We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



17

Some Special Traveling Salesman Problems with Applications in Health Economics

Liana Lupşa[1], Ioana Chiorean[1], Radu Lupşa[1] and Luciana Neamţiu[2]
[1]*Babeş-Bolyai University*
[2]*Cancer Institute Ion Chiricuţă*
Romania

1. Introduction

Starting from a practical *Health Economics* problem (the optimal planning of visits for a given medical registrar to its allocated cities, for selection and registration of eligible patients to be included in the regional non-communicable diseases registries for a specified type of chronic disease) we construct a mathematical model. We show that this model can be seen as a new generalization of the following problems: Prize Collecting Traveling Salesman Problem, Simple Cycle Problem, Capacitated Prize-Collecting Traveling Salesman Problem or Orienteering Problem.
Our purpose is that of finding a cycle, belonging to a directed graph, with a given number of vertices (nodes), among which one is fixed, so that the total bonus (which varies in time) is maximized and the total costs (transport costs plus accommodation costs) are minimized. A boundary condition must also be satisfied.
In contrast to the known generalizations of the Traveling Salesman Problem, the originality of our approach relies on three ideas:
- the fact that the exact number of cycle vertices is fixed,
- the bonus depends not only on the visited vertex, but also on the visiting time, that is, on the position of the vertex along the cycle, and
- the goal is expressed by a vector, with two components (bonus and cost).

The solving of the problem reduces to the determination of a lexicographic max-min non-dominant cycle, the choice of the lexicographic order being determined by the initial health economics problem.
It is important to mention that we do not just analyze the problem from a theoretical point of view, but also from a practical one, therefore we propose two algorithms: a Greedy algorithm and an exact one. They both generate good solutions. Unfortunately, the exact algorithm becomes slower as the number of vertices increases. That is, as in (Trobec et al., 2009), we also analyze the possibility of using parallel calculus for improving the execution time.
We are interested in continuing the research, focused on other health economics problems, whose mathematical model can be seen as a generalization of the classical Traveling Salesmen Problem.

2. The medical motivational background

During the last years, many studies belonging to a field known as *Health Economics Research*, have been developed. The term includes a multitude of issues related to the management of pharmaceutical products and of medical activities, from an optimization perspective. In this context, several major issues arise:

- finding new treatments with increased effectiveness and with little costs;
- making the treatment to be carried out to a patient as bearable as possible (little inconvenience, fewer side effects);
- detecting, in early stages, the disease, since it is known that healing is safer, faster and less expensive as the disease is diagnosed sooner;
- monitoring a chronic disease both for determining its objective causes, and for understanding its evolution trend, in order to anticipate correctly the necessary treatment costs;
- finding those treatments which may keep the patient's life quality at a high level, but with lower costs (usually constrained to fit into a given budget), in chronic diseases, when returning to the normal health state is impossible;
- preventing infectious diseases, by controlling the possible contacts, isolating the sick persons, and, in some cases vaccinating them.

The basic idea behind pharmacoeconomics studies is: *to win as a good health as possible with the smallest amount of money*. Therefore, in these studies which consider a treatment by its results reported to the cost, some ratios are generated, called indices. Among them, we recall ICER, INB, and NHB. It is clear that, in order to draw a conclusion closer to reality on the costs and effectiveness of a treatment, it is necessary to apply it to a sufficient number of patients. As a result, costs and effects will be replaced by average costs, average effects, respectively. The need of averages leads to the necessity of using results from mathematical statistics. Hence, if in their early phases, the pharmacoeconomics studies highlighted the treatment effects and identified the costs involved in applying it (the actual cost of the drugs used, their application costs, costs of the medical advices), nowadays these studies have been starting to use advanced mathematical tools like Probability Theory, Bayesian Analysis, Markov Processes and Multi-Criteria Optimization. We recall (Briggs & Sculpher, 1998; Briggs et al., 2006; Fayers, 1997) and (Willan & Briggs, 2006) as main reference works related to pharmacoeconomics studies, works that use the indices mentioned above.

Because health budgets are not unlimited, whatever the country, the health economics problems are, in most cases, optimization problems which get a multi-criteria character (see, for example, (Lupsa, 1999; 2000a;b; Tigan et al., 2001; Grosan et al., 2005; 2007)). Other works, as (Canfelt et al., 2004; Kang & Lagakos, 2007; Chiorean et al, 2008; Lupsa et al, 2008; Neamtiu, 2008; 2009), contain mathematical models which were built by using Bayesian Analysis, Markov Processes or Dynamic Optimization.

The aim of this chapter is to present an approach to a health-economics problem, using in its solving variants of the Traveling Salesman Problem. The problem was raised during the optimization process of the data collection for the cancer registry at the North-West Regional Cancer Registry (Romania). The research is sustained by a multidisciplinary consortium in the framework of CRONIS project (contract no. 11-003/2007, developed under PN II national R&D programme, financed by the Romanian Government).

3. The problem of medical registrars

To organize regional non-communicable diseases registries for a specified type of chronic disease (cancer, diabetes, cardiovascular), registrars (persons with specialized training who register data) must visit, at certain interval of times, the medical units subordinated (clinics, hospitals, etc.) for selection and registration of eligible patients to be included in the register. More precisely, registrars, based on medical documentation, have to decide whether a case should be registered or not, and, in the first situation, to specify, beside the patient's personal data, some disease-related remarks, remarks that are coded. For example, for cancer, the registration and coding rules are well established and standardized (see (World Health Organization, 1991)).

Knowing those medical units which are subordinated to a registrar, the problem is to plan the days in which he has to make the registrations at every medical unit. The maximum number of registrations that he/she can make in one day is known, as well as the fact that, in one day, he/she visits one medical unit, due to the fact that these units are in different locations (cities). The number of patients whose medical records have to be investigated is variable, it increases from one day to another. Specifically, every morning, in the stack of processed medical records, new ones are added. One makes the assumption that, at the very beginning, the number of existing medical records is known and also its growth rate, depending of time. In this paper we assume that the rate is constant, and equal to the monthly average number of new files. Planning should be such that the number of medical records left unchecked (from all the files gathered until the end of the period) to be as low as possible, and also the total costs (which include transportation costs plus subsistence and accommodation costs) to be as low as possible. The registrar may remain for several days to record documents in one medical unit. He/she can move from one unit to another or return to the base, case in which he/she is obliged to remain a day there.

The following data are known:

- n, the number of hospitals, numbered from 1 to n; with 0 being numbered the hospital to which the registrar belongs; it is the location where he starts the early registration period and where he must return at the end of the recording process;
- p, the number of days corresponding to the time period when the registration is made;
- c_{ij}, transport costs from hospital i to hospital j, for any $i, j \in \{0,1,\ldots,n\}$, $i \neq j$;
- q, maximum number of medical records that can be investigated by the registrar in a day;
- d_i, accommodation costs per night if remains to sleep overnight in the area corresponding to hospital i (if it coincides with the place where hospital 0 is located, we have obviously $d_i = 0$);
- f_i, estimated number of medical records that can exist in hospital i at the beginning of the work;
- r_i, the growth rate, from day to day, of the number of medical records;
- s, daily subsistence if, in that day, the registrar is located in another location than that where the hospital to which he belongs is;
- δ, the working number of hours/days.

We use the notation $T = \{1,\ldots,p\}$.

One of the difficulties in solving this problem is the fact that there are hospitals where the number of medical records to be processed exceeds the processing capacity of the registrar in a day. In that case, he has to work more than a day at that site and we must decide which is better for him, to stay over night there or to go back home and return the next day. Another difficulty arises from the fact that the total number of medical records to be processed at a site increases from one day to the next. To treat these difficulties, we introduce dummy (fictive)

hospitals, as it will be seen next.
Let us consider hospital i. The total number of unprocessed medical records that exist until the morning of day k is $f_i + (k-1)r_i$. To process the files collected until the morning of the last day (the p-th day), it would be necessary m_i working days, where

$$m_i = \begin{cases} \dfrac{f_i + (p-1)r_i}{q}, \text{ if } \dfrac{f_i + (p-1)r_i}{q} \in \mathbb{N}, \\ \left[\dfrac{f_i + (p-1)r_i}{q}\right] + 1, \text{ if } \dfrac{f_i + (p-1)r_i}{q} \notin \mathbb{N}. \end{cases} \tag{1}$$

Therefore, for the sake of simplicity, we consider that, instead of hospital i, we have m_i hospitals, denoted by $i_1,\ldots,i_{m_i}$. All the files of these hospitals come from the hospital i; the number of files in these hospitals will vary in time, but none of them will exceed the maximum number of files that can be processed in one day. More precisely, the moment when, at the hospital i_h, q files have been gathered for processing, the subsequent files are considered to be at hospital i_{h+1}; when for hospital i_{h+1}, q files are gathered, the next ones are sent to hospital i_{h+2}, etc.
We denote by $g_{i_h,k}$ the number of files to be processed, existing in hospital i_h in the day $k \in T$. This is equal to

$$g_{i_h,k} = \begin{cases} f_i + (k-1)r_i, \text{if } h = 1, f_i + (k-1)r_i \leqq q, \\ q, \text{ if } h = 1, f_i + (k-1)r_i > q, \\ 0, \text{ if } 1 < h, f_i + (k-1)r_i - (h-1)q \leqq 0, \\ f_i + (k-1)r_i - (h-1)q, \text{ if } 1 < h, 1 \leqq f_i + (k-1)r_i - (h-1)q \leqq q, \\ q, \text{ if} 1 < h, f_i + (k-1)r_i - (h-1)q > q. \end{cases} \tag{2}$$

