive attribute J describes the analyzed value (i.e. elevation, temperature, cellular traffic, etc.). As a result we obtain hyper-cuboid regions (squares in 2D) where the value is high above (or below) average for the hole space.
Sampled graphic files are almost the same case as raw spatial data. In [7] there is described a series of experiments that prove usefulness of new methodology to discover patterns in graphic files.

Future work planned in the area of data driven discovery of quantitative rules is as follows:

- rule management systems to filter interesting and representative rules,
- enhanced mining algorithms,
- better use of database systems – some experiments with multidimensional indexing [3] were already done,
- intelligent management of knowledge discovery process,
- the use of new rules for various form of multimedia data.

References

[1] P. Gawrysiak and M. Okoniewski, *Applying Data Mining Methods for Cellular Network Planning, Intelligent Information Systems*. Springer-Physica Verlag, 2000.

[2] P. Gawrysiak, M. Okoniewski, and H. Rybiński, *Regression – Yet Another Clustering Method, Intelligent Information Systems*. Springer-Physica Verlag, 2001.

[3] V. Geade and O. Günther, "Multidimensioanal access methods", *ACM Comput. Surv.*, vol. 30, no. 2, 1997.

[4] Y. Lindell and Y. Aumann, "Theory of quantitative association rules with statistical validation" in *Proc. Conf. SIGKDD*, San Diego, USA, 1999.

[5] R. J. Miller and Y. Yang, "Association rules over interval data" in *Proc. Conf. ACM SIGMOD'97*, 1997.

[6] M. Okoniewski, "Discovery of multi-dimensional quantitative association rules". Ph.D. thesis. Warsaw University of Technology, Warsaw, 2001.

[7] M. Okoniewski, Ł. Gancarz, and P. Gawrysiak, "Mining multidimensional quantitative associations" in *Rule-based Data Mining Workshop, Conf. INAP 2001*, Tokio, Japan, 2001.

[8] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables" in *Proc. Conf. SIGMOD*, Montreal, Canada, 1996.

[9] A. Skowron and S. H. Nguyen, "Quantization of real value attributes: rough set and boolean reasoning approach". Tech. Rep., Warsaw University of Technology, Warsaw, 1995.

Michał Okoniewski received the Ph.D. in computer science from Warsaw University of Technology, Poland, in 2002. He is currently post-doc researcher at Intelligent Systems Laboratory (ISLab) at the University of Antwerp, Belgium. His research interests include knowledge discovery, information systems and bioinformatics.

okoniews@ii.pw.edu.pl
Institute of Computer Science
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland
e-mail: Michal.Okoniewski@ua.ac.be
Intelligent Systems Laboratory (ISLab)
Department of Mathematics and Computer Science
University of Antwerp, RUCA
Middelheimlaan 1
B-2020 Antwerpen, Belgium

Regular paper

Near fields of elliptic dielectric lenses

Artem V. Boriskin and Marian Marciniak

Abstract — The focusing properties of an elliptic dielectric cylinder taken as a 2D model of dielectric lens are studied for the plane wave illumination. An algorithm based on the concept of analytical regularization is applied for the numerical solution of the corresponding wave scattering problem. Numerical results for the near-field patterns are presented.

Keywords — elliptic dielectric lens, method of analytical regularization, antennas.

1. Introduction

Planar slot or strip elements combined with dielectric lenses have the potential to be used in mm and sub-mm wave receivers [1, 2]. The wide attention they have been attracting recently is due to their capability of integration with electronic components such as detecting diodes, local oscillators and mixers. Furthermore, they provide good efficiency with respect to other antennas printed on homogeneous substrates. Besides, lenses are frequently employed in the laser technologies for the compression of the light beams. The elliptical shape of the lens provides focusing properties if its eccentricity is properly related to the dielectric constant. On the other hand the lens interface gives rise to reflection inside the lens that may significantly affect the input impedance and the radiation sensitivity. This aspect has not been properly investigated in the literature up to now, however, it is a critical point in the overall design of lens antennas. Although various analytical techniques have been applied for the dielectric lens analysis, they were commonly based on high-frequency approximations, neglected the lens curvature and finite beam size, and failed to characterize resonances. This appears a rough model description, as the actual size of the lens is usually of the scale of few wavelengths only [1, 2]. On the other hand, direct numerical simulations like FDTD may suffer from unclear and uncontrollable accuracy. As a consequence, there is a need to develop a reliable simulation technique capable to model wavelength-scale effects in addition to geometrical optics ones.

Our main efforts will be concentrated around the analysis of elliptic lenses in receiving mode. It is known [1, 2] that if the eccentricity of the ellipse is related to the lens dielectric constant as $e = 1/\sqrt{\varepsilon}$, all the rays outgoing from the focal point that impinge on the lens interface above the middle section form a parallel beam, i.e. plane wave. However, in practice it is very important to take into account finiteness of source size, and to select a proper source position and the lens geometry (Fig. 1).

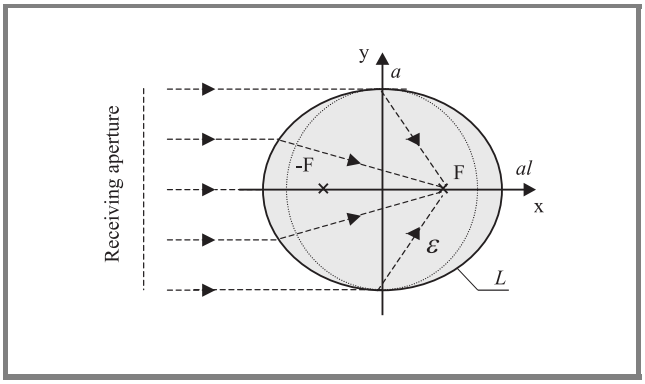


Fig. 1. Geometry and notations of the problem.

In the receiving mode, it is important to know the actual size and location of the focal domain for different electric sizes of lenses and different angles of arrival of the incident wave. Although those effects escape conventional asymptotic analyses, they will be accurately quantified with our full-wave mathematically correct integral-equation method applied here. We emphasize that there is no limitation on the dielectric constant contrast between the lens material and background medium.