The distance between any two dummy hospitals generated by hospital i is equal to d_i (instead of transport, the accommodation has to be paid), and the distance between any dummy hospital i_h and hospital k, $k \neq i$, is equal to the distance between hospital i and hospital k. Then, the following relation holds:

$$c_{i_h,k} = \begin{cases} c_{ik}, \forall\, h \in \{1,\ldots,m_i\},\ k \in \{0,1,\ldots,n\} \setminus \{i\}, \\ d_i, \forall\, h \in \{1,\ldots,m_i\},\ k \in \{i_1,\ldots,i_m\},\ i_h \neq k. \end{cases} \tag{3}$$

Following this transformation, the problem size increases by increasing the number of hospitals, but the problem is simplified by the fact that every day, the registrar can examine all existing files in any of the hospitals where he arrives. We denote by m the number of new hospitals resulting by performing this transformation (i.e. $m = m_1 + \cdots + m_n$).
We continue to denote by c_{ik} the transport costs between any two hospitals i and k, $i \neq k$. Since there may be days when the registrar does not work in the subordinated hospitals, we introduce more p dummy hospitals, numbered $m+1,\ldots,m+p$.
The transport costs for the new hospitals are defined by

$$c_{ik} = \begin{cases} 0, \text{ if } i,k \in \{m+1,\ldots,m+p\}, i \neq k, \\ c_{0k}, \text{ if } i \in \{m+1,\ldots,m+p\},\ k \in \{1,\ldots,m\}, \\ 0, \text{ if } i = 0,\ k \in \{m+1,\ldots,m+p\}, \\ c_{i0}, \text{ if } i \in \{1,\ldots,m\},\ k \in \{m+1,\ldots,m+p\}. \end{cases} \tag{4}$$

Every day, the number of existing medical records in the location $m+1,\ldots,m+p$ is equal to 0, hence we have

$$g_{ij}=0,\ \forall\, i\in\{m+1,\ldots,m+p\},\ j\in\{1,\ldots,p\}.$$

The accommodation costs will be also equal to 0, so

$$d_i=0,\ \forall\, i\in\{m+1,\ldots,m+p\}.$$

Because, under the new conditions, the registrar will not stay more than one day in a hospital, we attach the subsistence costs to the hospital. More precisely, we define the numbers s_i, $i\in\{1,\ldots,m+p\}$, by $s_i:=s$, if the hospitals i and 0 are not at the same location and $s_i:=0$ otherwise.

Hence, after making these changes, we consider that we have $m+p+1$ hospitals, from 0 to $m+p$, with 0 being the "home" hospital (the registrar's home). The following data are known:

- transport costs between any two locations $i,k\in\{0,1,\ldots,m+p\}$, $i\neq k$, denoted by c_{ik},
- number of medical records that can be processed in the hospital i in every day $j\in T$, denoted by g_{ij},
- the accommodation cost, d_i, and the subsistence cost, s_i, according to the area where the hospital i is located.

Example. Let us take $n=3$, $p=4$, $q=10$, $e=14$, $d_1=3$, $d_2=7$, $d_3=2$, $s=1$,

$$C=\begin{bmatrix}0&5&7&3\\5&0&2&4\\7&2&0&6\\3&4&6&0\end{bmatrix},$$

$$f_1=20,\ f_2=15,\ f_3=1,\ r_1=2,\ r_2=1,\ r_3=3.$$

According with the previous specification, we work with $m=3+2+1$ and $p=4$. In total, 10 hospitals, plus the "home" hospital, 0. The transport costs between any two hospitals are given in the table 1.

	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
S_0	0	5	5	5	7	7	3	0	0	0	0
S_1	5	0	0	0	2	2	4	5	5	5	5
S_2	5	0	0	0	2	2	4	5	5	5	5
S_3	5	0	0	0	2	2	4	5	5	5	5
S_4	7	2	2	2	0	0	6	7	7	7	7
S_5	7	2	2	2	0	0	6	7	7	7	7
S_6	3	4	4	4	6	6	0	3	3	3	3
S_7	0	5	5	5	7	7	3	0	0	0	0
S_8	0	5	5	5	7	7	3	0	0	0	0
S_9	0	5	5	5	7	7	3	0	0	0	0
S_{10}	0	5	5	5	7	7	3	0	0	0	0

Table 1. Transportation costs table

The medical records matrix will be

$$G = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 0 & 2 & 4 & 6 \\ 10 & 10 & 10 & 10 \\ 5 & 6 & 7 & 8 \\ 1 & 4 & 7 & 10 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The new vectors d and s will be

$$d = (3,3,3,7,7,2,0,0,0,0),$$

$$s = (1,1,1,1,1,1,0,0,0,0).$$

Our purpose is to obtain a planning for the registrar (this means to specify, for every day $j \in T$, the hospital where the registrar should go) such as the number of medical records that are processed to be as large as possible, and the costs involved in transportation, accommodation and subsistence to be as low as possible. The following assumptions are made:

1) If in the day j, $j \in T$, the registrar works at the hospital i, $i \in \{1,\ldots,m+p\}$, then in the following night he/she sleeps in the city where hospital i is located. Therefore, the total costs for that day contain the transportation costs from the location where the registrar worked in the previous day and the city i, the accommodation costs for one night in locality i, (equal to d_i) and the subsistence costs for locality i (equal to s_i).
2) The departure of the registrar is soon enough, so that, after reaching the destination, he may work δ hours.
3) The first day, the registrar leaves from hospital 0.
4) The registrar returns to hospital 0 only in the morning of day $p+1$.
5) One day, the registrar works exactly in one hospital.
6) In every hospital the registrar works at most one day.
7) The available amount of money which can be spent for the entire action is equal to e and it includes also the costs of returning to hospital 0 in the day $p+1$.

4. A mathematical model for the problem of medical registrars

In order to generate the mathematical model, we introduce the binary variables x_{ij}, $i \in \{1,\ldots,m+p\}$, $j \in \{1,\ldots,p\}$, where

$$\begin{cases} x_{ij} = 1, \text{ if in day } j \text{ the registrar is in hospital } i, \\ x_{ij} = 0, \text{ if not.} \end{cases} \tag{5}$$

Conditions 3)-5) imply that

$$\sum_{i=1}^{m+p} x_{ij} = 1, \ \forall\, j \in \{1,\ldots,p\}. \tag{6}$$

Condition 4) implies

$$\sum_{j=1}^{p}\sum_{i=1}^{m+p} x_{ij} = p. \tag{7}$$

Condition 6) means that

$$\sum_{j=1}^{p} x_{ij} \leqq 1, \forall\, i \in \{1,\ldots,m+p\}. \tag{8}$$

Considering the hypotheses 1) and 3), the first day costs will be

$$\sum_{k=1}^{m+p} (c_{0k} + d_k + s_k) x_{k1}.$$

According to hypothesis 4), the costs of returning to hospital 0 in day $p+1$ are

$$\sum_{k=1}^{m+p} c_{k0} x_{kp}.$$

According to 1), the costs for a day $j \in \{2,\ldots,p\}$ are

$$\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} c_{ik} x_{i,j-1} x_{kj} + \sum_{k=1}^{m+p} (d_k + s_k) x_{k,j}.$$

Then, the hypothesis 7) implies

$$\sum_{k=1}^{m+p} c_{0k} x_{k1} + \sum_{j=2}^{p}\left(\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} c_{ik} x_{i,j-1} x_{kj}\right) + \tag{9}$$

$$+ \sum_{j=1}^{p}\sum_{k=1}^{m+p} (d_k + s_k) x_{kj} + \sum_{k=1}^{m+p} c_{k0} x_{kp} \leqq e.$$

The set of matrices $X = [x_{ij}] \in \{0,1\}^{(m+p)\times p}$ which verify (6), (7) and (8) will be denoted by $\mathcal{X}$. The set of matrices $X = [x_{ij}] \in \mathcal{X}$ which verify in addition the condition (8), will be denoted by $\tilde{X}$.

As the registrar may process in one day all existing files in a location, he will do so. Therefore, considering that, in one day $j \in T$, he can be in one location, the number of medical records processed that day will be equal to

$$\sum_{i=1}^{m+p} g_{ij} x_{ij},$$

and the total number of processed medical records is equal to

$$\sum_{j=1}^{p}\sum_{i=1}^{m+p} g_{ij} x_{ij}.$$

The number of medical records that would be gathered at the end of the period is equal to

$$\sum_{i=1}^{m+p} \max\{g_{ij} \,|\, j \in \{1,\ldots,p\}\}.$$

The two objective functions are:
- F_f, which denotes the total processed medical records, i.e. $F_f : \{0,1\}^{(m+p)\times p} \to \mathbb{R}$,

$$F_f(X) = \sum_{i=1}^{m+p} \sum_{j=1}^{p} g_{ij} x_{ij}, \tag{10}$$

for all $X = [x_{ij}] \in \{0,1\}^{(m+p)\times p}$ and
- F_c, which denotes the total cost, i.e. $F_c : \{0,1\}^{(m+p)\times p} \to \mathbb{R}$,

$$F_c(X) = \sum_{k=1}^{m+p} c_{0k} x_{k1} + \sum_{j=2}^{p} \sum_{i=1}^{m+p} \sum_{k=1,k\neq i}^{m+p} c_{ik} x_{i,j-1} x_{kj} + \sum_{j=1}^{p} \sum_{k=1}^{m+p} (d_k + s_k) x_{kj} + \sum_{k=1}^{m+p} c_{k0} x_{kp}, \tag{11}$$

for all $X = [x_{ij}] \in \{0,1\}^{(m+p)\times p}$.
The corresponding mathematical model for the Registrar Problem is the *Max-min Lexicographical Optimization Problem* with the objective function $F = (F_f, F_c) : \{0,1\}^{(m+p)\times p} \to \mathbb{R}^2$, whose scalar components are given by (10), (11), and the feasible set is $\tilde{\mathcal{S}}$,

$$\tilde{\mathcal{S}} = \{X \in \{0,1\}^{(m+p)\times p} \,|\, \text{satisfies the conditions } (6), (7), (8), (9)\}, \tag{12}$$

Hence, our problem, denoted by (PB), is the following:

$$\begin{cases} \left.\begin{array}{rcl} F_f(X) & = & \displaystyle\sum_{i=1}^{m+p} \sum_{j=1}^{p} g_{ij} x_{ij} \\ F_c(X) & = & \displaystyle\sum_{k=1}^{m+p} (c_{0k} + d_k + s_k) x_{k1} \\ & + & \displaystyle\sum_{j=2}^{p} \sum_{i=1}^{m+p} \sum_{k=1,k\neq i}^{m+p} c_{ik} x_{i,j-1} x_{kj} \\ & + & \displaystyle\sum_{j=2}^{p} \sum_{k=1}^{m+p} (d_k + s_k) x_{kj} + \sum_{k=1}^{m+p} c_{k0} x_{kp} \end{array}\right\} \to lex - \max - \min \\ \displaystyle\sum_{i=1}^{m+p} x_{ij} = 1, \forall\, j \in \{1,\ldots,p\}, \\ \sum_{j=1}^{p} x_{ij} \leqq 1, \forall\, i \in \{1,\ldots,m+p\}, \\ \displaystyle\sum_{k=1}^{m+p} (c_{0k} + d_k + s_k) x_{k1} + \sum_{j=2}^{p} \sum_{i=1}^{m+p} \sum_{k=1,k\neq i}^{m+p} c_{ik} x_{i,j-1} x_{kj} + \\ \quad + \displaystyle\sum_{j=2}^{p} \sum_{k=1}^{m+p} (d_k + s_k) x_{kj} + \sum_{k=1}^{m+p} c_{k0} x_{kp} \leqq e, \\ x_{ij} \in \{0,1\}, \forall\, i\{1,\ldots,m+p\}, j \in \{1,\ldots,p\}. \end{cases}$$

Using an appropriate change of variables, the problem (PB) may be transformed in a linear one. It is easy to notice that, if $x, y, z \in \{0,1\}$, then we have $z = x \cdot y$ if and only if z is the unique solution of the system

$$\begin{cases} x + y - z \leqq 1 \\ -x - y + 2z \leqq 0 \\ z \in \{0,1\}. \end{cases} \tag{13}$$

So, for each $j \in \{2,\ldots,p\}$, $i \in \{1,\ldots,m+p\}$ and $k \in \{1,\ldots,m+p\}$, $k \neq i$, we introduce the binary variable

$$z_{ijk} := x_{i,j-1} \cdot x_{kj} \tag{14}$$

and the corresponding conditions

$$\begin{array}{l} x_{i,j-1} + x_{kj} - z_{ijk} \leqq 1, \\ -x_{i,j-1} - x_{kj} + 2z_{ijk} \leqq 0, \\ z_{ijk} \in \{0,1\}. \end{array}$$

Then the problem (PB) can be rewritten as

$$\left\{ \begin{array}{l} \left(\begin{array}{lcl} F_f(X) & = & \sum\limits_{i=1}^{m+p} \sum\limits_{j=1}^{p} g_{ij} x_{ij} \\ F_c(X) & = & \sum\limits_{k=1}^{m+p} (c_{0k} + d_k + s_k) x_{k1} \\ & + & \sum\limits_{j=2}^{p} \sum\limits_{i=1}^{m+p} \sum\limits_{k=1,k\neq i}^{m+p} c_{ik} z_{ijk} \\ & + & \sum\limits_{j=2}^{p} \sum\limits_{k=1}^{m+p} (d_k + s_k) x_{kj} + \sum\limits_{k=1}^{m+p} c_{k0} x_{kp} \end{array} \right) \rightarrow lex - \max - \min \\ \sum\limits_{i=1}^{m+p} x_{ij} = 1, \forall\, j \in \{1,\ldots,p\}, \\ \sum_{j=1}^{p} x_{ij} \leqq 1, \forall\, i \in \{1,\ldots,m+p\}, \\ \sum\limits_{k=1}^{m+p} (c_{0k} + d_k + s_k) x_{k1} + \sum\limits_{j=2}^{p} \sum\limits_{i=1}^{m+p} \sum\limits_{k=1,k\neq i}^{m+p} c_{ik} z_{ijk} + \sum\limits_{j=2}^{p} \sum\limits_{k=1}^{m+p} (d_k + s_k) x_{kj} + \\ \quad + \sum\limits_{k=1}^{m+p} c_{k0} x_{kp} \leqq e, \\ x_{i,j-1} + x_{kj} - z_{ijk} \leqq 1, \forall\, j \in \{2,\ldots,p\}, i \in \{1,\ldots,m+p\}, k \in \{1,\ldots,m+p\}, k \neq i, \\ -x_{i,j-1} - x_{kj} + 2z_{ijk} \leqq 0, \forall\, j \in \{2,\ldots,p\}, i \in \{1,\ldots,m+p\}, k \in \{1,\ldots,m+p\}, k \neq i, \\ x_{ij} \in \{0,1\}, \forall\, i \{1,\ldots,m+p\}, j \in \{1,\ldots,p\}, \\ z_{ijk} \in \{0,1\}, \forall\, j \in \{2,\ldots,p\}, i \in \{1,\ldots,m+p\}, k \in \{1,\ldots,m+p\}, k \neq i. \end{array} \right.$$

This problem can be solved by using any of the algorithms for solving pseudo-boolean linear optimization problems (see, for example (Hammer & Rudeanu, 1968; Crama, 1989)). But, it can be seen as a special type of the Prize Collecting Traveling Salesman Problem or as a special type of the Simple Cycle Problem, as it will be shown in the next section.

5. Max-min lexicographical traveling salesman problem with objective function depending on a parameter

Let m and p be natural numbers, $m \neq 0$, and let e, g_{ij}, a_{kj}, b_{ik}, c_{kj}, d_{ik}, $i \in \{1,\ldots,m+p\}$, $j \in \{1,\ldots,p\}$, $k \in \{1,\ldots,m+p\}$, $k \neq i$, be positive real numbers.
By $\mathcal{X}$ we denote the set of the matrices $X = [x_{ij}]$ with $m+p$ rows and p columns, whose elements satisfy the following three conditions:

$$\sum_{i=1}^{m+p} x_{ij} = 1, \forall\, j \in \{1,\ldots,p\}, \tag{15}$$

$$\sum_{j=1}^{p} x_{ij} \leqq 1, \forall\, i \in \{1,\ldots,m+p\}, \tag{16}$$

$$x_{ij} \in \{0,1\}, \forall\, i \in \{1,\ldots,m+p\},\ j \in \{1,\ldots,p\}, \tag{17}$$

and by $\tilde{\mathcal{X}}$ the set of matrices of $\mathcal{X}$ which satisfies, in addition, the bounded condition

$$\sum_{j=1}^{p}\sum_{k=1}^{m+p} c_{kj}x_{kj} + \sum_{j=2}^{p}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} d_{ik}x_{i,j-1}x_{kj} \leqq e. \tag{18}$$

Let us remark that from (15) we obtain

$$\sum_{i=1}^{m+p}\sum_{j=1}^{p} x_{ij} = p. \tag{19}$$

Then, from (16) it follows that there are p distinct indices $i_1,\ldots,i_p \in \{1,\ldots,m+p\}$ such that

$$\sum_{j=1}^{p} x_{i_k,j} = 1, \forall\, k \in \{1,\ldots,p\},$$

and

$$\sum_{j=1}^{p} x_{ij} = 0, \forall\, i \in \{1,\ldots,m+p\} \setminus \{i_1,\ldots,i_p\}.$$

Let us define the vector function $F = (F_1, F_2) : \mathcal{X} \to \mathbb{R}^2$, where

$$F_1(X) = \sum_{i=1}^{m+p}\sum_{j=1}^{p} g_{ij}x_{ij}, \forall\, X \in \mathcal{X}, \tag{20}$$

$$F_2(X) = \sum_{j=1}^{p}\sum_{k=1}^{m+p} a_{kj}x_{kj} + \sum_{j=2}^{p}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} b_{ik}x_{i,j-1}x_{kj}, \forall\, X \in \mathcal{X}. \tag{21}$$

The problem (PB) can be see as a special case of the pseudo-boolean lexicographical max-min optimization problem with the objective function F and the set of feasible solution equal to $\tilde{\mathcal{X}}$, i.e., the problem:

$$(PBG)\quad \begin{cases} \begin{pmatrix} \sum_{i=1}^{m+p}\sum_{j=1}^{p} g_{ij}x_{ij} \\ \sum_{j=1}^{p}\sum_{k=1}^{m+p} a_{kj}x_{kj} + \sum_{j=2}^{p}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} b_{ik}x_{i,j-1}x_{kj} \end{pmatrix} \to lex-\max-\min \\ \sum_{i=1}^{m+p} x_{ij} = 1, \forall\, j \in \{1,\ldots,p\}, \\ \sum_{j=1}^{p} x_{ij} \leqq 1, \forall\, i \in \{1,\ldots,m+p\}, \\ \sum_{j=1}^{p}\sum_{k=1}^{m+p} c_{kj}x_{kj} + \sum_{j=2}^{p}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} d_{ik}x_{i,j-1}x_{kj} \leqq e, \\ x_{ij} \in \{0,1\}, \forall\, i\{1,\ldots,m+p\}, j \in \{1,\ldots,p\}. \end{cases}$$

We show how the problem (PBG) can be modeled as a new special type of the Prize Collecting Traveling Salesman Problem, or as a new special type of the Simple Cycle Problem. We name it *max-min lexicographical Traveling Salesman Problem with objective function depending on a parameter*, and we denote it as *lex max-min BTSPPF*.
To this purpose, let us consider the following complete directed graph $G = (V, E)$, where

$$V = \{0, 1, \ldots, m + p\}$$

and

$$E = \{(i,k) \,|\, i \in V, k \in V, k \neq i\}.$$

Let $\mathcal{C}$ be the set of all cycles with the following properties:

– contain vertex 0,

– apart from vertex 0, they contain exactly p distinct vertices.