2. Outline of the solution

We use an efficient algorithm for the solution of 2D problem of wave scattering by a smooth dielectric cylinder that can be built on the concept of analytical regularization [3, 4]. The basic idea of the approach is as follows. The total field has to satisfy the Helmholtz equation with the coefficient k and $k_e = k\sqrt{\varepsilon}$ outside and inside the lens, respectively. Field functions are presented as single-layer potentials with the density functions to be determined:

$$U_{outside}(\mathbf{r}) = \int_L p(\mathbf{r}_s) G_0(\mathbf{r}, \mathbf{r}_s)\, d\, l_s + U_0\,, \qquad (1)$$

$$U_{inside}(\mathbf{r}) = \int_L q(\mathbf{r}_s) G_e(\mathbf{r}, \mathbf{r}_s)\, d\, l_s\,. \qquad (2)$$

Here U_0 is the incident field, and the kernels are the Green's functions of the free space and uniform media of relative permittivity ε, respectively:

$$G_0(\mathbf{r},\mathbf{r}_s)=\frac{i}{4}H_0(k|\mathbf{r}-\mathbf{r}_s|)\,, \tag{3}$$

$$G_e(\mathbf{r},\mathbf{r}_s)=\frac{i}{4}H_0(k_e|\mathbf{r}-\mathbf{r}_s|)\,, \tag{4}$$

and H_0 is the Hankel function of the 1st kind.
By applying the boundary conditions, a set of singular integral equations (IEs) of the first kind is obtained. Introducing a global parameterization $x(t)$, $y(t)$ of the contour L, they can be cast into the following form:

$$\begin{cases} \int\limits_0^{2\pi} q(t_s)G_\varepsilon(t,t_s)L(t_s)dt_s-\int\limits_0^{2\pi} p(t_s)G_0(t,t_s)L(t_s)dt_s= \\ =U_0(t_s) \\ \frac{q(t)+\alpha p(t)}{2\alpha}+\frac{1}{\alpha}\int\limits_0^{2\pi} q(t_s)\frac{\partial}{\partial n}G_\varepsilon(t,t_s)L(t_s)dt_s+ \\ -\int\limits_0^{2\pi} p(t_s)\frac{\partial}{\partial n}G_0(t,t_s)L(t_s)dt_s=\frac{\partial}{\partial n}U_0(t)\,, \end{cases} \tag{5}$$

where $t,t_s\in[0,2\pi]$, and $L(t)=\sqrt{(dx/dt)^2+(dy/dt)^2}$.
Equations (5) are uniquely solvable provided that the wavenumber k does not belong to the set D of discrete eigenvalues of the interior Dirichlet problem for L [5]. Direct discretization of IEs (5) is not efficient due to the singular character of the kernel functions. In order to transform them to the Fredholm second kind matrix equation with favorable features, the analytical regularization has to be done. Adding and subtracting the canonical-shape kernels

$$\hat{G}_0(t-t_s)=\frac{i}{4}H_0\left(2ka\left|\sin\frac{t-t_s}{2}\right|\right), \tag{6}$$

$$\hat{G}_e(t-t_s)=\frac{i}{4}H_0\left(2k_ea\left|\sin\frac{t-t_s}{2}\right|\right), \tag{7}$$

and their normal derivatives perform extraction of the singular parts of the IEs. Analytical inversion of the latter parts is carried out by using the IE discretization based on Galerkin's scheme with angular exponents as global basis functions. Thus, unknown density functions are sought as

$$\{p(t),q(t)\}L(t)=\frac{2}{i\pi}\sum_{m=-\infty}^{\infty}\{p_m,q_m\}e^{imt}. \tag{8}$$

Here, the angular exponents are the orthogonal eigenfunctions of the integral operators, for example,

$$\int_0^{2\pi} e^{imt_s}H_0\left(2kaSin|(t-t_s)/2|\right)dt_s=2\pi e^{imt}J_m(ka)H_m(ka), \quad m=0,\ \pm1,\ \pm2,\ldots \tag{9}$$

Resulting matrix equation is

$$Z+AZ=B, \tag{10}$$

where $Z=\{z_m^1,z_m^2\}\big|_{m=-\infty}^{\infty}$,

$$z_m^1=p_mH_m(ka)J_m(ka)-\alpha q_mH_m(k_ea)J_m(k_ea) \tag{11}$$

$$z_m^2=kap_mH_m'(ka)J_m(ka)-k_eaq_mH_m(k_ea)J_m'(k_ea) \tag{12}$$

and the elements A_{nm}^{ij} and $B_m^j(i,j=1,2)$ depend on the Fourier-expansion coefficients of the smooth functions. The latter are the differences between kernels (3) and (4) on L and on the canonical-shape contour, i.e. a circle of radius a. These matrix elements can be economically computed by using the DFFT algorithm. The coefficient α in Eqs. (11) and (12) is 1 or ε for E- or H- polarization, respectively.

Such a regularization plays the role of analytic preconditioning and guarantees point-wise convergence of the numerical solution (provided that $k\notin D$), i.e., a possibility to minimize the error to machine precision by solving progressively greater matrices.

Here, one has to note that the resulting computational error is determined by several factors: accuracy of cylindrical functions calculation, accuracy of FFT in the coefficients calculation, accuracy of numerical integration, and finally, the truncation error. In our algorithm, Bessel functions are calculated with digital precision that is achieved by using the recursion technique (backward for Bessel and forward for Neumann function). The bottleneck of the algorithm is accuracy of the matrix element calculation that is controlled by the order of FFT.

Under these conditions the rate of convergence of the algorithm can be estimated by plotting the normalized computational error, $e(N)$, in the sense of the l_2^2 norm, versus the matrix truncation number N:

$$e(N)=\left\|Z^N-Z^{N+1}\right\|\cdot\left(\left\|Z^N\right\|\right)^{-1}, \tag{13}$$

where $Z^N=\{z_n^{1N},z_n^{2N}\}$ are the expansion coefficients computed from the matrix equation with each block truncated after N equations. Details of the algorithm properties can be found in [4].

As it has been noted, there is a discrete set of wavenumbers, D, which are defective for the developed algorithm. When intermediate operations are done with finite precision, the condition number of the matrix (10) blows up in the vicinity of $k\in D$ that is comparable in size with the precision. This entails a spurious resonance in the field characteristics of the scatterer that is also called "numerical resonance" in the contrast to natural physical resonances. This feature is a demerit of the algorithm. It is caused by the implementation of the single-layer potentials (3) and (4) and can be overcome when using, instead, linear combination of single and double-layer potentials [5]. Nevertheless our approach is free from the inaccuracies near to natural resonances that are intrinsic for conventional numerical approximations [6].

As the far-field characteristics are of interest, the large-r approximation is used. This enables one to replace the Hankel

functions with its asymptotic and to reduce the integral in formula (1) to

$$U^{rad}(t) = \left[\left(1/i\pi kr\right)^{1/2} \cdot e^{ikr}\right] \cdot \Phi(t), \qquad (14)$$

where $\Phi(t)$ is the far-field radiation pattern determined as:

$$\Phi(t) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} p_n \int_0^{2\pi} e^{-ik\left(\cos(t)x(t_s)+\sin(t)y(t_s)\right)+\mathrm{int}_s} dt_s. \qquad (15)$$

The frequency dependence of the total scattered power can be characterized by the cumulative scattering characteristic such as a total-scattering cross-section σ_{tot} determined as:

$$\sigma_{tot} = \frac{2}{\pi k} \int_0^{2\pi} \left|\Phi(t)\right|^2 dt. \qquad (16)$$

3. Numerical results

The dielectric materials selected for the computations are high-density polyethylene (HDP), quartz, alumina, and silicon with ε of 2.31, 4.0, 9.8, 11.7, respectively. The dimensions of the considered lenses are about several wavelengths. The axial ratio of the considered lenses is $l = \sqrt{\varepsilon/(\varepsilon-1)}$ to provide the eccentricity condition $e = 1/\sqrt{\varepsilon}$.