Note that each cycle $C \in \mathcal{C}$ may be described by a $p+2$-dimensional vector $(0, u_1, \ldots, u_p, 0)$, which indicates the order in which the vertices follow one other in the cycle. Since this vector always contains 0 on the first and on the last position, the useful information is given by the p-dimensional vector $u = (u_1, \ldots, u_p)$, called *descriptor vector* or *the vector which describes the cycle C*.
For instance, for $m = 5$, $p = 3$, the cycle through the vertices 0, 1, 4, 2 and again 0 is described by the vector $u = (1,4,2)$. Similarly, $u = (1,5,4)$ describes the cycle consisting in the vertices 0, 1, 5 , 4 and returning to 0.

Remark 5.1 *The vector $u = (u_1, \ldots, u_p) \in \mathbb{R}^p$ describes a cycle from $\mathcal{C}$, if and only if the following two conditions are fulfilled:*

i1) $u_j \in \{1, \ldots, m + p\}$, for all $j \in \{1, \ldots, p\}$;

i2) for all $k,h \in \{1, \ldots, p\}$, $k \neq h$, we have $u_k \neq u_h$.

In the following, we use the function $sign : \mathbb{R} \to \{-1, 0, 1\}$,

$$\operatorname{sign} r = \begin{cases} 0, \text{ if } r = 0, \\ 1, \text{ if } r > 0, \\ -1, \text{ if } r < 0. \end{cases} \tag{22}$$

Remark 5.2 *If $u = (u_1, \ldots, u_p)$ is the descriptor vector of a cycle in $\mathcal{C}$, then the following conditions hold:*

i) For every $j \in \{1, \ldots, p\}$, there exists a unique $h_j \in \{1, \ldots, m + p\}$ such that $h_j = u_j$ and, therefore, if $h \in \{1, \ldots, m + p\}$, we have

$$(1 - sign(|u_j - h|)) = \begin{cases} 1, \textit{ if } h = h_j, \\ 0, \textit{ if } h \neq h_j. \end{cases} \tag{23}$$

ii) $\sum_{h=1}^{m+p} (1 - sign(|h - u_j|)) = 1.$

iii) For every $i \in \{1, \ldots, m + p\}$, there is at most an index $j_i \in \{1, \ldots, p\}$ such that $i = u_{j_i}$.

.

Remark 5.3 *If $X = [x_{ij}] \in \mathcal{X}$, then for every $j \in \{1,\dots,p\}$, the following statements hold:*

i) there exists a unique index $h_j \in \{1,\dots,m+p\}$ such that $x_{h_j,j} = 1$ and we have $x_{hj} = 0$, for all $h \in \{1,\dots,m+p\} \setminus \{h_j\}$;

ii) $\sum_{h=1}^{m+p} h x_{hj} = h_j.$

Theorem 5.4 *a) If $X = [x_{ij}] \in \mathcal{X}$, then the vector $u = (u_1,\dots,u_p)$, where*

$$u_j = \sum_{i=1}^{m+p} i x_{ij}, \ \forall\, j \in \{1,\dots,p\}, \tag{24}$$

is a descriptor of a cycle in $\mathcal{C}$.
b) If the vector $u = (u_1,\dots,u_n)$ is a descriptor of a cycle in $\mathcal{C}$, then the matrix $X = [x_{ij}]$, with

$$x_{ij} = 1 - sign(|i - u_j|), \ \forall\, i \in \{1,\dots,m+p\},\ j \in \{1,\dots,p\}, \tag{25}$$

verifies the conditions (15) - (17).

Proof. a) If $X = [x_{ij}] \in \mathcal{X}$, from Remark 5.3, we deduce that, for every $j \in \{1,\dots,p\}$, there is a unique $h_j \in \{1,\dots,m+p\}$ such that

$$\sum_{h=1}^{m+p} h x_{hj} = h_j.$$

Then, in view of (24), we have

$$u_j = h_j, \ \forall\, j \in \{1,\dots,p\}. \tag{26}$$

It follows that $u_j \in \{1,\dots,m+p\}$, for all $j \in \{1,\dots,p\}$.
There may not exist two distinct indices $j', j'' \in \{1,\dots,p\}$ such that $u_{j'} = u_{j''}$. Indeed, if we have $u_{j'} = u_{j''}$, then we get that $h_{j'} = h_{j''}$. Then, as $j' \neq j''$, we have

$$\sum_{h=1}^{m+p} h x_{hj'} + \sum_{h=1}^{m+p} h x_{hj''} = h_{j'} + h_{j''} > 1 + 1.$$

On the other hand, from (16), we obtain

$$\sum_{h=1}^{m+p} h x_{hj'} + \sum_{h=1}^{m+p} h x_{hj''} \leqq 1 + 1,$$

that contradicts the previous inequality.
Hence, the numbers u_j, $j \in \{1,\dots,p\}$, are p distinct elements in set $\{1,\dots,m+p\}$, so the vector u satisfies the conditions i) and ii) from Remark 5.1. Therefore, the vector $u = (u_1,\dots,u_p)$ is the descriptor of a cycle in $\mathcal{C}$.
b) Let us suppose that the vector $u = (u_1,\dots,u_p)$ is a descriptor of a cycle in $\mathcal{C}$. Let us consider $i \in \{1,\dots,m+p\}$, $j \in \{1,\dots,p\}$. Due to the fact that $|i - u_j| \geqq 0$, we have

$$\text{sign}(|i - u_j|) \in \{0, 1\},$$

which implies

$$x_{ij} = 1 - \text{sign}(|i - u_j|) \in \{1, 0\}.$$

Hence (17) holds. Based on Remark 5.2, for every $j \in \{1,\ldots,p\}$ we have

$$\sum_{i=1}^{m+p} x_{ij} = 1,$$

so (15) takes place.
Let $i \in \{1,\ldots,m+p\}$. From Remark 5.2, iii), it results that only the following two cases are possible:

a) There exists $j_i \in \{1,\ldots,p\}$ such that $u_{j_i} = i$ and $u_j \neq i$, for all $j \in \{1,\ldots,p\} \setminus \{j_i\}$; in this case we have

$$\sum_{j=1}^{p} x_{ij} = \sum_{j=1}^{p}(1 - \text{sign}(|i - u_j|)) = \sum_{j=1, j\neq j_i}^{p}(1-1) + (1 - \text{sign}(0)) = 1.$$

b) There is no $j \in \{1,\ldots,p\}$ such that $u_j = i$; in this case

$$\sum_{j=1}^{p} x_{ij} = \sum_{j=1}^{p}(1 - \text{sign}(|i - u_j|)) = \sum_{j=1}^{p}(1-1) = 0.$$

In both cases (17) holds. $\diamond$

Corollary 5.5 *The following statements hold:*

i) If $X \in \tilde{\mathcal{X}}$ is a feasible solution to problem (PBG), then the vector $u = (u_1,\ldots,u_p)$, where u_j is given by (24), for all $j \in \{1,\ldots,p\}$, is the descriptor vector of a cycle in $\mathcal{C}$, cycle which verifies the condition

$$\sum_{j=1}^{p} c_{u_j,j} + \sum_{j=2}^{p} d_{u_{j-1},u_j} \leqq e, \tag{27}$$

equivalent to

$$\sum_{j=1}^{p}\sum_{i=1}^{m+p} c_{ij}(1 - sign(|u_j - i|)) + \tag{28}$$

$$\sum_{j=2}^{p}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} d_{ik}(1 - sign(|u_{j-1} - i|))(1 - sign(|u_j - k|)) \leqq e.$$

ii) If $u = (u_1,\ldots,u_p)$ describes a cycle in $\mathcal{C}$ for which condition (27) is verified, then the matrix $X = [x_{ij}]$, where x_{ij} is given by (25), is a feasible solution to problem (PBG).

Proof. i) Since $X = [x_{ij}]$ is a feasible solution to problem (PBG), the conditions (15)- (17) will be verified. Hence, based on Theorem 5.4, the vector $u = (u_1,\ldots,u_p)$ describes a cycle in $\mathcal{C}$. Applying Remark 5.3 we deduce that for every $j \in \{1,\ldots,p\}$ there is a unique $h_j \in \{1,\ldots,m+p\}$ such that $x_{h_j,j} = 1$ and $x_{ij} = 0$, for all $i \in \{1,\ldots,m+p\} \setminus \{h_j\}$. Then

$$\sum_{i=1}^{m+p} c_{ij}x_{ij} = c_{h_j,j}. \tag{29}$$

Also, for $i,k \in \{1,\ldots,m+p\}$ and $j \in \{2,\ldots,p\}$, we have

$$d_{ik}x_{i,j-1}x_{k,j} = \begin{cases} d_{h_{j-1},h_j}, \text{ if } i = h_{j-1},\ k = h_j \\ 0, \text{ if } i \neq h_{j-1}, \text{or } k \neq h_j. \end{cases} \tag{30}$$

From (29) and (30) we get

$$\sum_{j=1}^{p}\sum_{k=1}^{m+p} c_{kj}x_{kj} + \sum_{j=2}^{p}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} d_{ik}x_{i,j-1}x_{kj} = \sum_{j=1}^{p} c_{h_j,j} + \sum_{j=1}^{p-1} d_{h_{j-1},h_j}.$$