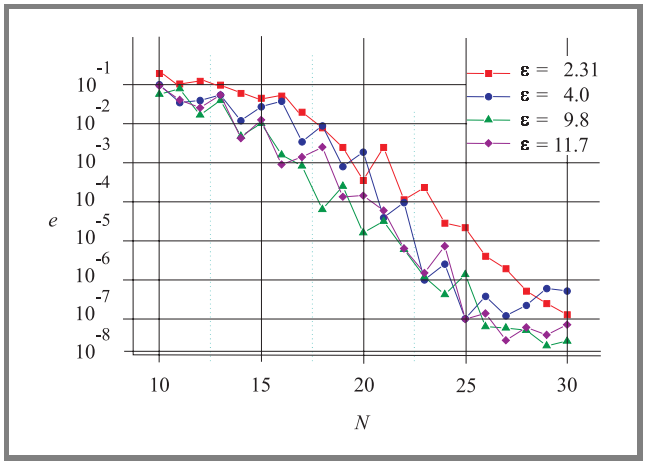


Fig. 2. Relative computational error versus truncation number for different dielectric materials and $ka = 10$.

Figure 2 shows the dependence of the relative computational error versus the truncation number N. Rapid fall of the error with the growth of N is due to the point-wise convergence of the solution that comes from the Fredholm nature of the considered matrix equation. All the following numerical results have been obtained with the error below 10^{-3}.

In Fig. 3 the total scattering cross-section versus the normalized wavelength parameter ka for various dielectric materials is shown. Sequences of extrema, which are well seen in the graph, are explained by the natural resonances. Extraordinarily high-Q maxima correspond to spurious numerical resonances that are involved in the true solution due to chosen representation of the fields.

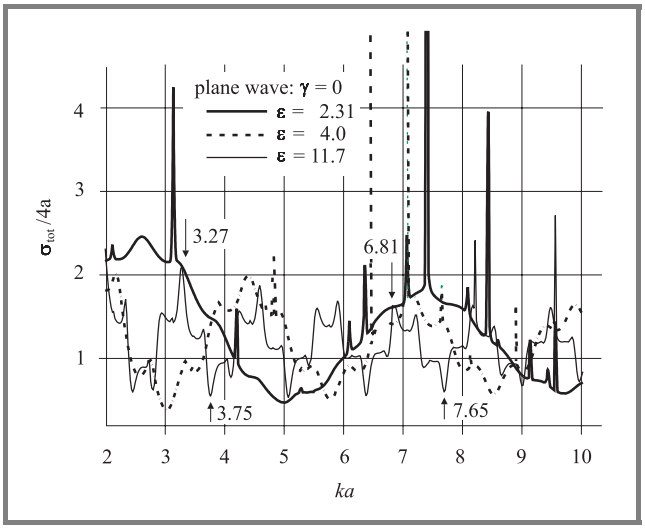


Fig. 3. Normalized total scattering cross-section of the elliptic cylinder versus normalized frequency parameter for different dielectric materials (E-polarized plane wave illumination in the frontal mode).

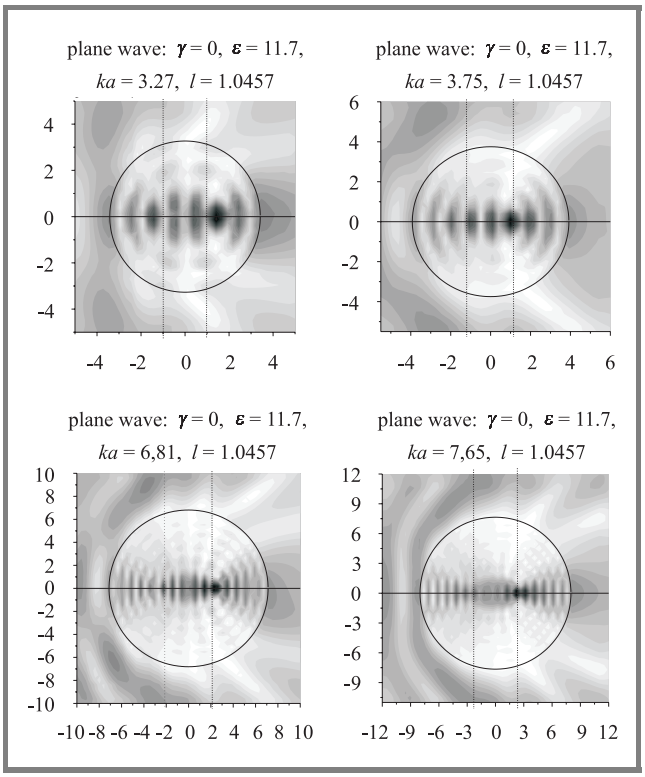


Fig. 4. Near-field portraits for a silicon lens symmetrically illuminated by the E-polarized plane wave. Intersections of the plane of symmetry with the vertical dashed lines indicate the focal points of the lens. Corresponding ka values are marked with arrows in Fig. 3.

There are two pairs of near-field portraits (Fig. 4) calculated for silicon lens with $ka = 3.27$, 3.75 and 6.81, 7.65, respectively. These values of the normalized wavelength parameter *ka* are marked with arrows in Fig. 3 and correspond to the local maxima and minima in the total scattering cross-section. It can be seen that the focal domain has finite size: the greater *ka* the smaller the area of the greatest field concentration. The shift of the focal domain from the geometrical focus of the ellipse is interrelated to the total scattering cross-section: maximum shift is for the maxima of σ_{tot}.

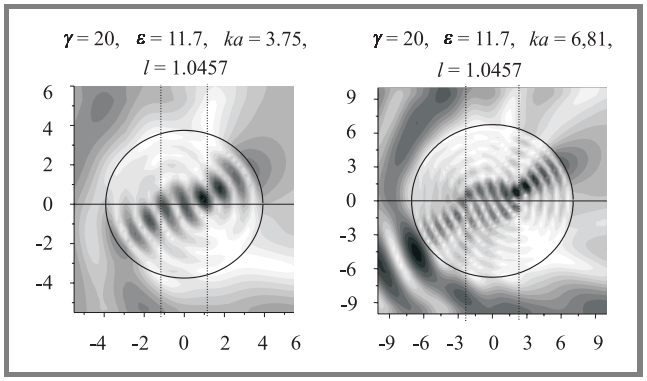


Fig. 5. The same as in Fig. 4 for the incident wave arrival angle $\gamma = 20°$.

Figure 5 demonstrates the transformation of the focal domain in the case of the plane wave illuminating the lens at the angle of 20°. It is shown that the focal domain shifts from the ellipse geometrical focus and becomes wider.

4. Conclusions

An efficient and accurate numerical method, based on the concept of analytical regularization, has been applied for the solution of the 2D problem of the plane wave scattering by an elliptic dielectric cylinder taken as a model of typical lens used in the mm and sub-mm wave antenna applications. Obtained numerical results demonstrate the effects such as focal domain shift and transformation for a dielectric lens of the size comparable to the wavelength that cannot be analyzed in geometrical or physical optics approximations. As it is shown, the resonant nature of the effects is still a challenge for more careful analysis. The question of nonuniqueness of the solution of IEs used in the mathematical model also requires a further study. Nevertheless the possibilities of the approach as to the accurate analysis of the smooth arbitrary shaped dielectric lenses have been demonstrated.

Acknowledgments

The authors express their greatest thanks to Dr. Svetlana V. Boriskina for her valuable assistance in computations and to Prof. Alexander I. Nosich for his highly relevant advices. This study has been included in the National Institute of Telecommunications' activities in the framework of European Research Program COST 268 "Wavelength-Scale Photonic Components for Telecommunications".

References

[1] D. F. Filippovich, S. S. Gearhart, and G. M. Rebeiz, "Double slot on extended hemispherical and elliptical silicon dielectric lenses", *IEEE Trans. MTT*, vol. MTT-41, no. 10, 1993.

[2] A. Neto, S. Maci, and P. J. I. de Maagt, "Reflections inside an elliptical dielectric lens antenna", *IEE Proc. Microw., Antenn., Propagat.*, vol. 145, no. 3, pp. 243–247, 1998.

[3] A. I. Nosich, "The method of analytical regularization in wave-scattering and eigenvalue problems: foundations and review of solutions", *IEEE Antenn. Propagat. Mag.*, vol. 41, pp. 34–49, 1999.

[4] S. V. Boriskina and A. I. Nosich, "Method of analytical regularization in the problems of wave diffraction by dielectric cylinders of arbitrary cross-sections", *Radio Phys. Radio Astron.*, vol. 3, no. 4, pp. 405–413, 1998 (in Russian).

[5] N. Morita, N. Kumagai, and A. Mautz, *Integral Equation Methods for Electromagnetics*. Boston: Artech House, 1990.

[6] G. L. Hower *et al.*, "Inaccuracies in numerical calculations of scattering near natural frequencies of penetrable objects", *IEEE Trans. Antenn. Propagat.*, vol. AP-41, no. 7, pp. 982–986, 1993.

Artem V. Boriskin was born in Kharkov, Ukraine, in 1977. He received his M.S. degree in radio-physics from the Kharkov National University, in 1999. He is currently a Ph.D. student at the Institute of Radio-Physics and Electronics of the National Academy of Sciences of Ukraine. His research interests are in the electromagnetic wave radiation and scattering by various dielectric objects with applications to dielectric lenses, rod antennas and antenna radomes. He received an IEEE Microwave Theory and Technique Society Graduate Student Fellowship Award in Microwave Engineering and a TUBITAK (Scientific and Technical Research Council of Turkey) – NATO Fellowship Award in 2000 and 2001, respectively.
e-mail: a_boriskin@yahoo.com
Institute of Radiophysics and Electronics
National Academy of Sciences
Proskury st 12
Kharkov 61085, Ukraine

Marian Marciniak Associate Professor has been graduated in solid state physics from Marie-Curie Sklodovska University in Lublin, Poland, in 1977. From 1985 to 1989 he performed Ph.D. studies in electromagnetic wave theory at the Institute of Fundamental Technological Research, Polish Academy of Sciences, followed by Ph.D. degree (with distinction) in optoelectronics received from Military University of Technology in Warsaw. In 1997 he received his Doctor of Sciences (habilitation) degree in physics/optics from Warsaw University of Technology. From 1978 to 1997 he held an academic position in the Military Academy of Telecommunications in Zegrze, Poland. In 1996 he joined the National Institute of Telecommunications in Warsaw where he actually leads the Department of Transmission and Fibre Technology. Previous activities have included extended studies of optical waveguiding linear and nonlinear phenomena with analytic and numerical methods including beam-propagation methods. Actual research interests include photonic crystal technology and phenomena, optical packet-switched networks, and the future global optical and wireless network. Recently he has introduced and developed a concept of a hybrid real-time service end photonic packet network. He is an author or co-author of over 190 technical publications, including a number of conference invited presentations and 13 books authored, co-authored and/or edited by himself. He is a Senior Member of the IEEE – Lasers & Electro-Optics, Communications, and Computer Societies, a member of The New York Academy of Sciences, The Optical Society of America, SPIE – The International Society for Optical Engineering and its Technical Group on Optical Networks, and of the American Association for the Advancement of Science. In early 2001 he originated the IEEE/LEOS Poland Chapter and he has served as the Chairman of that Chapter until July 2003. He is widely involved in the European research for optical telecommunication networks, systems and devices. He was the originator of accession of Poland to European Research Programs in the optical telecommunications domain, in chronological order: COST 240 *Modelling and Measuring of Advanced Photonic Telecommunication Components*, COST P2 *Applications of Nonlinear Optical Phenomena*, COST 266 *Advanced Infrastructure for Photonic Networks*, COST 268 *Wavelength-Scale Photonic Components for Telecommunications*, COST 270 *Reliability of Optical Components and Devices in Communications Systems and Networks*, COST 273 *Towards Mobile Broadband Multimedia Networks*, and very recently two new starting actions COST 288 *Nanoscale and Ultrafast Photonics* and COST P11 *Physics of linear, nonlinear and active photonic crystals*. In all but two those projects he acted as one of the originators at the European level. He has been appointed to Management Committees of all those Projects as the Delegate of Poland. In addition, he has been appointed as the Evaluator of the European Union's 5th Framework Program proposals in the Action Line *All-Optical and Terabit Networks*. He is a Delegate to the International Telecommunication Union, Study Group 15: *Optical and Other Transport Networks*, and to the International Electrotechnical Commission, Technical Committee 86 *Fibre Optics* and its two sub-Committees. He served as a member of Polish Delegation to the World Telecommunication Standards Assembly WTSA 2000. From 2002 he participates in the work of the URSI – *International Union of Radio Science, Commission D – Electronics and Photonics*. In 2000 he originated and actually serves as the Chairman of the Technical Committee 282 on *Fibre Optics* of the National Committee for Standardisation. Since May 2003 he serves as the Vice-President of the Delegation of Poland to the Intergovernmental Ukrainian-Polish Working Group for Cooperation in Telecommunications. He is the originator and the main organiser of the *International Conference on Transparent Optical Networks* ICTON starting in 1999, and a co-located events the *European Symposium on Photonic Crystals* ESPC and *Workshop on All-Optical Routing* WAOR since 2002. He is the Technical Program Committee Co-Chair of the *International Conference on Advanced Optoelectronics and Lasers* CAOL, and he participates in Program Committees of the *Conference on the Optical Internet & Australian Conference on Optical Fibre Technology* COIN/ACOFT, the *International Conference on Mathematical Methods in Electromagnetic Theory* MMET, the *International Workshop on Laser and Fiber-Optical Network Modeling* LFNM, and the *International School for Young Scientists and Students on Optics, Laser Physics and Biophysics/Workshop on Laser Physics and Photonics*. He serves as a reviewer for several international scientific journals, and he is a Member of the Editorial Board of *Microwave & Optoelectronics Technology Letters* journal, Wiley, USA, and the *Journal of Telecommunications and Information Technology*, National Institute of Telecommunications, Poland. Languages spoken: Polish (native), English, French, and Russian. His biography has been cited in Marquis *Who's Who in the World, Who's Who in Science and Engineering*, and in the *International Directory of Distinguished Leadership of the American Biographical Institute*.

e-mail: M.Marciniak@itl.waw.pl
e-mail: marian.marciniak@ieee.org
Department of Transmission and Fibre Technology
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Regular paper

A solution for increasing data rate of Doppler-RAKE system

Minh Nguyen Nguyen and Józef Modelski

Abstract — Doppler-RAKE detection in the CDMA system has been further developed and offers better performances in comparison to conventional RAKE detection, especially in fast-fading environments. Also, the multi-user Doppler-RAKE system works more effectively with channel coding applications. However, by means of exploring the Doppler effect, the system's data rate is decreased. We propose a simple solution to increase the data rate for the system while keeping the Doppler gain.