As X is a feasible solution to (PBG), inequality (18) holds. Therefore, from the previous equality, we get

$$\sum_{j=1}^{p} c_{h_j,j} + \sum_{j=1}^{p-1} d_{h_{j-1},h_j} \leqq e. \tag{31}$$

Taking into account condition ii) from Remark 5.3, for vector $u = (u_1,\ldots,u_p)$, where u_j is given by (24), we have

$$u_j = h_j, \text{ for any } j \in \{1,\ldots,p\}, \tag{32}$$

which, based on (31), implies that we have

$$\sum_{j=1}^{p} c_{u_j,j} + \sum_{j=2}^{p} d_{u_{j-1},u_j} \leqq e,$$

so (27) holds.
We will prove that (27) is equivalent to (28).
Let us consider $j \in \{1,\ldots,p\}$. From (32), successively, we get

$$\sum_{i=1}^{m+p} c_{ij}(1 - \text{sign}(|u_j - i|)) = c_{h_j,j}, \tag{33}$$

and

$$\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} d_{ik}(1 - \text{sign}(|u_{j-1} - i|))(1 - \text{sign}(|u_j - k|)) = d_{h_{j-1},h_j}. \tag{34}$$

From (33) and (34), it results that, if (28) holds, then (27) also holds, and vice-versa. So, (28) is equivalent to (27).
ii) Let $u = (u_1,\ldots,u_n)$ be a vector which describes a cycle in $\mathcal{C}$ and verifies the condition (28). Consider also the matrix $X = [x_{ij}]$, with x_{ij} given by (25). From theorem 5.4, we know that X verifies (15)-(17). We still have to prove that the condition (18) holds. Let us consider $j \in \{1,\ldots,p\}$. Using (25), we get:

$$\sum_{i=1}^{m+p} c_{ij}x_{ij} = \sum_{i=1}^{m+p} c_{ij}(1 - \text{sign}(|i - u_j|)) = c_{u_j,j}. \tag{35}$$

Hence

$$\sum_{j=1}^{p}\sum_{i=1}^{m+p} c_{ij}x_{ij} = \sum_{j=1}^{p} c_{u_j,j}. \tag{36}$$

Also,

$$\sum_{j=1}^{p-1}\sum_{i=1}^{m+p}\sum_{k=1}^{m+p} d_{ik}x_{i,j-1}x_{k,j} = \sum_{j=1}^{p-1}\sum_{i=1}^{m+p}\sum_{k=1,k\neq i}^{m+p} d_{ik}(1-\text{sign}(|i-u_j|)(1-\text{sign}(|i-u_{j+1}|) \tag{37}$$

$$= \sum_{j=1}^{p-1} d_{u_j,u_{j+1}}.$$

From (36)-(37), taking into account (27), it results that (18) holds. $\diamond$
Let us denote by $\tilde{\mathcal{C}}$ the set of cycles from $\mathcal{C}$ which verify the condition (28), and let $U(\mathcal{C})$ be the set of vectors which describe the cycle in $\mathcal{C}$. On the set $U(\mathcal{C})$, we define the vector function $\tilde{F} = (\tilde{F}_1, \tilde{F}_2)$, where

$$\tilde{F}_1(u) = \sum_{j=1}^{p} g_{u_j,j}, \forall\, u \in U(\mathcal{C}), \tag{38}$$

and

$$\tilde{F}_2(u) = \sum_{j=1}^{p} a_{u_j,j} + \sum_{j=2}^{p} b_{u_{j-1},u_j}, \forall\, u \in U(\mathcal{C}), \tag{39}$$

respectively.

Remark 5.6 *It is not difficult to prove that:*

i) If $X = [x_{ij}] \in \mathcal{X}$, and $u = (u_1, \ldots, u_p)$ is the vector whose components are defined by (24) for all $j \in \{1, \ldots, p\}$, then

$$F_1(X) = \tilde{F}_1(u); \tag{40}$$

ii) If $u = (u_1, \ldots, u_p) \in U(\mathcal{C})$, then, for the matrix $X = [x_{ij}]$ whose components are defined by (25), we have

$$\tilde{F}_2(u) = F_2(X). \tag{41}$$

We say that a cycle $C^0 \in \tilde{\mathcal{C}}$ is *lexicographical max-min non dominate with respect to the set* $\tilde{\mathcal{C}}$ if there is no cycle $C \in \tilde{\mathcal{C}}$ such that:

$$\tilde{F}_1(C) > \tilde{F}_1(C^0), \text{ or } \tilde{F}_1(C) = \tilde{F}_1(C^0) \text{ and } \tilde{F}_2(C) < \tilde{F}_2(C^0).$$

The problem of finding a lexicographically max-min non dominated cycle with respect to set $\tilde{\mathcal{C}}$ will be denoted as (PG). We describe this problem as:

$$(PG) \quad \begin{cases} \begin{pmatrix} \sum_{j=1}^{p} g_{u_j,j} \\ \sum_{j=1}^{p} a_{u_j,j} + \sum_{j=2}^{p} b_{u_{j-1},u_j} \end{pmatrix} \rightarrow lex - \max - \min \\ \sum_{j=1}^{p} c_{u_j,j} + \sum_{j=2}^{p} d_{u_{j-1},u_j} \leqq e, \\ u = (u_1, \ldots, u_p) \in U(\mathcal{C}). \end{cases} \tag{42}$$

By *optimal solution for* (PG) we understand each lexicographical max-min non dominated cycle with respect to set $\tilde{\mathcal{C}}$.

Corollary 5.7 *The following statements hold:*
i) If X is an optimal solution to problem (PBG), meaning that it is lex-max-min with respect to set $\mathcal{X}$, then the vector $u = (u_1, \ldots, u_p)$, where u_j is given by (24), for all $j \in \{1, \ldots, p\}$, describes a cycle which is lexicographical max-min with respect to set $\tilde{\mathcal{C}}$.
ii) If the vector $u = (u_1, \ldots, u_p)$ describes a cycle which is lexicographical max-min with respect to the set $\tilde{\mathcal{C}}$, then the matrix $X = [x_{ij}]$, where x_{ij} are given by (25), is an optimal solution to problem (PB), which means that it is a lexicographical max-min point according with set $\tilde{X}$.

Proof. Let $X = [x_{ij}]$ be a point that is lexicographical max-min non dominated with respect to the set $\tilde{X}$. Based on Consequence 5.5, the vector $u = (u_1, \ldots, u_n)$ is a feasible solution to problem (PG). Let us suppose that u is not a lexicographically max-min cycle with respect to the set $\tilde{\mathcal{C}}$. Two cases are possible:
Case I. There exists $C^* \in \tilde{\mathcal{C}}$ such that

$$\tilde{F}_1(u) < \tilde{F}_1(v), \tag{43}$$

where v denotes the vector which describes the circuit C^*. Based on Consequence 5.5, the matrix $Y = [y_{ij}]$, where y_{ij} are given by

$$y_{ij} = 1 - \text{sign}(|i - v_j|), \forall i, j \in \{1, \ldots, n\} \tag{44}$$

is a feasible solution to problem (PBG). Based on Remark 5.6, we have $F_1(Y) = \tilde{F}_1(v)$ and $F_2(Y) = \tilde{F}_2(v)$. Now, by taking into account (43), we get $F_1(Y) > F_1(X)$, which contradicts the hypothesis that X is an optimal solution to problem (PBG), which means that it is lexicographical max-min non dominated with respect to the set of feasible solutions to the problem. Since this is a contradiction, we deduce that there cannot exist a cycle $C^* \in \mathcal{C}$ such that (43) holds.
Case II. There exists $C^* \in \tilde{\mathcal{C}}$ such that

$$\tilde{F}_1(u) = \tilde{F}_1(v), \text{ but } \tilde{F}_2(u) > \tilde{F}_2(v), \tag{45}$$

where v is the descriptor vector of the cycle C^*. As in the previous case, we consider the matrix $Y = [y_{ij}]$ whose components are given by (44), matrix which is a feasible solution to problem (PBG). Based on Remark 5.6, we get that $F_1(X) = \tilde{F}_1(u)$, $\tilde{F}_2(u) = F_2(X)$ and $F_1(Y) = \tilde{F}_1(v)$, $\tilde{F}_2(v) = F_2(Y)$. Therefore, by taking into account (45), the following relations hold:

$$F_1(X) = F_1(Y) \text{ and } F_2(X) > F_2(Y),$$

which contradicts the hypothesis that X is a lexicographical max-min point with respect to the set of feasible solutions to problem (PBG). Similarly, ii) may be proved. $\diamond$
From Theorem 5.4 and Consequences 5.5, 5.7, it results that solving the problem (PG) may be reduced to solving problem (PBG) and vice-versa. Therefore, the Registrar Problem consists in finding a cycle, C^0, which is lexicographical max-min non dominated with respect to the set $\tilde{\mathcal{C}}$.

6. The problem (PG) from general point of view

The (PG) problem can be seen, in terms of the Traveling Salesman Problem, as follows: a traveling salesman must take a tour through exactly p towns, at his choice among the $m + p$ given towns, and stays exactly one day in each town. The costs b_{ik} of going from town i to

town k are also given, for all towns i and k, $i \neq k$. For each town i that is visited, the salesman gets a bonus g_{ij} that depends not only on the location i, but also on the day j he gets to that location. Also, for each visited location i, in days j, he pays a_{ij} for accommodation. The problem is that of finding a cycle so that the total bonus is maximized and the total costs (transport costs plus accommodation costs) are minimized. A boundary condition (18) must also be satisfied.

For this reason, the problem (PG) can be seen as a generalization of the following problems: Prize Collecting Traveling Salesman Problem, Simple Cycle Problem, Capacitated Prize-Collecting Traveling Salesman Problem or Orienteering Problem. Note that all these variants of the Traveling Salesman Problem allow that the cycle not to visit all the vertices of the graph.