Keywords — diversity, Doppler-RAKE, CDMA, multi-path, fast-fading, spread-time signaling, multi-user detection, channel coding.

1. Introduction

As it was discussed in [1], channel coding with convolutional code and interleaving for the multi-user Doppler-RAKE detection offered improved performance. However, due to the length of the code, the data rate of the transmission is decreased. A method for increasing the data rate, while keeping the Doppler gain for the system, is to make the channel more time selective as discussed in [2, 3]. In this paper, we propose the signal dividing process, which helps to increase the data rate. The advantage is that this process has a realization similar to multi-user detection, which is already applied in the system. Therefore, our solution is more general and easier for system application and calculation.

2. System representation

A mobile wireless channel can be generally described as a time-varying linear system. The base-band signal $r(t)$ at the receiver is given by

$$r(t) = s(t) + n(t) = \int_0^{\infty} h(t,\tau)\, x(t-\tau)\, d\tau + n(t), \quad (1)$$

where $x(t)$ is the transmitted base-band signal, $h(t,\tau)$ is the time-varying channel impulse response and $n(t)$ is the zero-mean, complex, circular additive white Gaussian noise (AWGN) with power spectral density N_o.

An equivalent representation of the channel, in terms of the spreading function is defined as:

$$H(\theta,\tau) \stackrel{def}{=} \int h(t,\tau) e^{-j2\pi\theta t} dt \quad (2)$$

and

$$s(t) = \int_0^{Tm} \int_{-Bd}^{Bd} H(\theta,\tau)\, x(t-\tau)\, e^{j2\pi\theta\tau} d\delta\, d\tau, \quad (3)$$

where Tm is the multi-path spread and Bd is the Doppler spread (one-sided) of the channel.

The time-varying impulse response $h(t,\tau)$ is best modeled by a wide-sense stationary uncorrelated scatterer (WSSUS) channel. The received signal consists of a linear combination of time-shifted and frequency-shifted (Doppler) copies of the transmitted signal. Its finite-dimensional representation is:

$$s(t) \approx \frac{T_c}{T_s} \sum_{n=0}^{N} \sum_{k=-K}^{K} H\left(\frac{k}{T_s}, nT_c\right) x_{k,n}(t), \quad (4)$$

where $N = [Tm/T_c] \approx [TmB]$, $K = [BdT_s]$, T_c is the chip interval of spread codes, B is the signal bandwidth, T_s is the symbol duration.

By virtue of orthogonality of the basis waveforms $x_{k,n}(t)'s$, and the statistical independence of the channel coefficients $H(\theta,\tau)$, the representation (4) effectively decomposes the channel into $(N+1)\times(2K+1)$ independent, flat-fading (diversity) channels by appropriately sampling the multi-path Doppler plane.

Note that the number of diversity channels is proportional to the product – $TmBd(T_s/T_c)$. Thus, for fixed channel parameters – Tm and Bd – the level of diversity is proportional to the time-bandwidth product (TBP), $T_sB \approx (T_s/T_c)$, of the signal waveform. This also illustrates the remarkable ability of CDMA systems with spread-spectrum signals to exploit channel diversity.

Based on the concept described above, the detector structure (time-frequency (TF) RAKE receiver) for joint multi-path Doppler diversity is developed in [4, 5], which consists of a bank of conventional RAKE receivers shifted in time and frequency to take samples. These samples are combined (by the maximum ratio combining (MRC) method) to estimate the transmitted signal. Analytical results demonstrate that even the small Doppler spreads encountered in practice can be leveraged into significant diversity gains in the proposed Doppler-RAKE detection system.