The *Prize Collecting Traveling Salesman Problem* was posed in 1986 by E. Balas. A synthesis of the results on this topic and of the solving methods for this problem can be found in (Balas, 2002). In accordance with this paper, in the Prize Collecting Traveling Salesman Problem: *a salesman gets a prize w_k in every city k that he visits and pays a penalty c_l to every city l that he fails to visit. Traveling at cost c_{ij} between cities i and j our salesman wants to find a tour that minimizes his travel costs and penalties, subject to a lower bound w_0 on the amount of prize money he collects*.

In *Simple Cycle Problem* (see (Fischetti et al, 2002)), a complete undirected graph $G = (V, E)$ is given. A cost c_e is associated to each edge $e \in E$ and a prize p_n is associated to each node $n \in V$. The cost of a cycle C is given by the difference between the sum of all costs corresponding to the edges of this cycle and the sum of all prizes corresponding to the nodes of this cycle. The problem is to find a min-cost cycle of at least 3 vertices. A further requirement may be that a given vertex be part of the cycle.

Capacitated Prize-Collecting Traveling Salesman Problem is derived from simple cycle problem. For this problem, to each node it is assigned a *weight* and there is a further restriction that the sum of weights of visited vertices not to exceed a given maximum value.

In the case of the *Orienteering Problem* or *Selective Traveling Salesman Problem*, the transport costs assigned to all edges are zero, that is, $c_e = 0$, for all $e \in E$. In exchange, to each edge $e \in E$ it is assigned a positive value, a duration t_e, and it is required that the sum of durations for all visited edges not to exceed a given value.

All these problems can be seen as some variants or generalization of the Traveling Salesman Problem. Practically (see (Balas, 2002)) each situation involving decisions that affect the sequence in which various actions, tasks or operations are to be executed, has a traveling salesman problem aspect to it. We remember that in accordance with (Punnen, 2002) the *Traveling Salesman Problem* (TSP) is to find a routing of a salesman who starts from a home location, visits a prescribed set of cities and returns to the original location in such a way that the total distance traveled is minimum and each city is visited exactly once.

The differences between the above four TSP variants and the (PG) problem are:

– the fact that the exact number of cycle vertices is fixed,

– the bonus depends not only on the visited vertex, but also on the visiting time, that is, on the position of the vertex along the cycle;

– the goal is expressed by a vector, with two components (bonus and cost).

The authors of this work do not have knowledge of a paper discussing a TSP variant, where the bonus depends on the visiting time or where the exact number of vertices to be used is given (except, obviously, the case where all vertices are to be visited). These are new original topics, discussed in this work.

The subject of multi-criteria Traveling Salesman Problem, in a general context, is treated, for example, by (Ehrgott, 2005). To transfer the concept of optimal solutions to multi-criteria problems, the notion of Pareto curves was introduced
A Pareto curve is the set of Pareto points or, equivalent, efficient points.
Let $D \subseteq \mathbb{R}^n$ be a nonempty set, $f = (f_1, ..., f_p) : D \to \mathbb{R}^p$ a vectorial function, and $S \subseteq D$. A point $x \in S$ is said to be a max-efficient point of f with respect to S or a max-Pareto point of f with respect to S if there is no $y \in S$ such that

$$f_i(x) \leqq f_i(y), \forall i \in \{1, ..., p\} \text{ and } \sum_{i=1}^{p} f_i(x) < \sum_{i=1}^{p} f_i(y),$$

or, equivalent,

$$f_i(x) \leqq f_i(y), \forall i \in \{1, ..., p\} \text{ and there is } h \in \{1, ..., p\} \text{ such that } f_h(x) < f_h(y).$$

Unfortunately, Pareto curves cannot be computed efficiently in many cases because they are often of exponential size and NP-hard to compute even for otherwise easy optimization problems. Therefore, sometimes, one prefers to choose a point on this curve, point subject to some additional restriction which derives from the scalar components of the scope function. This point is, often, a maximum point of the weighted sum of these scalar components. In other cases, it is chosen to be a non-dominant point into a lexicographical ordering relation. This is exactly the situation presented in this paper.
There exist, also, papers which present possibilities to approximate the Pareto curve (see, for example (Warburton, 1987), (Angel et al., 2004) or (Manthey, 2009)).
The globally convex structure of Pareto curves in studied for example in (Borges & Hansen, 2001) and (Villagra et al., 2006).
We mention that, in (Feillet et al., 2005), the authors show that if, in the TSP, we consider that we have two objective functions, the profit and the cost, we obtain a bi-criteria TSP. According to the results of multi-criteria optimization, for solving bi-criteria TSP three cases are considered by the authors:
- Both objectives are combined in the objective function and the aim is to find a circuit that minimizes travel costs minus collected profit, i.e. the bi-criteria Traveling Salesman Problem may be seen as a prize collecting traveling salesman problem;
- The travel costs objective is stated as a constraint and the aim is to find a circuit that maximizes collected profit such that travel costs do not exceed a preset value, i.e. the bi-criteria Traveling Salesman Problem may be seen as a orienteering problem;
- The profit objective is stated as a constraint and the aim is to find a circuit that minimizes travel costs and whose collected profit is not smaller than a preset value, i.e. the bi-criteria Traveling Salesman Problem may be seen as a capacitated prize-collecting Traveling Salesman Problem.
However, the above transformations cannot be applied to our problem because the profit and the costs depending, in addition, on the time. Based on these considerations we choose to work with the lexicographic ordering. We remind that the idea to use the lexicographic ordering in a Traveling Salesman Problem was used in (Lupsa et al, 2008), where an application in health economics is also given.

7. Algorithms for solving problem (PG)

It is known that, for solving a route choice problem, techniques of branch and bound type are often used. Based on this technique, in the following, three algorithms are given for solving problem (PG): a greedy algorithm, a parallel approach algorithm and an exact algorithm.

7.1 A greedy algorithm

The algorithm 0.1 constructs a cycle, in a graph, with $p+1$ distinct vertices (vertex 0 plus other p) which satisfies the boundary condition (18). The algorithms signals the case where it is impossible to construct such a cycle. For its construction, one always goes to the branch where the growth of the first component, $\tilde{F}_1$, of the goal function, is the highest. If there are several possibilities of getting the same growth, then it favors that whose second component, $\tilde{F}_2$, is the lowest. In the algorithm, the p-dimensional vector u is used. If $\tilde{\mathcal{C}} \neq \emptyset$, it becomes the descriptor vector of the optimal solution of problem (PG). Also, the following sets are used:
- V_j, that contains, at every iteration j, the candidate vertices to generate the cycle;
- V_j^*, that, at every iteration j, contains those vertices from V_j, whose corresponding coefficients from component $\tilde{F}_1$ of the scope function are maximum;
If the set V_j^* has more than one element, that vertex in the circuit will be chosen, for which we obtain a minimal increase in the value of $\tilde{F}_2$, the second component of the scope function. This can be fulfilled by using the real number r. The cost of the current circuit is stored in r_j. If when building a circuit we determine that the restriction on costs is not satisfied, the last added arc is abandoned, the output vertex being temporarily removed from the set of candidates.

Algorithm 0.1 The Greedy Algorithm.
input: the natural numbers m and p;
the elements of matrices: $G = [g_{ij}]$, $i \in \{0,1,\ldots,m+p\}$, $j \in \{1,\ldots,p\}$;
$A = [a_{ij}]$, $i \in \{0,1,\ldots,m+p\}$, $j \in \{1,\ldots,p\}$;
$B = [b_{ik}]$, $i \in \{0,1,\ldots,m+p\}$, $k \in \{0,1,\ldots,m+p\}$;
$C = [c_{ij}]$, $i \in \{0,1,\ldots,m+p\}$, $j \in \{1,\ldots,p\}$;
$D = [d_{ik}]$, $i \in \{0,1,\ldots,m+p\}$, $k \in \{0,1,\ldots,m+p\}$;
output: *ok* — *true* if a solution exists
u — the solution
algorithm:
$u_k := 0, \forall k \in \{0,1,\ldots,p\}$;
$r_0 := 0$;
finish := *false*;
$j := 1$;
while not *finish* do
$V_j := \left\{ x \in \{1,\ldots,m+p\} \setminus \{u_1,\ldots,u_{j-1}\} \,\middle|\, r_{j-1} + c_{x,j} + d_{u_{j-1},x} \leqq e \right\}$;
if $V_j \neq \emptyset$ then
$g^* := \max\{g_{x,j} \,|\, x \in V_j\}$;
$V_j^* := \{x \in V_j \,|\, g_{x,j} = g^*\}$;
$r^* := \min\{a_{x,j} + b_{u_{j-1},x} \,|\, x \in V_j^*\}$;
$R^* := \{x \in V_j^* \,|\, a_{x,j} + b_{u_{j-1},x} = r^*\}$;
choose $x \in R^*$;

```
                u_j := x;
                r_j := r_{j-1} + c_{x,j} + d_{u_{j-1},x};
                j := j + 1;
                if j > p then
                    ok := true
                    finish := true
                end if
            else
                if j = 1 then
                    ok := false
                    finish := true
                end if
                V_j := V_j \ {u_{j-1}};
                j := j - 1;
            end if
        end while
end algorithm
```

It is possible to make a small change of this algorithm by introducing that vertex which minimizes the ratio between the growth of function $\tilde{F}_1$ and the growth of function $\tilde{F}_2$ relatively to the set of vertex that do not close the cycle; we mention, also, that the introduction of this vertex does not exceed the cost limit.