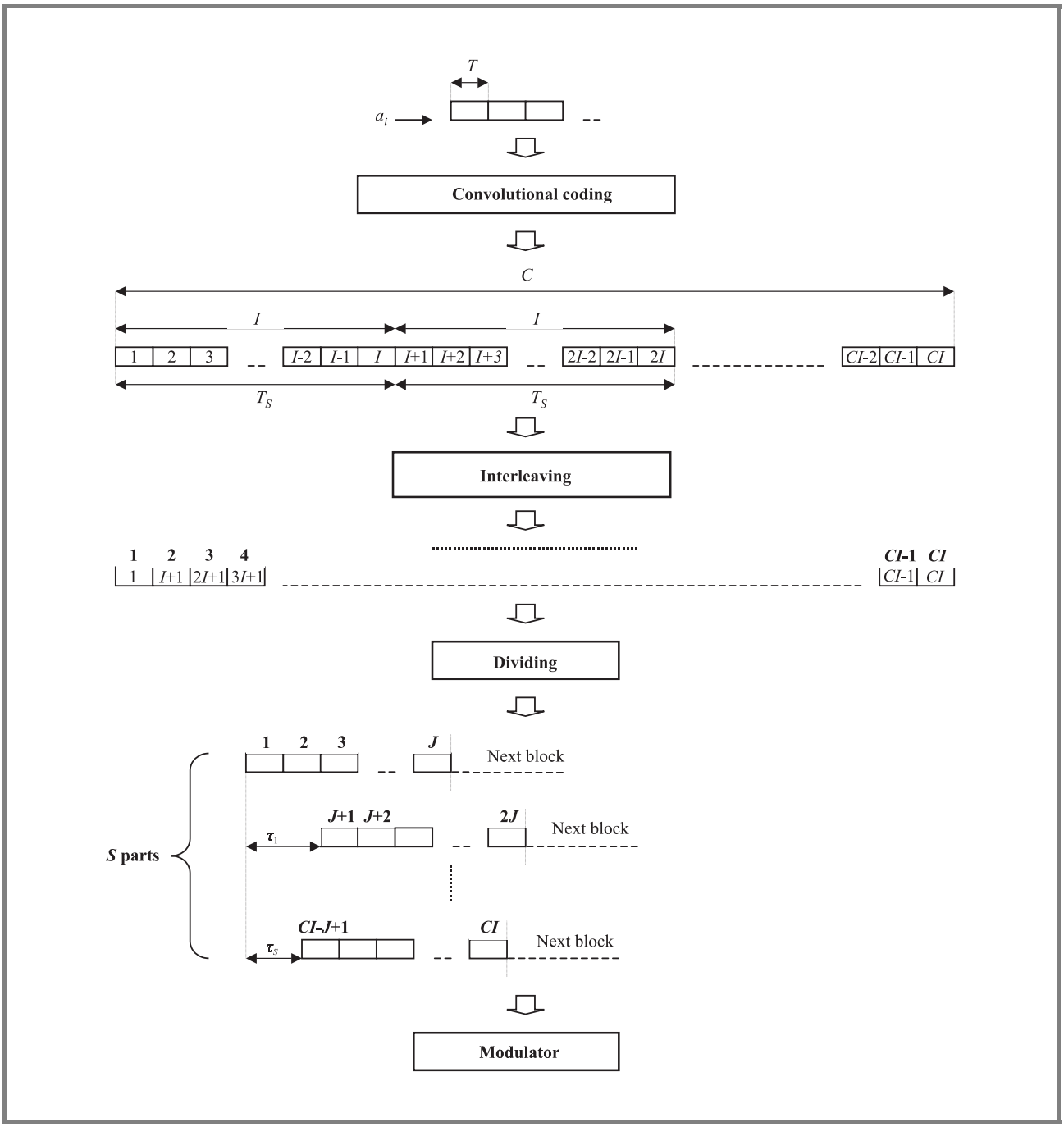


Fig. 1. Divide the symbols to increase data rate.

Also, because the system's performance strongly relies on channel estimation accuracy, the application of channel coding (i.e. convolutional coding and interleaving) has proven to have useful results. Channel coding mitigates the estimation errors and increases Doppler diversity gain.

3. System signaling

The symbol duration for the system without channel coding is T. After channel coding the symbols are spread, and their duration now is $T_s(> T)$. According to formula (4), for maximal exploitation of channel diversity, T_s should be increased as much as possible. Our intention is to make the desired inter-symbol duration T', which determines the data rate, smaller than T_s. So, the basic idea is to divide the symbol block after channel coding into several parts and send them in parallel. We could use regulated time shifts τ_s between parts to gain Doppler effects and channel coding to prevent interference. Figure 1 illustrates the process. During the channel coding, the symbols from the source are first coded by convolutional code of length I,

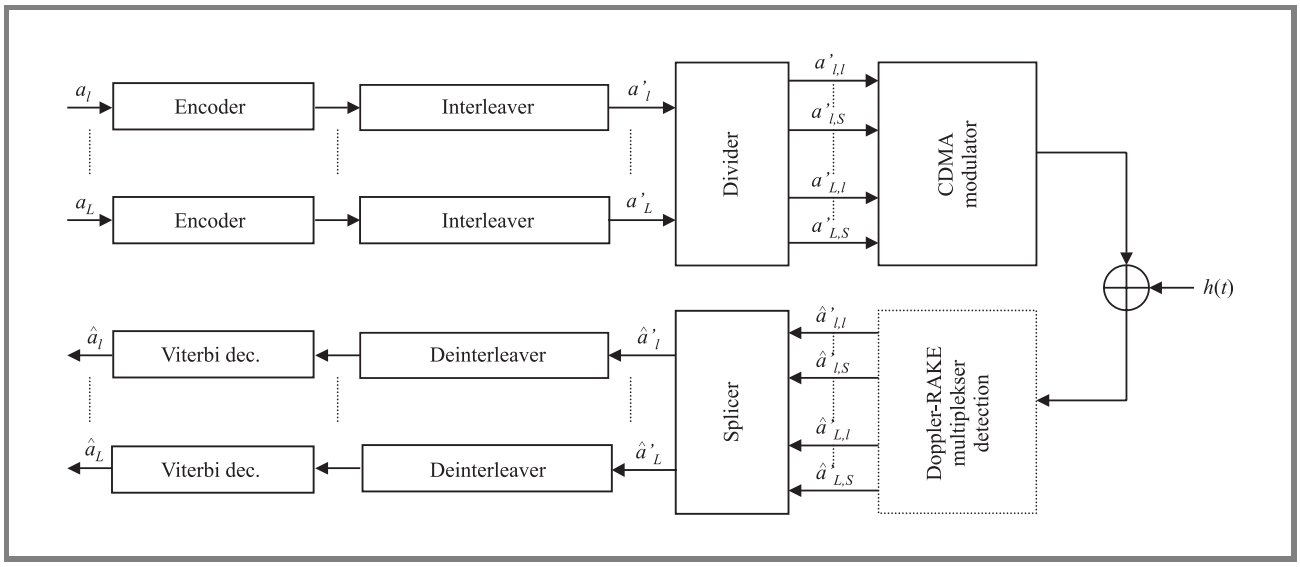


Fig. 2. Multi-user Doppler-RAKE detection structure.

then every block of C coded symbols is interleaved. In the dividing process, each of these blocks then is divided into S parts of J bits and sent in parallel. The inter-symbol duration T' in this method is (assuming that $\tau_s \ll T_s$):

$$T' \cong \frac{T_s}{S} = \frac{IT}{S}. \tag{5}$$

So, if $S \geq I$ we have the data rate, after dividing, comparable with the data rate from the source (i.e. the data rate before channel coding with symbol duration T):

$$R' = \frac{1}{T'} \geq R = \frac{1}{T}. \tag{6}$$

Even in the case that $R' > R$ we cannot make the transmission faster than the case without channel coding, because the maximal data rate is R, due to the relevant signal transmission speed from the source. So we are only interested in the case where $T_s < T' \leq T$ (i.e. $1 < S \leq I$). The symbol duration after dividing is: $T'_s = T_s I/J$ and $T'_s > T$. This retains the Doppler effect. Also, the choice of S, J and τ_s (as well as C[1] if needed) depends on the interference between signals, interleaving and Doppler effects. This provides many variances in the dividing process that need to be analyzed and compared to find the optimal solution. The time-selective signaling method mentioned above, with parameters: $\tau_s = T$, $J = I$, and $S = I$, is one of the cases which has no interleaving processes.

After Doppler-RAKE detection, the signals are rejoined by the splicing process, and then de-interleaved and decoded by the Viterbi algorithm [6].

[1]Normally, C is dependent on the effect of data interleaving and should be as small as possible because it decreases data rate.

Detection technique. For easier representation, consider a CDMA system with L users and employing binary phase-shift key (BPSK) signaling. The signal at the receiver is:

$$r(t) = s(t) + n(t) = \sum_{l=1}^{L} \sum_{s=1}^{S} a'_{l,s} s_{l,s}(t-\tau) + n(t), \tag{7}$$

where $a'_{l,s} \in \{-1, 1\}$ is the bit sign and $s_{l,s}(t)$ is the received baseband for the sth part of the lth user; $n(t)$ is an AWGN. $s_{l,s}(t)$ can be expressed as:

$$s_{l,s} \approx \frac{T_c I}{T_s J} \sum_{n=0}^{N} \sum_{k=-K}^{K} H_{l,s}\left(\frac{k}{K}, nT_c\right) x_{l,s}^{k,n}(t-\tau_s). \tag{8}$$

Here $x_{l,s}^{k,n}(t)$ is the spread waveform of the lth user, $H_{l,s}(k/K, nT_c)$ is the channel coefficient, $N = [Tm/T_c] \approx [TmB]$ and $K = [BdT_s I/J] = [BdT'_s]$, where T_s and T'_s are symbol durations, respectively, before and after dividing, (if $I = J$ then $T'_s = T_s$).

Multi-user Doppler-RAKE detection was described in [5], though we now have LxS users instead of L users. Figure 2 shows the proposed detection scheme. The Doppler-RAKE detector's structure (dotted rectangle) is similar to that mentioned above in Section 2. For multi-user cases, a detection process was added to combat multi-access interference (MAI). This process has many solutions that were discussed in [5]. Because of increased complexities within the system, due to the data-dividing process, we applied our PIC (parallel interference cancellation) method to this case. The PIC solution has offered positive results while its calculation was much simpler than the others.

4. Performance analysis

As discussed above, the system's performance is similar to the case described in [5] but with a higher number of users (*LxS* instead of *L*). While the Doppler effect remains nearly the same, the data rate is increased. This is achieved at a relatively minor cost of encountering more multi-user interference and more complex calculations, each increasing proportionally to the rise in the number of users.

For numerical simulation, the time-varying channel is simulated using the Jakes model [7] corresponding to a data rate $(R = R')$ of 2 kHz (i.e. 2000 symbols/s) and a carrier frequency of 1.8 GHz. Here a low data rate is chosen to show the fast-fading effect. Also, in practice, a longer code length will cause a lower data rate. The code length is 64. A system of two multi-paths $(N = 1)$ and three Doppler paths $(K = 1)$ with four users $(L = 4)$ is applied. Tests are over 10 000 symbols for each variance. The TBd's are 0.1 and 0.2 corresponding to Doppler spread *Bd* 200 Hz and 400 Hz, mobile speed 120 km/h and 240 km/h. Fast-fading channels encountered in practice exhibit Doppler spreads on the order of 100-200 Hz $(TBd = 0.1)$ due to the relative movement of the users (60-120 km/h). The parameters used for the convolutional code are: constraint length $m = 3$, code rate $Rc = 1/2$ $(n = 2)$, free distance $d_f = 5$. The interleaving parameters are: $I1 = 2$, $C1 = 2$ (intended choice) and $I2 = 7$, $C2 = 3$ (accidental choice). The dividing parameters are chosen to keep the data rate the same as that after convolutional coding. Also, a pilot-based estimation process is used for channel estimation [8].

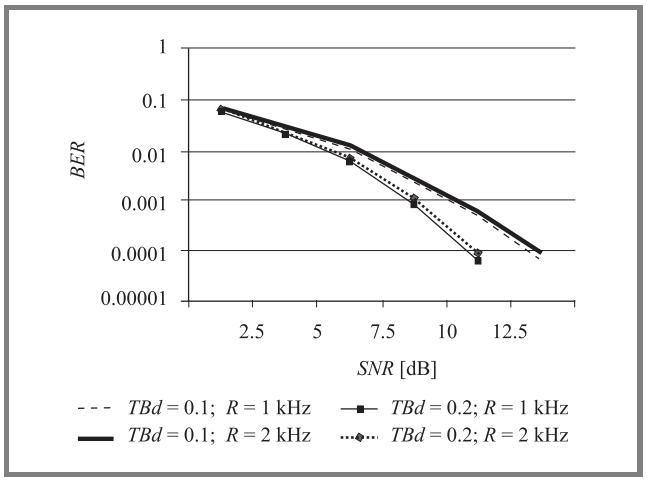


Fig. 3. BER as a function of SNR_user1 (of 4 user-Doppler-RAKE system).

The performance results of the two choices of interleaving are nearly the same. We surmise that this is because of limiting the parameter values for simulation simplification. But the main results, which verify our intentions, are shown.

As we can see, the simulation shows that the performance of the system is nearly as efficient as that of the system without the dividing process (see Fig. 3), and we have a gain in data rate (in this case – 2 times higher). Also, it can be seen that with more Doppler effect, the system out-performs the conventional RAKE system. In Fig. 4, when TBd is increased from 0.1 to 0.2, the efficiency is more obvious: at $BER = 10^{-4}$, Doppler -RAKE system gains about 3 dB of *SNR*.

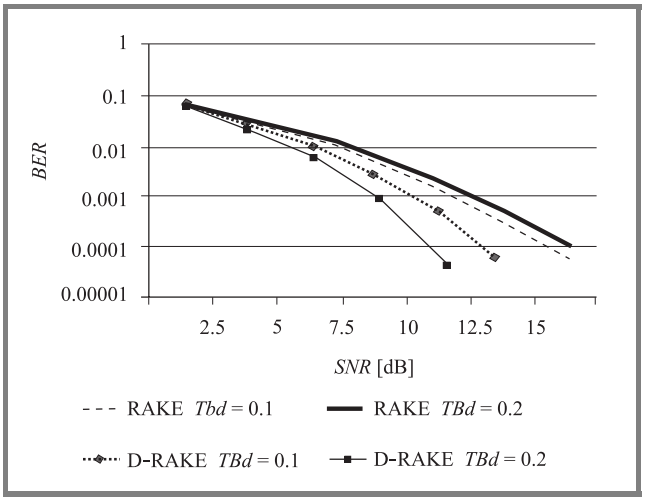


Fig. 4. BER as a function of SNR_user1 (of 4 user conventional RAKE and Doppler (D)-RAKE systems).

5. Conclusions

Interleaving is beneficial in cases where several consecutive bits are damaged. Our solution is more general in dealing with these situations than the time-selective case provided in [2].

Channel coding (including convolutional coding and interleaving) is an indispensable process of the Doppler-RAKE system. By taking into account advantage of the code and block code length of these processes, we can increase both Doppler effect and data rate. With good multi-user detection methods, as were provided, the downside effects of increased signal interference and more complex calculations will have an insignificant impact on performance results. Our analysis is simplified because of the complexity of the calculations. In the next step, more practical propagation model will be needed for more thorough investigation. We offer that our resolution can make the Doppler-RAKE system work better.

References

[1] N. N. Minh and J. Modelski, "Channel coding for multiuser Doppler-RAKE detection" in *Proc. XXVI Krajowa Konferencja Elektroniki i Telekomunikacji Studentów i Młodych Pracowników Nauki*, Warsaw, Poland, 2002.

[2] S. Bhashyam, A. M. Sayeed, and B. Aazhang, "Time-selective signaling and reception for communication over multipath fading channels", *IEEE Trans. Commun.*, vol. 48, no. 1, 2000.

[3] N. N. Minh and J. Modelski, "Multiuser detection of time-selective signaling in fast-fading multipath environments", *Viet. Posts Telecomm. J.*, pp. 21–27, 2001.

[4] A. M. Sayeed and B. Aazhang, "Joint multipath-Doppler diversity in mobile wireless communications", *IEEE Trans. Commun.*, vol. 47, no. 1, 1999.

[5] N. N. Minh and J. Modelski, "MUD and IC for multiuser Doppler-RAKE detection" in *Proc. NATO Reg. Conf. Milit. Commun. Inform. Sys.*, Zegrze, Poland, 2002.

[6] "Coding and encryption in telecommunication", http://users.utu.fi/samnuutt/CET

[7] W. C. Jakes, *Mobile Radio Propagation Microwave Mobile Communications*. Wiley, 1974.

[8] N. N. Minh and J. Modelski, "Pilot-based channel estimation for multiuser time-frequency RAKE detection", *Vietn. DGPT-Post Telecommun. Inform. Technol.*, Dec. 2001.

Minh Nguyen Nguyen received his M.Sc. degree in radioelectronics from the Institute of Radioelectronics, Warsaw University of Technology, in 1999. Currently he is working toward the Ph.D. degree under supervision of Prof. Józef Modelski. His research interests include: digital radio radio transmission, modulation and demodulation techniques for third generation communication systems.
e-mail: nguyenminh@poczta.onet.pl
Institute of Radioelectronics
Warsaw University of Technology
Nowowiejska st 15/19
00-655 Warsaw, Poland

Józef Modelski received the M.Sc., Ph.D. and D.Sc. (habilitation) degrees in electronics from Warsaw University of Technology in 1973, 1978, and 1987, respectively. In 1994 he obtained the State title of Professor. Since 1973 he has been with the Institute of Radioelectronics, Warsaw University of Technology (WUT) holding in sequence all academic positions from teaching/research assistant to tenured professor (1991). In 1976/77 he spent 13 months in U.S. as a Fulbright grantee working with the microwave laboratories: at the Texas University at Austin, Cornell University, and Communication Satellite Corporation COMSAT. In 1986 he joined for two years the Braunschweig Technical University (Germany) as a senior scientist. His research interests are in the areas of microwave phase modulators and shifters with semiconductor and ferrite elements, dielectric resonators and their applications, integrating waveguide technology for mass-production components, and design of communications antennas. He has published over 150 technical papers and 4 books, obtained 8 patents and many awards. Since 1996 J. Modelski has been a Director of the Institute of Radioelectronics, WUT. He is chairing scientific boards in two research centers and International Microwave Conference MIKON, is a member of the TPCs of MTTS International Microwave Symposium, European Microwave Conference and many local conferences. He is IEEE Fellow Member.
e-mail: j.modelski@ire.pw.edu.pl
Institute of Radioelectronics
Warsaw University of Technology
Nowowiejska st 15/19
00-655 Warsaw, Poland

INFORMATION FOR AUTHORS

The *Journal of Telecommunications and Information Technology* is published quarterly. It comprises original contributions, both regular papers and letters, dealing with a broad range of topics related to telecommunications and information technology. Items included in the journal report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

The *Journal* is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, speech devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology. Suitable research-related manuscripts should hold the potential to advance the technological base of telecommunications and information technology. Tutorial and review papers are published by invitation only.

Papers published by invitation and regular papers should contain up to 15 and 8 printed pages respectively (one printed page corresponds approximately to 3 double-space pages of manuscript, where one page contains approximately 2000 characters).

Manuscript: An original and two copies of the manuscript must be submitted, each completed with all illustrations and tables attached at the end of the papers. Tables and figures have to be numbered consecutively with Arabic numerals. The manuscript must include an abstract limited to approximately 100 words. The abstract should contain four points: statement of the problem, assumptions and methodology, results and conclusion, or discussion, of the importance of the results. The manuscript should be double-spaced on only one side of each A4 sheet (210 × 297 mm). Computer notation such as Fortran, Matlab, Mathematica etc., for formulae, indices, etc., is not acceptable and will result in automatic rejection of the manuscript. The style of references, abbreviations, etc., should follow the standard IEEE format.

References should be marked in the text by Arabic numerals in square brackets and listed at the end of the paper in order of their appearance in the text, including exclusively publications cited inside. **The reference entry** (correctly punctuated according to the following rules and examples) **has to contain**.

From journals and other serial publications: initial(s) and second name(s) of the author(s), full title of publication (transliterated into Latin characters in case it is in Russian, possibly preceded by the title in Russian characters), appropriately abbreviated title of periodical, volume number, first and last page number, year. E.g.:

[1] Y. Namihira, "Relationship between nonlinear effective area and modefield diameter for dispersion shifted fibres", *Electron. Lett.*, vol. 30, no. 3, pp. 262–264, 1994.

From non-periodical, collective publications: as above, but after title – the name(s) of editor(s), title of volume and/or edition number, publisher(s) name(s) and place of edition, inclusive pages of article, year. E.g.:

[2] S. Demri, E. Orłowska, "Informational representability: Abstract models versus concrete models" in *Fuzzy Sets, Logics and Reasoning about Knowledge*, D. Dubois and H. Prade, Eds. Dordrecht: Kluwer, 1999, pp. 301–314.

From books: initial(s) and name(s) of the author(s), place of edition, title, publisher(s), year. E.g.:

[3] C. Kittel, *Introduction to Solid State Physics*. New York: Wiley, 1986.

Figure captions should be started on separate sheet of papers and must be double-spaced.

Illustration: Original illustrations should be submitted. All line drawings should be prepared on white drawing paper in black India ink. Drawings in Corel Draw and Postscript formats are preferred. Colour illustrations are accepted only in exceptional circumstances. Lettering should be large enough to be readily legible when drawing is reduced to two- or one-column width – as much as 4:1 reduction from the original. Photographs should be used sparingly. All photographs must be gloss prints. All materials, including drawings and photographs, should be no larger than 175 × 260 mm.

Page number: Number all pages, including tables and illustrations (which should be grouped at the end), in a single series, with no omitted numbers.

Electronic form: A floppy disk together with the hard copy of the manuscript should be submitted. It is important to ensure that the diskette version and the printed version are identical. The diskette should be labelled with the following information: a) the operating system and word-processing software used, b) in case of UNIX media, the method of extraction (i.e. tar) applied, c) file name(s) related to manuscript. The diskette should be properly packed in order to avoid possible damage during transit.

Among various acceptable word processor formats, TEX and LATEX are preferable. The *Journal's* style file is available to authors.

Galley proofs: Proofs should be returned by authors as soon as possible. In other cases, the article will be proof-read against manuscript by the editor and printed without the author's corrections. Remarks to the errata should be provided within two weeks after receiving the offprints.

The copy of the "Journal" shall be provided to each author of papers.

Copyright: Manuscript submitted to this journal may not have been published and will not be simultaneously submitted or published elsewhere. Submitting a manuscript, the authors agree to automatically transfer the copyright for their article to the publisher if and when the article is accepted for publication. The copyright comprises the exclusive rights to reproduce and distribute the article, including reprints and also all translation rights. No part of the present journal may be reproduced in any form nor transmitted or translated into a machine language without permission in written form from the publisher.

Biographies and photographs of authors are printed with each paper. Send a brief professional biography not exceeding 100 words and a gloss photo of each author with the manuscript.

National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Editorial Office

tel. +48(22) 872 43 88
tel./fax: +48(22) 512 84 00
e-mail: redakcja@itl.waw.pl
http://www.itl.waw.pl/jtit