7.2 Parallel approach for the registrar route problem

The algorithm previously presented is a serial one, which means that it is perform with one processing unit, and depends on the number of hospitals that the registrar has to browse. A more general situation is when the graph is not complete, it means that the registrar may not move to each hospital, but only to some of them, let us say n. If this number is big enough, the serial execution will take a lot of time. In order to speed up and get faster the result, a parallel approach is more convenient. As in (Feilmeier, 1982) applications for high performance computing are given. If we use more than one processing unit, we may obtain the desired result approximately n times faster. We know that the registrar starts from node 0 and has to browse n different locations, being constrained by the restrictions of possible total cost.

The parallel algorithm is of "master-slave" type and has the following steps:

1. According to the graph of problem (PG), the algorithm generates the adjacent matrix, denoted by $E = [e_{ik}]$, where $e_{ik} = 1$,if in graph G there is an edge from i to k, and $e_{ik} = 0$ if not, for every $i,k \in \{0,\ldots,m+p\}$.
2. In the system work as many processors, as many figures 1 appear on the first row of matrix A. For instance, if five figures 1 appear on the first row, then five processors will work. Obviously, the maximum number of processors is n. Every processor will store a copy of the matrix E.
3. According with the adjacent matrix, every processor builds its own circuit and verifies the restrictions. If they are fulfilled, the processor memorizes the circuit and sends a message to the processor master. If one processor finds that there exist more than one admissible circuit, it gives a signal to the master processor, which will allocate another processor in charge with the new circuit.

4. Finally, the processor master will decide which circuit is optimal for problem (PG).

Generally, the parallel machine on which the following algorithm may be implemented is of SIMD type (Single Instruction stream, Multiple Data stream). Due to this fact, no communication among Slave processors is needed. They communicate only with the processor Master. Any type of network may be defined on the parallel machine (for the parallel calculus see, for example (Chiorean, 2000)). When the parallel network is defined by the program, every processor gets an identification number. The Processor Master receives 0, and all the other processors in the system receive numbers starting with 1.

Algorithm 0.2 ParPg
algorithm:
Define _Network {determine the configuration of processors, get their Id's}
if $id = 0$ then
Generate_Adjacent_Matrix(G,E);{input graph, output matrix E}
Determine_Number_of_Slaves (E, n);{n, equal to values of 1 that there are on the first row}
Send_Message_to_Slaves(E){send matrix E in every processor Slave}
else
for $i = 1$ to n in parallel do
Get_Message_from_Master(E);
{copy matrix E from Master};
Verify_Number_of_Cycles (E, nr);
if $nr \geqq 1$ then
for $j = 1$ to nr do
Verify_Restrictions_on_the_Cycle(E, j, $finalvalues$);
{every processor will work on the "nr" of cycles};
{in "$finalvalue$" memorizes the values of variables for the objective function};
Memorize_Cycle ($finalvalues$,j,u);
{the corresponding circuit is kept in vector u};
Send_Message_to_Master(j,u);
end for
end if
end for
end if
if $id = 0$ then
Get_Message_from_Slaves(u){Master collects all the possible cycles}
end if
Determine_Optimum_cycles($final_u$){Compares cycles and determines the optimum one, denoted "$final_u$"}
Print($final_u$)
end algorithm

Working this way, with more than one processor, the result is obtained n times faster, where n is the number of processors in the system.

7.3 An exact algorithm of the Little type

For solving the problem (PG), we built an exact but not polynomial algorithm, based on the branch and bound method as in (Little et al, 1963). For its construction, one always goes to the branch where the growth of the first component, $\tilde{F}_1$, of the goal function, is the highest. If there are several possibilities of getting the same growth, then it favors that whose second component, $\tilde{F}_2$, is the lowest. The bound is made by using two real improper numbers, bf and bc. These numbers are defined iteratively. In the beginning, $bf := 0$ and $bc := +\infty$. If we have at least a cycle, in the graph attached to our problem, then the value of the pair (bf, bc) is equal to the lexicographical max-min of the set of all such pair attached to the cycles which we have been obtain until this moment.
A finite sequence $(k_h, I_h, J_h, G^h, f_h, \gamma_h, \delta_h, w^h)_{h=1}^{N}$, where $N \leqq (m+p)\dots(p+1)$, is built.

- k_h is a binary variable: it has the value 1 if the term h was not studied yet and the value 0 otherwise.
- G^h, is the work matrix in the h-th iteration.
- $u^h \in IR^p$ will be used for building the descriptor of a cycle.
- the numbers f_h and γ_h are equal to $\tilde{F}_1(u^h)$, $\tilde{F}_2(u^h)$, respectively, and there are computed iteratively.
- δ_h is equal to the value of right hand side term of bounded equation corresponding to u^h; it is also iteratively computed.
- The vector w^k is used to inhibit branching; its components are iteratively computed.

Algorithm 0.3 An Exact Algorithm
input: the natural numbers m and p;
the elements of matrices: $G = [g_{ij}]$, $i \in \{0,1,\dots,m+p\}$, $j \in \{1,\dots,p\}$;
$A = [a_{ij}]$, $i \in \{0,1,\dots,m+p\}$, $j \in \{1,\dots,p\}$;
$B = [b_{ik}]$, $i \in \{0,1,\dots,m+p\}$, $k \in \{0,1,\dots,m+p\}$;
$C = [c_{ij}]$, $i \in \{0,1,\dots,m+p\}$, $j \in \{1,\dots,p\}$;
$D = [d_{ik}]$, $i \in \{0,1,\dots,m+p\}$, $k \in \{0,1,\dots,m+p\}$;
output:
algorithm:
$I_1 := \{0,1,\dots,m+p\}$;
$J_1 := \{1,\dots,p\}$;
$t_1 := 1$;
$n_1 := 0$;
$u^1 := (u_0, u_1, \dots, u_n) = (0,\dots,0)$;
$u^* := (0,\dots,0) \in \mathbb{R}^{p+1}$;
$f_1 := 0$;
$\gamma_1 := 0$;
$\delta_1 := 0$;
$bf := 0$;
$bc := +\infty$;
$\forall i \in I_1$, $\forall j \in J_1$, $g_{ij}^1 := g_{ij}$;
$\forall j \in J_1$, $w_j^1 := \max\{g_{ij}^1 \,|\, i \in I_1\}$;
$h := 1$;
$K := \{k \in \{1,\dots,h\} \,|\, t_k = 1\}$;

while $K \neq \emptyset$ do
$s := (\max\max\min)_{\text{lex}} \{(n_k, f_k, \gamma_k) \,|\, k \in K\}$;
choose $k \in K$ such that $(n_k, f_k, \gamma_k) = s$;
$j := n_k + 1$;
$I^* := \{i \in I_k \,|\, \delta_k + c_{ij} + d_{u^k_{j-1},i} > e\}$;
if $I_k \setminus I^* = \emptyset$ then
$t_k := 0$;
else
$\forall i \in I^*$, $g^k_{ij} := -\infty$;
$w^k_j := \max\{g_{ij} \,|\, i \in I_k \setminus I^*\}$;
if $f_k + \sum_{\alpha \in J_k} w^k_\alpha < bf$ then
$t_k := 0$;
else
$M := \{i \in I_k \setminus I^* \,|\, g_{ij} = w^k_j\}$;
$v := \max\{a_{ij} + b_{u_{j-1},i} \,|\, i \in M\}$;
$V := \{i \in M \,|\, a_{ij} + b_{u_{j-1},i} = v\}$;
choose $u^k_j \in V$;
if $(I_k \setminus (I^* \cup \{u^k_j\}) \neq \emptyset)$ and

$$f_k + \sum_{\alpha=1 \alpha>j}^{p} w^k_\alpha + \max\{g_{ij} \,|\, i \in I_k \setminus (I^* \cup \{u^k_j\})\} \geqq bf$$

then
$h := h + 1$;
$t_h := 1$;
$n_h := n_k$;
$u^h := u^k$;
$u^h_0 := 0$;
$f_h := f_k$;
$\gamma_h := \gamma_k$;
$\delta_h := \delta_k$;
$I_h := I_k \setminus I^*$;
$J_h := J_k$;
$\forall i \in I, \forall j \in J$, $g^h_{rs} := g^k_{rs}$;
$g^h_{u_j,j} := -\infty$;
$\forall r \in J \setminus \{j\}$, $w^h_r := w^k_r$;
$w^h_j := \max\{g^h_{ij} \,|\, i \in I_k\}$;
end if
$f_k := f_k + g^k_{u^k_j,j}$;
$\gamma_k := \gamma_k + a_{u^k_j,j} + b_{u^k_{j-1},u^k_j}$;
$\delta_k := \delta_k + c_{u^k_j,j} + d_{u^k_{j-1},u^k_j}$;
if $j = p$ then
if $f_k \geqq bf$ then
$bf := f_k$;
$bc := \gamma_k$;

$u^* := (u_1^k, \ldots, u_p^k)$;
end if
else
$h := h + 1$;
$t_h := 1$;
$n_h := n_k + 1$;
$u^h := u^k + u_j^k e^j$;
$f_h := f_k$;
$\gamma_h := \gamma_k$;
$\delta_h := \delta_k$;
$I_h := I_k \setminus \{u_j^k\}$;
$J_h := J_k \setminus \{j\}$;
$\forall i \in I$, $\forall j \in J$, $g_{rs}^h := g_{rs}^k$;
$\forall r \in I$, $g_{rj}^h := -\infty$;
$\forall s \in J$, $g_{u_j,s}^h := -\infty$;
$\forall r \in J \setminus \{j\}$, $w_r^h := w_r^k$;
$w_j^h := g_{u_j^k,j}^h$;
$t_k := 0$;
end if
end if
end if
$K := \{k \in \{1, \ldots, h\} | t_k = 1\}$;
end while
if bf=0 then
output: the problem (PBG) is inconsistent;
else
output: u^* is an optimal solution of (PBG);
end if
end algorithm

The algorithm 0.3 is very easy to implement and it has been used on a number of test cases for determining the optimal route of the registrar. The output result is optimal.

8. References

Angel, E.; Bampis, E. & Gourvès, L. (2004). Approximating the Pareto curve with local search for the bi-criteria TSP(1,2) problem. *Theoretical Computer Science*, Vol. 310, no. 1-3, 135-146, ISSN 0304-3975

Balas, E. (2002). The Prize Collecting Traveling Salesman Problem and its Application, In: *T*raveling Salesman Problem and its Variations, Gutin, G. & Punnen, A. (Ed.), 663-695, Kluwer Academic Publishers, ISBN 1-4020-0664-0, Dordrecht

Borges, P.C. & Hansen, P.H. (2001). A study of global convexity for a multiple objective travelling salesman problem. In Ribeiro, C.C. & Hansen, P. (Ed.), *Essays and Surveys in Metaheuristics*, 129-150, Kluwer Academic Publishers, ISBN 978-0-7923-7520-3

Briggs, A. & Sculpher, M. (1998). An Introduction to Markov Modelling for Economic Evaluation. *Pharmacoeconomics*, Vol. 13, no. 4, 397-409, ISSN 1170-7690

Briggs, A.; Claxton, K. & Sculpher, M. (2006). *Decision Modelling for Health Economic Evaluation*.

Oxford University Press, ISBN 978-0-19-852662-9, Oxford
Canfelt, K.; Barnabas, R.; Patnik, J. & Bera, V. (2004). The predicted effect of changes in cervical screening practice in the U.K: results from a modelling study. *British Journal of Cancer*, Vol. 91, 530-536, 0007-0920
Chiorean, I. (2000). *Parallel Calculus* (in roumanian), Ed.Albastra, ISBN 973-9215-08-4, Cluj-Napoca
Chiorean, I.; Lupsa, L. & Neamtiu, L. (2008). Markov Models for the Simulation of Cancer Screening Process, In *International Conference on Numerical Analysis and Applied Mathematics proceedings*, 143-147, ISBN 978-0-7354-0576-9, Kos, Greece, September 2008, American Institute of Physics
Crama, Y. (1989) Recognition problems for special classes of polynomials in 0-1 variables. *Mathematical Programming*, Vol. 44, 139-155, ISSN 0025-5610
Ehrgott, M. (2005). *M*ulticriteria Optimization, Second ed., Springer, ISBN 3-540-21398-8, Berlin Heidelberg New York
Eveden, D.; Harper, P.R.; Brailsford, S.C. & Harindra, V. (2005). System Dynamics modelling of Chlamydia infection for screening intervention planning cost-benefit estimation. *IMA Journal of Management Mathematics*, Vol. 16, no. 3, 265-279, ISSN 1471-678X
Fayers, P. M.; Ashby, D. & Parmer, M. K. B. (1997). Bayesian data monitoring in clinical trials, *Statistics in Medicine*, Vol. 16, 1413-1430, ISSN 0277-6715
Feillet, D.; Dejax, P. & Gendreau, M. (2005). Traveling Salesman Problems with Profits, *T*ransportation Science, Vol. 39, May 2005, 188-205, ISSN 0041-1655
Feilmeier, M. (1982). Parallel numerical algorithms, In *Parallel Processing Systems*, Evans, D. (ed.), 285-338, Cambridge University Press, ISBN 0-521-24366-1, Cambridge
Fischetti, M.; Salazar-González, J.-J. & Toth, P. (2002). The Generalized Traveling Salesman and Orienteering Problems, In: *T*raveling Salesman Problem and its Variations, Gutin, G. & Punnen, A. (Ed.), 609-662, Kluwer Academic Publishers, ISBN 1-4020-0664-0, Dordrecht
Grosan, C.; Abraham, A.; Campian, R. & Ţigan, Şt. (2005). Evolution Strategies for Ranking Several Trigeminal Neuralgia Treatments, *Applied Medical Informatics*, Vol. 17, no. 3-4, 72-78, ISSN 103-443-015
Grosan, C.; Abraham, A. & Ţigan, Şt. (2007). Multicriteria programming in medical diagnosis and treatments, *Applied Soft Computing*, Vol. 8, no. 4 (September), 1407-1417, ISSN 1568-4946
Hammer (Ivănescu), P.L. & Rudeanu, S. (1968). *B*oolean Methods in Operations Research and Related Areas, Springer, ISBN 0387042911, Berlin
Kang, M. & Lagakos, St. W. (2007). Statistical methods for panel data from a semi-Markov process, with application to HPV, *Biostatistics*, Vol. 8, no. 2, 252-264, ISSN 1465-4644
Little, J.D.C.; Murty, K.G.; Sweeney & D.W.; Karel, K. (1963). An algorithm for the traveling salesman problem. *Operation Research*, Vol. 11, No. 6, November-December 1963, 972-989, ISSN 0160-5682
Lupşa, L. (1999) A Criterion for Caracterizing a Medical Treatment that Uses Multicriteria Programming in Pharmacoeconomics. In *Analysis, Functional Equations, Approximation and Convexity*. Editura Carpatica, 142-146, ISBN 979-97664-9-8, Cluj-Napoca
Lupşa, L. (2000) Multicriteria Programming Used in Medico-Economic Analysis of Treatment Protocols. In *Proceedings of the "Tiberiu Popoviciu" Itinerant Seminar of Functional Equations, Approximation and Convexity*. Editura SRIMA, 103-111, ISBN 973-98591-9-4,

Cluj-Napoca

Lupşa, L. (2000). The sort of a class of drugs using the balanced points methods. In *Séminaire de la Théorie de la Meilleure Approximation, Convexité et Optimisation*, Editura SRIMA 2000, 171-181, ISBN 973-99781-0-X, Cluj-Napoca

Lupşa, R.; Lupşa, L. & Neamtiu, L. (2008):.Optimal Model to Solve the Transport Model for Mammography Screening, 2008 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008) THETA 16th edition proceedings, tome 3, pp. 139-142, ISBN 978-1-4244-2576-1, Cluj-Napoca

Manthey, B. (2009). On Approximating Multi-Criteria TSP, In *Symposium on Theoretical Aspects of Computer Science*, 637648, Freiburg, 2009, www.stacs-conf.org

Neamţiu, L. (2008). *Modeling and Decision in Health Economics* (in roumanian), Casa cărţii de ştiinţa, ISBN 978-973-133-380-9, Cluj-Napoca

Neanţiu, L. (2009). A Medical Resources Allocation Problem, *Results in Mathematics*, Vol. 53, nr. 3-4 (July), 341-348, ISSN 1422-6383

Punnen, A. P. (2002). The Traveling Salesman Problem: Applications, Formulation and Variations, In: *T*raveling Salesman Problem and its Variations, Gutin, G. & Punnen, (Ed.), 1-28, Kluwer Academic Publishers, ISBN 1-4020-0664-0, Dordrecht

Trobec, R.; Vajtersic, M. & Zinterhof, P. (eds) (2009). *Parallel Computing. Numerics, Applications, and Trends.* Springer, ISBN 978-1-84882-408-9, Dordrecht Heidelberg London New York

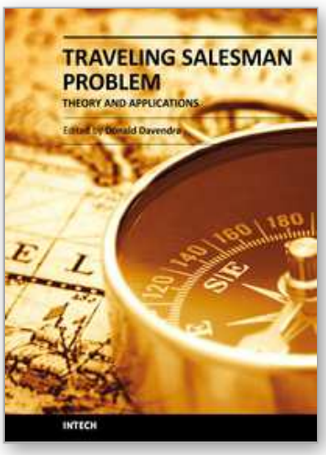
Ţigan, Şt.; Achimaş, A.; Coman, I.; Drugan, T. & Iacob, E. (2001). *Multifactorial Decisions* (in roumanian) . Editura SRIMA, ISBN 973-99781-3-4, Cluj-Napoca

Villagra, M.; Barán, B. & Gómez, O. (2006). Global Convexity in the Bi-Criteria Traveling Salesman Problem. In *Artificial Intelligence in Theory and Practice*, 217-226, Springer, ISBN 978-0-387-34654-0, Boston

Warburton, A. (1987). Approximation of Pareto optima in multiple-objective shortest path problems. *Operations Research*, Vol. 35, no. 1 (Jan-Feb.), 7079, ISSN 0030-364X

Willan, A. R. & Briggs, A. H. (2006). *Statistical Analysis of Cost-effectiveness Data*. John Wiley & Sons Ltd, ISBN 978-0-470-85626-0, Chichester

World Health Organization (1999). *C*ancer Registration: Principles and Methods, Jensen, O.M.; Parkon, D.M.; MacLennan, R.; Muir, C.S.; Skeet, R.G. eds., IARC Scientific Publications no. 95, ISBN 92-832-1195-2, Lyon



Traveling Salesman Problem, Theory and Applications
Edited by Prof. Donald Davendra

ISBN 978-953-307-426-9
Hard cover, 298 pages
Publisher InTech
Published online 30, November, 2010
Published in print edition November, 2010

This book is a collection of current research in the application of evolutionary algorithms and other optimal algorithms to solving the TSP problem. It brings together researchers with applications in Artificial Immune Systems, Genetic Algorithms, Neural Networks and Differential Evolution Algorithm. Hybrid systems, like Fuzzy Maps, Chaotic Maps and Parallelized TSP are also presented. Most importantly, this book presents both theoretical as well as practical applications of TSP, which will be a vital tool for researchers and graduate entry students in the field of applied Mathematics, Computing Science and Engineering.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Liana Lupsa, Ioana Chiorean, Radu Lupsa and Luciana Neamtiu (2010). Some Special Traveling Salesman Problems with Applications in Health Economics, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9, InTech, Available from: http://www.intechopen.com/books/traveling-salesman-problem-theory-and-applications/some-special-traveling-salesman-problems-with-applications-in-health-economics

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen