Zeszyty Naukowe
Politechniki Białostockiej
INFORMATYKA
Numer 5

Oficyna Wydawnicza Politechniki Białostockiej

Białystok 2010

Artykuły zamieszczone w *Zeszytach Naukowych Politechniki Białostockiej. Informatyka*
otrzymały pozytywne opinie recenzentów wyznaczonych przez Redaktora Naczelnego i Radę Naukową

The articles published in *Zeszyty Naukowe Politechniki Białostockiej. Informatyka*
have been given a favourable opinion by reviewers designated by Editor-In-Chief and Scientific Board

ISSN 1644-0331

ADRES DO KORESPONDENCJI/THE ADRESS FOR THE CORRESPONDENCE:

„*Zeszyty Naukowe Politechniki Białostockiej. Informatyka*"
Wydział Informatyki /Faculty of Computer Science
Politechnika Białostocka /Bialystok University of Technology
ul. Wiejska 45a, 15-351 Białystok
tel. 85 746 90 50, fax 85 746 97 22
e-mail: znpb@irys.wi.pb.edu.pl
http://irys.wi.pb.edu.pl/znpb

Druk: Oficyna Wydawnicza Politechniki Białostockiej
Nakład: 100 egzemplarzy

CONTENTS

AN IMPROVED APPROXIMATION ALGORITHM FOR OPTIMAL ROUTES GENERATION IN PUBLIC TRANSPORT NETWORK

Jolanta Koszelew[1]

[1]Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: This paper presents a new version of Routes Generation Matrix Algorithm, called Routes Generation Matrix Improved Algorithm (RGMIA), for determining routes with optimal travel time in public transport network. The method was implemented and tested on the real public transport network in Warsaw city. This network was completed with walk links and therefore resultant routes are more practical and can perform various users' preferences. Effectiveness of the improved method was compared in two aspects: time complexity and quality of results, with another two algorithms - previous version of Routes Generation Matrix Algorithm (RGMA) and Routes Generation Genetic Algorithm (RGGA). RGMA and RGGA algorithms were described in previous author's papers [9,10].

Keywords: public transport network, time-dependent shortest path, optimal routes, genetic algorithm

1. Introduction

The shortest path problem is a core model that lies at the heart of network optimization. It assumes that weight link in traditional network is static, but is not true in many fields such as Intelligent Transportation Systems (ITS) [16]. The optimal path problems in variable-time network break through the limit of traditional shortest path problems and become foundation theory in ITS. The new real problems make the optimal path computing to be more difficult than finding the shortest paths in networks with static and deterministic links, meanwhile algorithms for a scheduled transportation network are time-dependent.

A public transportation route planner is a kind of ITS and provide information about available public transport journeys. Users of such system determine source and destination point of the travel, the start time, their preferences and system returns

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 5-17, 2010.

as a result, information about optimal routes. In practice, public transport users' preferences may be various, but the most important of them are: a minimal travel time and a minimal number of changes (from one vehicle to another). Finding routes with minimal number of changes is not a difficult problem, but generating routes with minimal time of realization, on the base of dynamic timetables, is much more complexity task.

Moreover, standard algorithms considered graphs with one kind of links (undirected or directed) which have no parallel arcs. Graph which models a public transport network includes two kinds of edges: directed links which represent connections between transport stops and undirected arcs correspond to walk link between transport stops.

Additionally, with each node in a graph which represents a transportation network, is concerned detail information about: timetables, coordinates of transport stops, etc. This information is necessarily to determine weights of links during realization of the algorithm.

Besides, standard shortest path algorithms generate only one optimal path, but methods used in journey planners must return few alternative optimal paths. These four differences between standard shortest path problem and routing problem in public transportation network cause that time complexity of algorithms which solve this problem may be very high.

Many algorithms has been developed for networks whose edge weights are not static but change with time but most of them take into consideration a network with only one kind of link, without parallel links and returns only one route. Cooke and Halsey [5] modified Bellman's [3] "single-source with possibly negative weights" algorithm to find the shortest path between any two vertices in a time-dependent network. Dreyfus [6] made a modification to the standard Dijkstra algorithm to cope with the time-dependent shortest path problem. Orda and Rom [13] discussed how to convert the cost of discrete and continuous time networks into a simpler model and still used traditional shortest path algorithms for the time-dependent networks. Chabini [4] presented an algorithm for the problem that time is discrete and edge weights are time-dependent. Other algorithms deal with finding the minimum cost path in the continuous time model. Sung [14] et al. gave a similar result using a different version of cost and improved the algorithm's efficiency. Ahuja [1] proved that finding the general minimum cost path in a time-dependent network is NP-complete and special approximation method must be used to solve this problem.

The RGMA [9] and RGGA [10] are approximation methods which generates k routes with optimal travel time. Like the k-shortest paths algorithm [11], since these methods generate multiple "better" paths, the user can have more choices from where

he or she can select based on different preferences such as total amount of fares, convenience, preferred routes and so on.

RGMA realizes label-setting strategy [2] for construct optimal routes and uses special matrices which are applied as heuristics in this algorithm. RGGA algorithm is a genetic algorithm [12] which starts with a population of randomly generated initial set of routes and tries to improve individuals in population by repetitive application of selection and crossover operators. Both algorithm was implemented and tested on realistic data - from Bialystok city public transport network [10]. Computer experiments had shown that genetic algorithm (RGGA) generates routes as good as matrix based algorithms (RGMA), but significantly faster.

In this paper author of RGMA presents a new improved version of this method called Routes Generation Matrix Improved Algorithm (RGMIA) which has lower time-complexity than RGMA and generates more optimal routes than RGMA and RGGA. In this paper next section includes definition of optimal routes generation problem and description of public transport network model. In the third section author presents common idea of RGMA and RGMIA and illustrates it by a simple example. Section 4 is concerned on detail description of differences between RGMIA and RGMA. Subject of Section 5 is the comparison of effectiveness of RGMIA, RGMA and RGGA methods in two aspects: time complexity and quality of results. This comparison is based on experimental results which were performed on realistic data. The paper ends section with some remarks about future work on possibilities of further improving of RGMIA.

2. Network Model and Problem Definition

A public transportation network in our model is represented as a bimodal weighted graph $G = \langle V, E \rangle$ [8], where V is a set of nodes, E is a set of edges. Each node in G corresponds to a certain transport stop (bus, tram or metro stop, etc.), shortly named stop. We assume that stops are represented with numbers from 1 to n. The directed edge $(i, j, l, t) \in E$ is an element of the set E, the line number l connects the stop number i as a source point and the stop number j as a destination. One directed edge called transport link corresponds to one possibility of the connection between two stops. Each edge has a weight t which is equal to the travel time (in minutes) between nodes i and j which can be determined on the base of timetables. A set of edges is bimodal because it includes, besides directed links, undirected walk links. The undirected edge $\{i, j, t\} \in E$ is an element of the set E, if walk time in minutes between i and j stops is not greater than $limit_w$ parameter. The value of $limit_w$

parameter has a big influence on the number of network links (density of graph). The t value for undirected edge $\{i, j\}$ is equal to walk time in minutes between i and j stops. We assume, for simplification, that a walk time is determined as an Euclidian distance between stops.

A graph representation of public transportation network is shown in Fig. 1. It is a very simple example of the network which includes only nine stops. In the real world the number of nodes is equal to 3500 for the city with about 1 million of inhabitants.

Formal definition of our problem is the following. At the input we have: graph of transportation network, $timetable(l)$ - times of departures for each stops and line l, source point of the travel (o), destination point of the travel (d), starting time of the travel $(time_o)$, number of the resultant paths (k), maximum number of changes (max_t) and limit for walk links $(limit_w)$. At the output we want to have the set of resultant routes, containing at most k quasi-optimal paths with minimal time of realization (in minutes) with at most max_t changes. max_t parameter takes into account only transfers from one vehicle to another (not from vehicle to walk or inversely).

Weight of transport link (i, j, l, t) is strongly dependent on the starting time parameter $(time_o)$ and $timetable(l)$ which can be changed during the realization of the algorithm. The t value of (i, j, l, t) link is equal to the result of subtraction: time of arrival for line l to the stop j and start time for stop i - $time_i$.

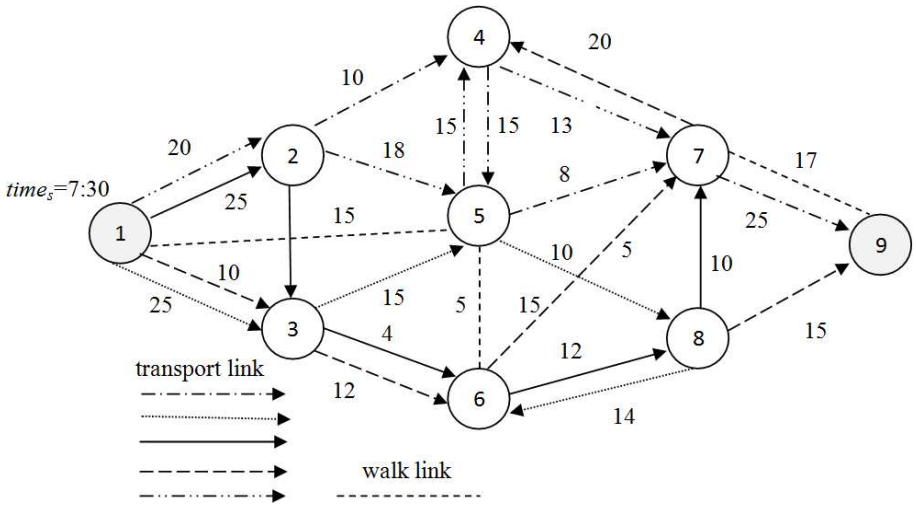

Fig. 1. Representation of a simple transportation network (different styles of lines mark different transport links; dot lines mark walk links)

3. The Common Idea of RGMA and RGMIA

RGMA and RGMIA algorithms of determining paths with the minimal travel time are based on label-setting method of the shortest paths generation. In order to find optimal paths, for $k \geq 1$, it is important to choose different routes throughout the network [15]. It can be realized by labeling nodes and edges or by removal of a node or an edge. Because it is easier to implement the labeling algorithms than the path deletion algorithms for the transportation network, the algorithm described in this section is based on the label-setting technique [2].

The idea of both methods is the same. Before first iteration of the algorithm we must labeled each node in the network. The initial value of label (marked as *et*) of the node o is equal to k ($et(o) = k$) and 0 for other nodes. In first step of the method we find the closest node u to the start point o. Node u is the closest to node o iff $H(o) = u$, where H is an heuristic which determines the choice of closest node. This heuristic is different in RGMA and RGMIA. The label of the closest node u is increasing at that moment. Next, we add to the graph G new arcs: from o to each node v which incidences with node u. The weight t_{ov} of the new arc is equal to $t_{ou} + t_{uv}$. Next step executes as the same way as the first step. The algorithm stops, when the label of the end node d is equal to k or there is no nodes closest to o. We have k (or less than k if there is no k paths from o to d in network) paths from o to d as a result of the method.

The common idea of RGMA and RGMIA is written in the psedocode form presented bellow. Line number six in this pseudocode realizes different heuristics H for RGMA and RGMIA which are detail described in Section 4.

```
 Pseudocode CommonIdea(G,o,d,timetables,k)
1:  Begin
2:    for i:=1 to n do et(i):=0;
3:    et(o):=k;
4:    while et(o)<k do
5:      Begin
6:        u:=H(o);
7:        if u not exist then break;
8:        if u=d then return route from o to d;
9:        add new arcs (o,v,t) to G for each node v which incidences with u
10:       et(u):=et(u)+1;
11:     End
12: End;
```

In Fig. 2 and Fig. 3 we illustrate first step of realization of common idea of RGMA and RGMIA on the example network presented in Fig. 1. In this presentation we assume that $u = H(o)$ iff there is a link from o to u and $et(u)$ is minimal in first order and travel time from o to u is minimal in second.

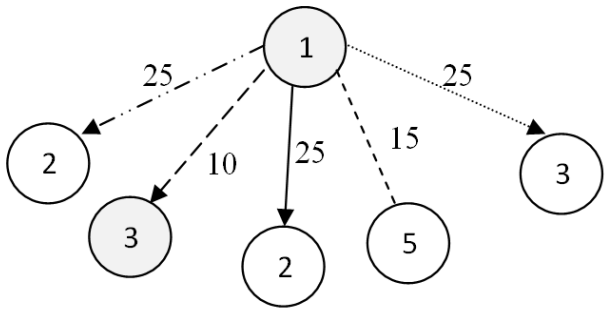

Fig. 2. Choice of the closest node in first step of pseudocode CommonIdea realized on network presented in Fig. 1.

If start node is equal to 1 and destination node is equal to 9 then the closest node in network presented in Fig. 1 is equal to 3, because of this node has a minimal value of travel time from node 1. Narrowly, there are five links between nodes 1 and 9. Four transport links: $(1,2,20)$,$(1,2,25)$,$(1,3,10)$, $(1,3,25)$ and one walk link $(1,2,20)$ (the number of line is omitted for simplification). The algorithm chooses node 3 as the closest node because tranport link $(1,3,10)$ has the smallest travel time.

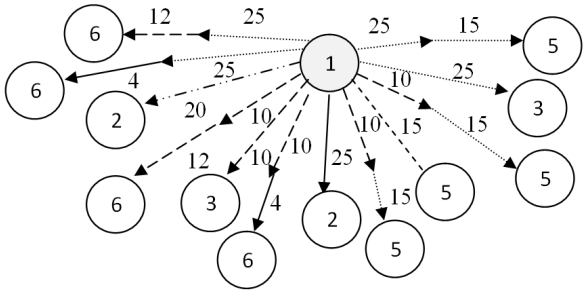

Fig. 3. Addition new arcs to the network in second step of RGMIA-RGMA realized on network in Fig. 1

In the second step of the algorithm seven new arcs are added to the graph: $(1,5,39)$,$(1,5,35)$,$(1,6,14)$,$(1,6,22)$,$(1,6,29)$,$(1,6,39)$. The travel time t for new link $(1,u,t)$ is equal to the sum of travel time from 1 to 3 travel time from 3 to u. There are twelve links begining in node 1 in the graph now.

The basic difference between RGMA and RGMIA rests on the heuristic H which determines conditions of the closest node choosing. We detail describe this difference in the next section.

4. Differences between RGMA and RGMIA

To detail present differences between RGMA and RGMIA we must define special matrices used in definition of H heuristic in both methods:

1. $Q = q[i,j]_{i,j=1..n}$ - minimal number of changes matrix. $q[i,j]$ element is equal to the number of minimal changes in route from stop i to stop j. The algorithm which determines Q matrix is detail presented in [8].
2. $D = d[i,j]_{i,j=1..n}$ - minimal number of stops matrix. $d[i,j]$ element is equal to the minimal number of stops in route from stop i to stop j. We can calculate D matrix using standard Breath First Search algorithm for each variant of start stop i in the network.
3. $T_{rh} = t_{rh}[i,j]_{i,j=1..n}$ - minimal travel-time in rush hours matrix. $t_rh[i,j]$ element is equal to the minimal travel time in rush hours in route from stop i to stop j. Rush hours are specific for a given city (from 7:00 a.m. to 10:00 a.m. and from 03:00 p.m. to 07:00 p.m. for example in Warsaw). We can obtain T_{rh} matrix on the base of fragment of timetables which concerned on rush hours.
4. $T_{oh} = t_{oh}[i,j]_{i,j=1..n}$ - minimal travel-time outside of rush hours matrix. $t_oh[i,j]$ element is equal to the minimal travel time outside of roush hours in route from stop i to stop j. We can determine T_{oh} matrix on the base of fragment of timetables which concerned on hours outside rush.

In practical implementation of RGMIA it is possible to determine more than two kinds of minimal travel-time matrix, dividing twenty-four hours into parts, taking into consideration an intensity of traffic - specific for a given city. It's very important that we can determine each of above matrix only one time - before first execution of RGMA or RGMIA. Therefore the time-complexity of algorithms which calculate these special matrices doesn't have an influence on time-complexity of RGMA and RGMIA.

Now, we can define H heuristics for our methods: H_{RGMA} and H_{RGMIA}. $H_{RGMA}(s) = u$ iff:

1. $\{o,u,t\} \in E$ and $et(u)$ is minimal in first order and
2. value of $t + D[o,u]$ is minimal or differs from minimal not greater than ε (ε is a parameter of closeness to o given on the input of the algorithm) in second order and
3. $Q[o,u]$ is minimal in third order.

$H_{RGMIA}(s) = u$ iff:

1. $\{o,u,t\} \in E$ and $et(u)$ is minimal in first order and
2. value $t + T_{rh}[o,u]$ is minimal or differs from minimal not greater than ε if parameter $time_o$ belongs to rush hours or value $t + T_{oh}[o,u]$ is minimal or differs from minimal not greater than ε if parameter $time_o$ doesn't belong to rush hours) in second order and
3. $Q[o,u]$ is minimal in third order.

Intuitively H_{RGMIA} heuristic may be a better heuristic then H_{RGMA} because of T_{rh} and T_{oh} matrices are time-dependent and give information about lower bound of travel time not only for first link in route but for a whole route. Matrix D used in H_{RGMA} is time-independent and therefore we can choose worse (then in RGMIA) closest node in each step of RGMA. This intuitively observation was confirmed by experimental results on real data.

5. Experimental Results

There were a number of computer tests conducted on real data of transportation network in Warsaw city. This network consists of about 4200 stops, connected by about 240 bus, tram and metro lines. Values of common parameters for RGMA, RGMIA and RGGA were following: $max_t = 5$, $k = 3$, $limit_w = 15$. The value of $limit_w$ is very important because it influences the density of network. The bigger value of $limit_w$, the more possibilities of walk links in a network. Density of network is of a key importance for time-complexity of algorithms. The parameter of closeness ε in RGMA and RGMIA was equal to 5 minutes. This parameter also has an influence on quality of results and time-complexity of these methods. The bigger value of ε, the more possibilities of choice of closest node. We performed three kinds of tests. We examined routes from the centre of the city to the periphery of the city (set $C-P$), routes from the periphery of the city to the centre of the city (set $P-C$) and routes from the periphery of the city to the periphery of the city (set $P-P$). Each of these sets includes 50 specification of first (o) and last (d) stops in the route which are difficult cases for each algorithm. First matter is a long distance from o to d (in $P-P$

set), the second is a high density of the network in o or d localization (in $C-P$ and $P-C$ sets).

Selected results of tests for 10 chosen specification of o and d for each examined set of routes, generated by RGMIA, RGMA and RGGA, are presented in Tab. 1, 2, 3. For each of algorithm we show in these tabels: o and d specification, minimal travel-time for k resultant routes (RGMIA-t,RGGA-t, RGMA-t), number of changes for route with minimal travel-time (RGMIA-ch, RGGA-ch, RGMA-ch).

Table 1. The results for routes from set $P-C$; $time_s = 7:30$; o = Skolimowska 02

d-destination stop	RGMIA-t	RGGA-t	RGMA-t	RGMIA-ch	RGGA-ch	RGMA-ch
Pl.na Rozdrożu 01(al. Ujazd.)	57	59	61	2	2	2
Pl.Konstytucji 04(Piękna)	62	63	71	2	2	2
GUS 08(Wawelska)	60	62	66	3	1	1
Dw. Centr.17(Świętokrz.)	64	66	69	2	1	1
Mennica 01(Grzybowska)	79	80	81	3	3	3
Ordynacka 02(Nowy Świat)	62	66	64	2	2	2
Siedmiogrodzka 02(Grzybowska)	78	80	82	3	3	3
Młynarska 04(Młynarska)	71	79	81	3	3	3
Nowy Świat 04(Świętokrzyska)	65	67	69	2	2	2
Emilii Platter(PŁN. Dw. Central)	71	73	73	3	3	3

Table 2. The results for routes from set $C-P$; $time_s = 15:30$; o = Mennica 01 (Grzybowska)

d-destination stop	RGMIA-t	RGGA-t	RGMA-t	RGMIA-ch	RGGA-ch	RGMA-ch
Plantanowa(Młochów)	104	112	124	3	4	4
Ogodowa 01(Głosków)	82	83	97	3	3	3
Kępa Okrzewska 01	64	84	88	3	4	4
Dziechciniec-Sklep02	103	86	88	3	3	3
Wąska(Józefów)	81	81	84	2	3	3
Struga 02(Marki)	63	64	63	2	2	2
Stokrotki 02(Nieporęt)	75	78	83	3	2	2
Orzechowa 02(Łopuszańska)	35	37	44	3	3	2
Długa 02(Dawidy)	67	73	84	3	3	3
3 Maja(Legionowo)	66	68	71	3	2	3

All results presented in above tables are confirmation of good quality of routes of RGMIA algorithm because the values of travel-time for the best and worst route are significantly less than for other comparable method. Generally, for $P-C$ set, in

Table 3. The results for routes from set $P-P$; $time_s = 7:30$; o = Struga 01(Marki)

d-destination stop	RGMIA-t	RGMA-t	RGGA-t	RGMIA-ch	RGMA-ch	RGGA-ch
Plantowa(Pruszków)	76	88	91	3	4	4
Ogrogowa 01(Głosków)	117	137	149	4	4	4
Kępa Okrzewska 01	103	123	138	5	5	5
Dziechciniec-Sklep 02	99	149	181	3	4	5
Mennica 01(Grzybowska)	79	80	81	3	3	3
Wąska(Józefów)	96	108	111	3	4	4
Stokrotki 02(Nieporęt)	77	84	93	0	0	4
Orzechowa 02(Łopuszańska)	75	78	81	4	5	5
Długa 02(Dawidy)	105	106	120	5	4	4
3 Maja(Legionowo)	82	87	96	3	4	4

29 cases of $o-d$ specification RGMIA generates the best routes, in 17 cases RGGA was the best and only in 4 cases RGMA resultant routes were the best. For $P-C$ set, in 26 of cases of $o-d$ specification RGMIA generates the best routes, in 21 of cases RGGA was the best and only in 3 cases RGMA resultant routes were the best. For $P-C$ set, in 27 of cases of $o-d$ specification RGMIA generates the best routes, in 23 of cases RGGA was the best and only in 0 cases RGMA resultant routes were the best.

The last experiment was focused on comparison of time complexity of algorithms. The results are presented in Fig. 4.

In this experiment we tested examples of routes with a minimal number of stops, between 22 and 47. On the horizontal axis there are points representing the minimal number of stops on a route. These values were computed as a result of standard *BFS* graph search method and they are correlated with difficulty of the route. On the vertical axis there is marked time of execution in ms (processor Pentium 3.0 GHz). Each possible route with a given number of the minimal number of bus stops was tested by two algorithms at starting time at 7:30 a.m., weekday. The executing time of algorithms was averaged over every tested routes. We can see that RGMIA performs in significantly shorter time than RGMA and insignificantly than RGGA, specially for routes with minimal number of stops greater than 35.

We can conclude on the base of our experiments that RGMIA returns results better then RGMA and even RGGA and is significantly faster than its previous version.

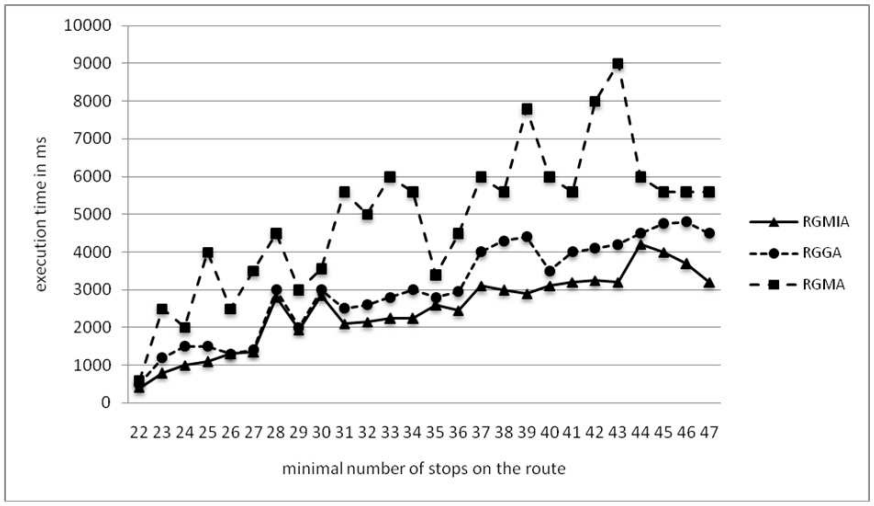

Fig. 4. The comparison of time-complexity of RGMIA, RGMA and RGGA

6. Conclusions

The author's motivation was to try to improve the RGMA in two aspects: the quality of resultant routes and time-complexity[10]. Computer experiments have shown that RGMIA - improved version of RGMA performs much more better than RGMA and significantly faster and is unexpectedly better than RGGA in both examined aspects.

Future work will be concentrated on testing RGMIA and RGGA on another transport networks for big metropolises which have different size, density and topology than network for Warsaw topology, such as Gornoslaski Okrag Przemyslowy (Silesian Industrial Region). The transport network of this region is very special because it consists of many big cities (hubs) connected by very rare fragment of network. If tests show poor performance of RGMIA or/and RGGA the new heuristics must be added to the algorithm. The proposal of improvement which can be considered includes to the algorithm information about geographic location of start and destination stops.

References

[1] Ahuja, R. K., Orlin, J. B., Pallotino, S., Scutella, M.G.: Dynamic shortest path minimizing travel times and costs, Networks, 41 (4), 2003, pp. 197-205.

[2] Ahuja R. K., Magnanti T. L., Orlin, J. B.: Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 2008.
[3] Bellman R. E.: On a Routing Problem, Journal Quarterly of Applied Mathematics 16, 1958, pp. 87-90.
[4] Chabini I.: Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time, Journal Transportation Research Records, 1998, pp. 170-175.
[5] Cooke K.L., Halsey, E.: The shortest route through a network with time-dependent intermodal transit times, Journal Math. Anal.Appl., 14, 1996, pp. 493-498.
[6] Dreyfus, S.E.: An Appraisal of Some Shortest-path Algorithms, In Journal Operations Research, vol.17, 1969, pp. 395-412.
[7] Hansen P.: Bicriterion path problems. In Multicriteria decision making: theory and applications, Lecture Notes in Economics and Mathematical Systems 177, Eds. G. Fandel, T. Gal. Heidelberg, Springer-Verlag 1980, pp. 236-245.
[8] Koszelew, J.: The Theoretical Framework of Optimization of Public Transport Travel, In: Proceedings of 6th International Conference on Computer Information Systems and Industrial Management Applications: CISIM 2007, IEEE Computer Society, 2007, pp. 65-70.
[9] Koszelew, J.: Approximation method to route generation in public transportation network, Polish Journal Environment Studies, Vol.17, Nr 4C, 2008, pp. 418-422.
[10] Koszelew, J., Piwonska, A.: A new genetic algorithm for optimal routes generation in public transport network, Proceedings of 13th International Conference on System Modelling Control: SMC'2009, Lodz University of Technology, 2009.
[11] Lawler, E. L.: A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem, Management Science 18, 1972, pp. 401-405.
[12] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs WNT, Warsaw, 1996.
[13] Orda, A. and Rom, R.: Shortest path and minimum - delay algorithms in networks with time-dependent edge-length, In Journal Assoc. Computer Mach, 37(3), 1990, pp. 607-625.
[14] Sung, K., Bell, M.G.H., Seong, M. and Park, S.: Shortest paths in a network with time-dependent flow speeds, European Journal Operational Research, 121(1), 2000, pp. 32-39.
[15] WU, Q., Hartley J. K.: Accommodating User Preferences in the Optimization of Public Transport Travel, International Journal of Simulation Systems, Science

and Technology: Applied Modeling and Simulation, Vol.5, No 3-4, 2004, pp. 12-25.

[16] Wikipedia, the free encyclopedia [http://en.wikipedia.org/wiki/Intelligent_transportation_system].

POPRAWIONA WERSJA PEWNEGO APROKSYMACYJNEGO ALGORYTMU GENERUJĄCEGO OPTYMALNE TRASY W SIECI TRANSPORTU PUBLICZNEGO

Streszczenie Artykuł zawiera opis poprawionej wersji algorytmu generującego optymalne trasy w sieci transportu publicznego uzupełnionej o linki piesze, nazywanego przez autora Routes Generation Matrix Improved Algorithm (RGMIA). Trasy generowane przez RGMIA są optymalne pod względem czasu realizacji i mogą zawierać odcinki piesze, co sprawia, że wynikowe ścieżki są bardziej praktyczne i mogą spełniać określone preferencje użytkowników środków transportu. Algorytm został zaimplementowany i przetestowany na danych realnej sieci transportowej. Efektywność poprawionej metody została porównana w dwóch aspektach: złożoności czasowej i jakości wynikowych tras, z poprzednią wersją algorytmu nazwaną Routes Generation Matrix Algorithm (RGMA) oraz z metodą genetyczną Routes Generation Genetic Algorithm (RGGA). Algorytmy RGMA oraz RGGA zostały opisane w poprzednich artykułach autora [9,10].

Słowa kluczowe: sieć transportu publicznego, problem najkrótszych ścieżek zmiennych w czasie, optymalne trasy, algorytm genetyczny

Artykuł zrealizowano w ramach pracy badawczej S/WI/3/08

LEARNING FINITE GAUSSIAN MIXTURES USING DIFFERENTIAL EVOLUTION

Wojciech Kwedlo[1]

[1]Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: In the paper the problem of parameter estimation of finite mixture of multivariate Gaussian distributions is considered. A new approach based on differential evolution (DE) algorithm is proposed. In order to avoid problems with infeasibility of chromosomes our version of DE uses a novel representation, in which covariance matrices are encoded using their Cholesky decomposition. Numerical experiments involved three version of DE differing by the method of selection of strategy parameters. The results of experiments, performed on two synthetic and one real dataset indicate, that our method is able to correctly identify the parameters of the mixture model. The method is also able to obtain better solutions than the classical EM algorithm.

Keywords: Gaussian mixtures, differential evolution, EM algorithm.

1. Introduction

Mixture models [13] are versatile tools used for modeling complex probability distributions. They are capable to model observations, which are produced by a random data source, randomly selected from many possible sources. Estimation of the parameters of these sources and identifying which source produced each observation leads to clustering of data [10]. Mixture models can be also used for feature selection [17] or representation of class-conditional probability density functions in discriminant analysis [9].

A finite mixture model $p(\mathbf{x}|\Theta)$ can be described by a weighted sum of $M > 1$ components $p(\mathbf{x}|\theta_m)$:

$$p(\mathbf{x}|\Theta) = \sum_{m=1}^{M} \alpha_m p(\mathbf{x}|\theta_m), \tag{1}$$

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 19-33, 2010.

where $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$ is the d-dimensional data vector, $\alpha_1, \alpha_2, \ldots, \alpha_M$ are mixing probabilities, which satisfy the following conditions:

$$\alpha_m > 0, \;\; m = 1, \ldots, M \text{ and } \sum_{m=1}^{M} \alpha_m = 1.$$

θ_m is the set of parameters defining the mth component and $\Theta = \{\theta_1, \theta_2, \ldots, \theta_M, \alpha_1, \alpha_2, \ldots, \alpha_M\}$ is the complete set of the parameters needed to define the mixture. In the paper we consider a class of finite mixture models called Gaussian mixtures in which each component $p(\mathbf{x}|\theta_m)$ follows multivariate normal (Gaussian) distribution:

$$p(\mathbf{x}|\theta_m) = \frac{1}{(2\pi)^{d/2}|\Sigma_m|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu_m)^T \Sigma_m^{-1}(\mathbf{x} - \mu_m)), \tag{2}$$

where μ_m and Σ_m denote the mean vector and covariance matrix, respectively, $|\cdot|$ denotes a determinant of a matrix. The set parameters of the mth component is $\theta_m = \{\mu_m, \Sigma_m\}$. The set of the parameters of the complete Gaussian mixture model can be defined as:

$$\Theta = \{\mu_1, \Sigma_1, \ldots, \mu_M, \Sigma_M, \alpha_1, \ldots, \alpha_M\}. \tag{3}$$

Estimation of the parameters of the mixture model is usually performed using the maximum likelihood (ML) approach. Given a set of independent and identically distributed samples $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$, called the training set, the log-likelihood corresponding to M-component mixture is given by:

$$\log p(X|\Theta) = \log \prod_{i=1}^{N} p(\mathbf{x}^i|\Theta) = \sum_{i=1}^{N} \log \sum_{m=1}^{M} \alpha_m p(\mathbf{x}^i|\theta_m). \tag{4}$$

The maximum likelihood estimate of the parameters

$$\Theta_{ML} = \operatorname*{argmin}_{\Theta} \{-\log p(X|\Theta)\}$$

cannot be found analytically (e.g. [3]). Therefore, the estimation must be performed using an optimization algorithm.

In the paper a novel application a global optimization algorithm called differential evolution (DE) to the problem of Gaussian mixture learning is proposed. The rest of this paper is organized as follows. Section 2 discusses the research related to our work. In Section 3 the details of the proposed application of DE to the problem of Gaussian mixture learning are described. Section 4 presents the results of computational experiments. The last section of this paper contains conclusions.

2. Related Work

The usual choice of obtaining ML parameters of a Gaussian mixture model is the EM algorithm [6]. It is iterative procedure for minimizing $-\log p(X|\Theta)$. The EM algorithm for Gaussian mixtures is easy to implement and computationally efficient [13]. However, it has an important drawback. Being a local search algorithm, it can be easily trapped in a local minimum of $-\log p(X|\Theta)$. Thus, the quality of the obtained solutions is highly dependent on the initialization of the EM algorithm. The most simple solutions include using multiple random starts and choosing the final estimate with the smallest $-\log p(X|\Theta)$ [13] and initialization by the clustering (e.g. *k*-means) algorithms [3,13].

Many more elaborate extensions of the EM approach have been suggested in order to tackle the problem of convergence to a local optimum. Verbeek et. al. [20] proposed a deterministic greedy method in which the mixture components are inserted into the mixture one after another. The method starts with the optimal one-component mixture, the parameters of which can be easily found. After each insertion of a new component the EM algorithm is applied to the new mixture. This approach is much lest sensitive to initialization than the original EM algorithm. Figueiredo and Jain [7] also used component-wise approach. Using minimum message length criterion they developed a method, which is robust with respect to the initialization and capable of discovering the number of the components. In the paper by Ueda et al. [19] the EM algorithm was extended by a split-and-merge technique in order to alleviate the problem of local optima.

Evolutionary algorithms (EAs) [14] are stochastic search techniques inspired by the concept of the Darwinian evolution. Unlike local optimization methods, e.g. the EM algorithm, they simultaneously process a population of problem solutions, which gives them the ability to escape from local optima of the fitness function. Recently, applications of EAs to the problem of ML estimation of parameters of a Gaussian mixture model have been proposed. Most of the researches used a hybrid scheme, which alternates between a step of EA consisting of selection and recombination and a step consisting of iterations of EM. This approach was used by Martinez and Vitrià [12], who employed selection and the mutation operators of evolution strategies [2]. In their method the value of the fitness function is obtained by running the EM algorithm (until it the converges) on mixture parameters encoded by a chromosome. Pernkopf and Bouchaffra [15] proposed a combination of an EA with EM, which by using fitness function based on the minimum description length principle, is able to estimate the number of the components in a mixture.

Differential evolution (DE) is an evolutionary algorithm proposed by Storn and Price [18], employing a representation based on real-valued vectors. It has been successfully applied to many optimization problems. DE is based on the usage of vector differences for perturbing the population elements. Many researches suggested extensions to the original DE. For an overview of recent developments and practical applications of DE the reader is referred to [5]. The version of DE with the self-adaptation of control parameters, which we employ in the paper, was proposed by Brest et al. [4]. According to our knowledge no application of DE to the problem of Gaussian mixture learning has been proposed so far.

3. DE algorithms for Gaussian mixtures

3.1 Differential evolution

Several versions of DE have been proposed. For the purpose of this study the most common variant is used, which, according to the classification proposed by [18] can be described as DE/rand/1/bin.

Like all EAs, DE maintains a population of S solutions of the optimization problem. At the start of the algorithm, members of the population are initialized randomly with the uniform distribution. Then DE performs multiple iterations in three consecutive steps: reproduction (creation of a temporary population), computing of the fitness function for all members of the temporary population, and selection.

Let $u_{i,G}$ denote the i-th member ($i = 1,\ldots,S$) of the population in the G-th iteration. Usually $u_{i,G}$ takes a form of a D-dimensional real-valued vector, i.e. $u_{i,G} \in \Re^D$.

Reproduction in DE creates a temporary population of trial vectors. For each solution $u_{i,G}$ a corresponding trial vector $y_{i,G}$ is obtained by mutation and crossover operators. The mutation operator generates a mutant vector $y'_{i,G}$ according to the equation:

$$y'_{i,G} = u_{a,G} + F * (u_{b,G} - u_{c,G}), \tag{5}$$

where $F \in [0,2]$ is a user-supplied parameter called amplification factor and $a,b,c \in 1,\ldots,S$ are randomly selected in such way that $a \neq b \neq c \neq i$.

The final trial vector $y_{i,G}$ is obtained by the crossover operator, which mixes the mutant vector $y'_{i,G}$ with the original vector $u_{i,G}$. Let us assume that $u_{i,G} = (u_{1i,G}, u_{2i,G}, \ldots, u_{Di,G})$. Each element $y_{ji,G}$ (where $j = 1,\ldots,D$) of the trial vector $y_{i,G}$ is generated as:

$$y_{ji,G} = \begin{cases} y'_{ji,G} & \text{if } rnd(j) < CR \text{ or } j = e \\ u_{ji,G} & \text{otherwise} \end{cases}. \tag{6}$$

where $CR \in [0,1]$ is another user-supplied parameter called crossover factor, $rnd(j)$ denotes a random number from the uniform distribution on $[0,1]$ which is generated independently for each j. $e \in 1,\ldots,S$ is a randomly chosen index which ensures that at least one element of the trial vector $y_{i,G}$ comes from the mutant vector $y'_{i,G}$.

The remaining two phases of DE generation are computation of the fitness for all members of the trial population and selection. Selection in differential evolution is very simple. The fitness of each trial solution $y_{i,G}$ is compared to the fitness of the corresponding original solution $u_{i,G}$. The trial vector $y_{i,G}$ survives into the next iteration becoming $u_{i,G+1}$ if its fitness is better. Otherwise $y_{i,G}$ is discarded and $u_{i,G+1}$ is set to $u_{i,G}$

3.2 Choice of control parameters F and CR

The parameters F and CR have a significant impact on convergence speed and robustness of the optimization process. The choice of their optimal values is an application-dependent task. In [18] the values $F = 0.5$ and $CR = 0.9$ were suggested. F and CR may be also selected using a trial-and-error approach, which requires many optimization runs and may be infeasible in many practical applications.

To alleviate the problem of parameter tuning, Brest et al. [4] proposed a self-adaptation method for F and CR. In this method, amplification and crossover factors evolve together with population members. Each member of the both trial and target populations is augmented with its own amplification factor and crossover factor. Let us denote by $F^u_{i,G}$ and $F^y_{i,G}$ the amplification factors associated with the vectors $u_{i,G}$ and $y_{i,G}$, respectively. Similarly, let us denote by $CR^u_{i,G}$ and $CR^y_{i,G}$ the crossover factors associated with the vectors $u_{i,G}$ and $y_{i,G}$, respectively.

Before the mutation $F^y_{i,G}$ is generated as:

$$F^y_{i,G} = \begin{cases} L + rnd_2 * U & \text{if } rnd_1 < \tau_1 \\ F^u_{i,G} & \text{otherwise} \end{cases}. \tag{7}$$

rnd_1 and rnd_2 are uniform random values from $[0,1]$, $\tau_1 \in [0,1]$ is the probability of choosing new random value of $F^y_{i,G}$, L and U are the parameters determining the range for $F^y_{i,G}$.

Similarly to $F^y_{i,G}$, $CR^y_{i,G}$ is generated before the mutation as:

$$CR^y_{i,G} = \begin{cases} rnd_3 & \text{if } rnd_4 < \tau_2 \\ CR^u_{i,G} & \text{otherwise} \end{cases}, \tag{8}$$

where $\tau_2 \in [0,1]$ is the probability of choosing new random value of $CR^y_{i,G}$.

$F^y_{i,G}$ obtained by (7) is used to generate the mutant vector according to (5). $CR^y_{i,G}$ obtained by (8) is used to generated the trial vector according to (6). The amplification and crossover factors undergo selection together with associated vectors. If in the selection process $u_{i,G+1}$ is set to $y_{i,G}$ then $F^u_{i,G+1}$ is set to $F^y_{i,G}$ and $CR^u_{i,G+1}$ is set to $CR^y_{i,G}$. Otherwise, $F^u_{i,G+1}$ is set to $F^u_{i,G}$ and $CR^u_{i,G+1}$ is set to $CR^u_{i,G}$.

It may seem that self-adaption of F and CR introduces another four parameters (L, U, τ_1, τ_2) which must be fine-tuned by the user by means of the trial-and-error method. However, Brest et al. [4] used fixed values of these parameters in all experiments and obtained promising results. Following their suggestion in our experiments we used $\tau_1 = \tau_2 = 0.1$. L and U were set to 0.05 and 0.35 respectively, which ensured that $F^y_{i,G} \in [0.05, 0.4]$.

The original method of Brest et. al. self-adapted both parameters F and CR. It was tested on benchmark numerical optimization problems, with dimension $D \leq 30$. However, our preliminary experiments indicated, that it yielded poor results, when applied to the much more difficult problem of Gaussian mixture learning. Therefore in the paper we propose to use a new approach, in which F is self-adapted in a manner described in [4], whereas CR is set to a constant value close to 0 according to the following experimentally developed formula: $CR = 2/D$, where D is the dimension of a population element.

3.3 The fitness function

The fitness function used by the DE is $-\log p(X|\Theta)$, where $\log p(X|\Theta)$ is defined by (4). Because of the minus sign, the algorithm is configured to minimize the fitness.

3.4 Encoding of the parameters of a Gaussian mixture model by chromosomes

In order to apply a DE algorithm to the problem of Gaussian mixture learning we have to work out a method for encoding mixture parameters (3) by real-valued vectors (chromosomes). The encoding of mixing probabilities $\alpha_1, \alpha_2, \ldots, \alpha_M$ and mean vectors $\mu_1, \mu_2, \ldots, \mu_M$ is very straightforward. The mixing probabilities (M real values) and all the elements of mean vectors (dM real values) are directly encoded in a chromosome using the floating-point representation.

Unfortunately, the elements of covariance matrices cannot be encoded in a similar way. A covariance matrix Σ_m, $m = 1, \ldots, M$ of each Gaussian component of a mixture (1) must be a symmetric positive-definite matrix, i.e. for each non-zero $\mathbf{x} \in \Re^d$ $\mathbf{x}^T \Sigma_m \mathbf{x} > 0$ [11]. Since, each Σ_m is symmetric, only its $d(d+1)/2$

diagonal and over-diagonal elements need to be encoded. However, if we encoded these elements directly, the purely random crossover and mutation operators of DE would not preserve positive-definiteness of the covariance matrices Σ_m. In almost all cases the matrices encoded in a trial vector $y_{i,G}$ would not be positive-definite and thus could not be interpreted as covariance matrices.

To overcome the above obstacle, the covariance matrices are encoded in a chromosome using their Cholesky decomposition [16]. Each covariance matrix Σ_m, $m = 1,\ldots,M$ can be expressed as a product of a lower triangular matrix L_m and its transpose:

$$\Sigma_m = L_m L_m^T. \tag{9}$$

The diagonal elements of the lower triangular matrix L_m must be strictly positive. This condition is much easier to satisfy during the DE evolution, than the positive-definiteness of the covariance matrix Σ_m. For that reason we have chosen to encode elements of L_m rather than Σ_m in a chromosome. To ensure the positive-definiteness of Σ_m we inspect the trial vector $y_{i,G}$ and check each diagonal element of L_m. If the value of the diagonal element is lower than zero, it is replaced in the trial vector $y_{i,G}$ by its absolute value.

4. Experiments

4.1 Experimental setup

In the experiments we used three versions of DE differing by the method of parameter control:

- A version using fixed values of the parameters (i.e. $F = 0.5, CR = 0.9$ as proposed in [18]) during the run of the algorithm.
- A version using self-adaptive F and CR [4] described in the subsection 3.2.
- A version using self-adaptive F only.

Additionally, the EM algorithm [13], which is the standard method for estimation of the parameters of the Gaussian Mixture Model was used in the comparison.

In all the experiments the population size S was set to 32. All the results reported in this section are the average over 30 independent runs using random starting conditions. The EM algorithm was initialized using a random method described in [7].

All the algorithms were implemented in the C++ language and compiled by the Intel C++ version 10.1 compiler using optimizing options (-O3 -ipo -xT -fno-alias). It was run on a system with two quad-core Intel Xeon 5355 (2.66 GHz) processors.

The comparison of the DE algorithms was performed on the equal CPU time basis: all three versions of DE were were allocated 200 seconds of CPU time, and the final result was the fitness of the best individual in the population obtained after 200 seconds of evolution.

4.2 Synthetic datasets

The first example uses 900 samples from 3-component bivariate mixture from [7]. For this example $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$, $\mu_1 = [0,-2]^T$, $\mu_2 = [0,0]^T$, $\mu_3 = [0,2]^T$ and all three covariance matrices are equal: $\Sigma_1 = \Sigma_2 = \Sigma_3 = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 0.2 \end{smallmatrix}\right]$.

Fig. 1a shows the true mixture and 900 samples of it, whereas Fig. 1b shows the mixture estimated by the DE with self-adaptive F (mixtures estimated by the other two versions were almost identical). It can be seen, that the DE algorithm has correctly identified the mixture parameters. Fig. 2 shows the convergence of three versions of DE. The curves represent the fitness of the best individual in the DE population plotted as a function of the number of fitness function calls performed by the algorithms. For that dataset all the algorithms converged towards global minimum, although the DE with self-adaptive F converged slower than the other two methods.

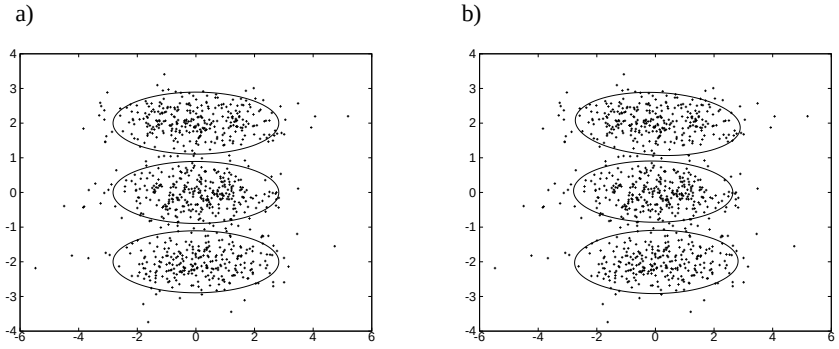

Fig. 1. Estimation of parameters of 3-component bivariate mixture. The solid ellipses are level-curves of each component: a) true mixture b) mixture estimated by DE with self-adaptive F.

In the second example, the mixture components overlap. Two of them share a common mean, but have different covariance matrices. The 1000 samples were taken from the mixture with following parameters: $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$ and

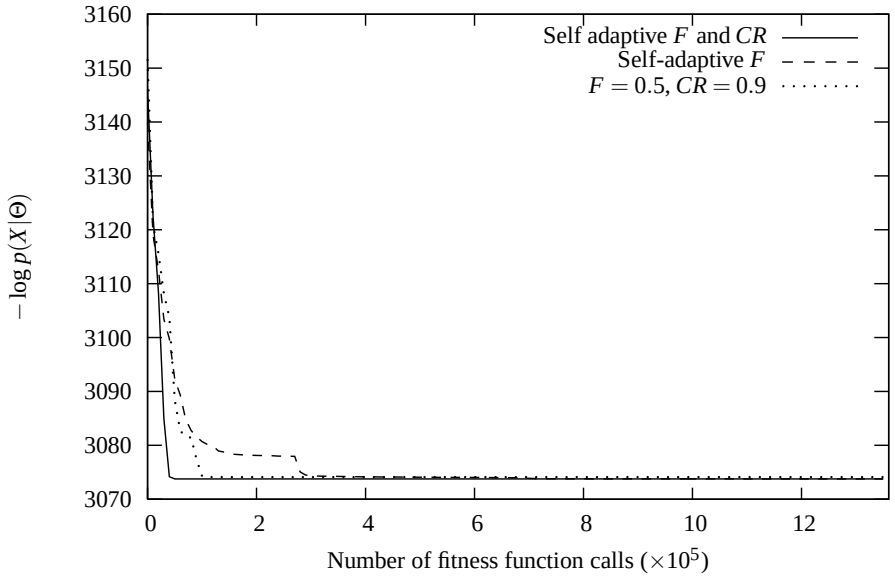

Fig. 2. Convergence of three versions of DE for the 3-component mixture. The curves are average over 30 independent runs.

$\mu_1 = \mu_2 = [-4,4]^T$, $\mu_3 = [2,2]^T$, $\mu_4 = [-1,-6]^T$ and $\Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 6 & -2 \\ 2 & 6 \end{bmatrix}$, $\Sigma_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$, $\Sigma_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}$. Fig. 3a shows the true mixture and 1000 samples of it, whereas Fig. 3b shows the mixture estimated by DE with self-adaptive F. Once again it can be seen, that the algorithm has correctly identified the mixture parameters. Fig. 4 shows the convergence of three variants of DE. This example was clearly too difficult for the DE with constant vales of F and CR as it was unable to find the global minimum of the log-likelihood. It can be also seen, that DE with self adaptive F and CR converged slightly faster than DE with self-adaptive F only.

4.3 Real-life dataset

In the last experiment, the well-known iris dataset (150 samples, $d = 4$) [8] describing iris flowers from the Gaspé peninsula was used. This dataset was obtained from the UCI machine learning repository [1]. The iris flowers described in this dataset are divided into three classes. For that reason we used $M = 3$. Of course, since it is a real-life, not synthetic dataset, the underlying probability density functions are not known and could not be visualized. Therefore Fig. 5 shows only the mixture estimated by

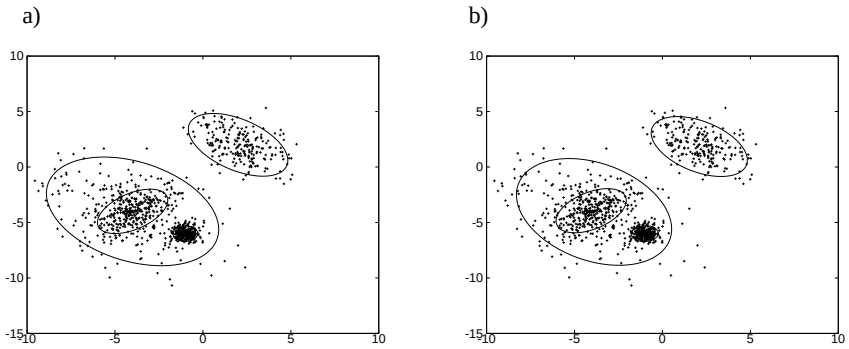

Fig. 3. Estimation of parameters of 4-component bivariate Gaussian mixture with overlapping components. The solid ellipses are level-curves of each component: a) true mixture b) mixture estimated by DE with self-adaptive F.

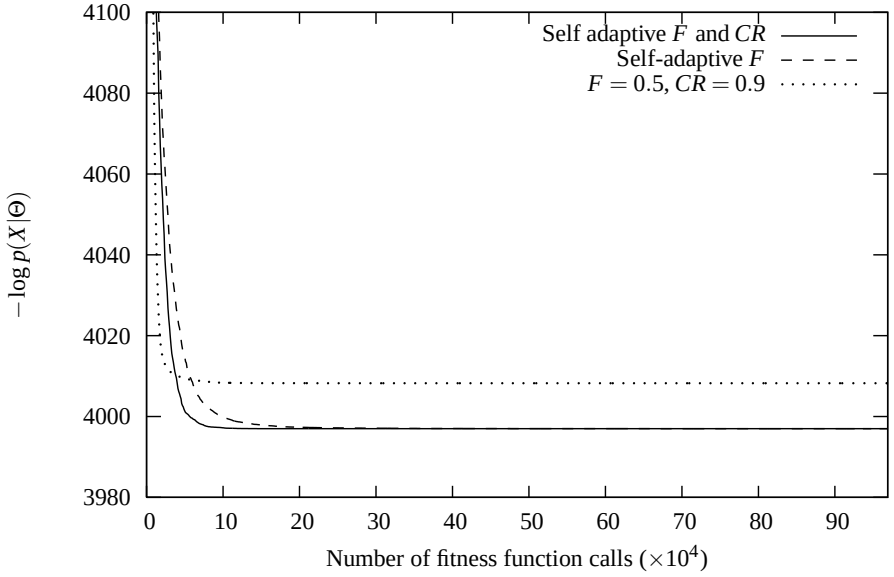

Fig. 4. Convergence of three versions of DE for the 4-component mixture with overlapping components. The curves are average over 30 independent runs.

the DE with self-adaptive F. Since iris is four-dimensional dataset, we used principal component analysis (PCA) [11] to project data and the estimated mixture on the first

two principal components. Fig. 6 shows the convergence of three versions of DE. This experiment clearly demonstrated that fast convergence in the initial phase of the evolution does not necessarily lead to best final solution, as the slowest algorithm (i.e. DE with self-adaptive F) found the solution with the lowest $-\log p(X|\Theta)$.

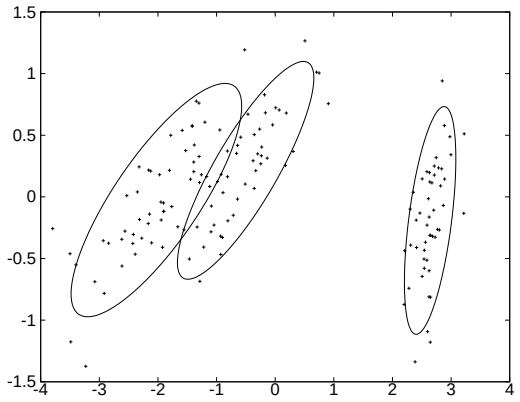

Fig. 5. iris data (projected on two principal components) and the mixture estimated by the DE with self-adaptive F. The solid ellipses are level-curves of each component.

4.4 Summary of the experiments and the comparison with the EM algorithm

The experiments are summarized by the Table 4.4, which shows the final $-\log p(X|\Theta)$ obtained by the DE algorithms after 200 seconds of evolution. Additionally, the last column of the table shows the result obtained by the EM algorithm [13]. The results suggest the following conclusions:

- All versions of DE outperform the local search algorithm (EM). However it should be noted that EM algorithm is much faster than DE, requiring much less than one second (on our hardware) to converge.
- DE with self-adaptive F only outperforms two other versions of DE.

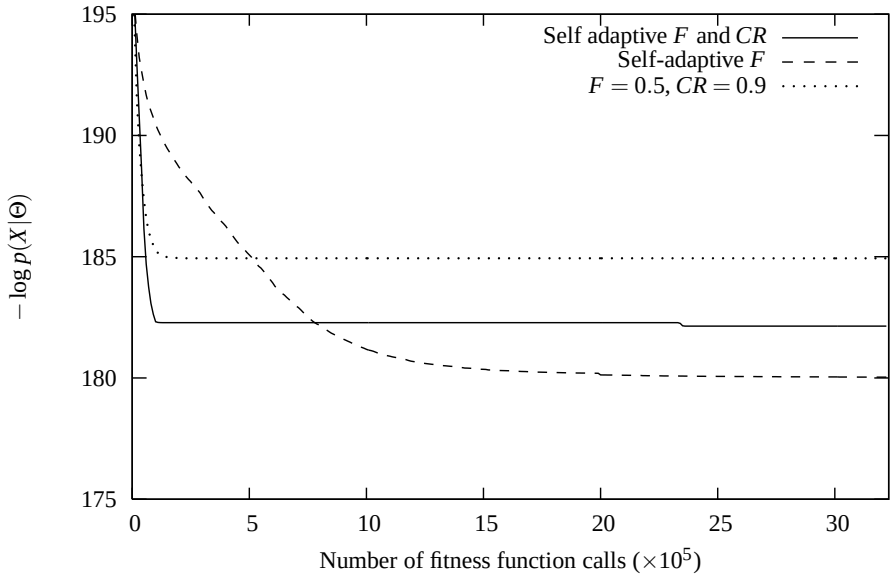

Fig. 6. Convergence of three versions of DE for the iris dataset. The curves are average over 30 independent runs.

5. Conclusions

In this paper an application of DE to the problem of Gaussian mixture learning was described. To avoid a problem with infeasibility of chromosomes we proposed a novel encoding method, in which covariance matrices were encoded using their Cholesky decomposition. In the experiments three versions of DE, differing by the method of parameter control, were examined and compared to the EM algorithm.

The results of our study allow us to recommend DE with self-adaptive F over the other two versions and the EM algorithm. Although it converges slowly in the initial phase of evolution, it is able to find solutions not worse (and sometimes better, if the problem is difficult enough as it was demonstrated by the experiment with the iris dataset) than other versions of DE.

There are several possible directions of future work. One of them is the development of a memetic algorithm combining DE with fast local search procedure (e.g. the EM algorithm). Such hybrid method would be able to retain the advantages of both approaches i.e. the fast convergence of EM and the ability find a global optimum of DE.

Table 1. The final $-\log p(X|\Theta)$ values obtained by the four tested methods. The results are average over 30 independent runs.

Dataset	$F = 0.5, CR = 0.9$	Self adaptive F and CR	Self adaptive F	EM
3-component mixture	3074.1	3073.8	3073.8	3148.8
4-component mixture	4008.2	3997	3997	4085.6
iris	184.93	182.14	180.03	187.58

Another direction of future research is related to the development of a parallel version of our approach. The most time-consuming step of the algorithm is computation of the fitness function since it requires the M passes over the training set. It would be quite straightforward to parallelize this step by concurrently computing the fitness of the different population members on different processors of a parallel system.

References

[1] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[2] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[4] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.

[5] U. K. Chakraborty, editor. *Advances in Differential Evolution*. Springer, 2008.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[7] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[8] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.

[9] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):155–176, 1996.

[10] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[11] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 6th edition, 2007.

[12] A. M. Martinez and J. Vitria. Learning mixture models using a genetic version of the EM algorithm. *Pattern Recognition Letters*, 21(8):759–769, 2000.

[13] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley and Sons, 2000.

[14] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1996.

[15] F. Pernkopf and D. Bouchaffra. Genetic-based EM algorithm for learning Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1344–1348, 2005.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.

[17] P. Pudil, J. Novovicova, N. Choakjarernwanit, and J. Kittler. Feature selection based on the approximation of class densities by finite mixtures of special type. *Pattern Recognition*, 28(9):1389–1398, 1995.

[18] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[19] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.

[20] J. J Verbeek, N. Vlassis, and B. Kröse. Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15(2):469–485, 2003.

UCZENIE SKOŃCZONYCH MIESZANIN ROZKŁADÓW NORMALNYCH PRZY POMOCY ALGORYTMU EWOLUCJI RÓŻNICOWEJ

Streszczenie W artykule rozważono problem uczenia parametrów skończonej mieszaniny wielowymiarowych rozkładów normalnych. Zaproponowano nową metodę uczenia opartą na algorytmie ewolucji różnicowej. W celu uniknięcia problemów z niedopuszczalnością chromosomów algorytm ewolucji różnicowej wykorzystuje nową reprezentację, w której macierze kowariancji są reprezentowane przy pomocy dekompozycji Cholesky'ego. W eksperymentach wykorzystano trzy wersje algorytmu ewolucji różnicowej różniące się

metodą doboru parametrów. Wyniki eksperymentów, przeprowadzonych na dwóch syntetycznych i jednym rzeczywistym zbiorze danych, wskazują, że zaproponowana metoda jest w stanie poprawnie identyfikować parametry modelu. Metoda ta osiąga również lepsze wyniki niż klasyczyny algorytm EM.

Słowa kluczowe: mieszaniny rozkładów normalnych, ewolucja różnicowa, algorytm EM

Artykuł zrealizowano w ramach pracy statutowej S/WI/2/08

APPLICATION OF DYNAMIC BAYESIAN NETWORKS TO RISK ASSESSMENT IN MEDICINE

Agnieszka Oniśko[1,2]

[1]Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland
[2]Magee Womens Hospital, University of Pittsburgh Medical Center, Pittsburgh, USA

Abstract: Dynamic Bayesian networks (DBNs) offer a framework for explicit modeling of temporal relationships, and are useful as both prognostic and diagnostic tools. In medicine, for example, they can assist in planning treatment options or in clinical management of patients. They have been also widely applied to genomics and proteomics.
This paper shows how dynamic Bayesian networks can be used in a risk assessment in medicine and presents an example of an application to cervical cancer screening. The model is a convenient tool for assessing the risk of cervical precancer and invasive cervical cancer over time. These quantitative risk assessments are helpful for establishing the optimal timing of follow-up screening and are the first step toward generating individualized reevaluation scheduling.

Keywords: dynamic Bayesian networks, risk assessment in medicine

1. Introduction

There is a variety of approaches to temporal modeling and reasoning in medicine (see [1] and [2] for accessible summaries). These include hidden Markov models, Markov decision processes, dynamic Bayesian networks, and dynamic influence diagrams. Markov models have been used widely in medical decision-analytic and cost-effectiveness models [25]. Ground breaking work based on dynamic models in medicine was performed by Leong, Harmanec, Xiang, and colleagues [12,16,27], who, in addition to Bayesian networks (BNs) and dynamic Bayesian networks (DBNs), used successfully a combination of graphical models with Markov chains to address different medical problems, including colorectal cancer management, neurosurgery ICU monitoring, and cleft lip and palate management. Several applications of dynamic Bayesian networks have been proposed in medicine. For

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 35-49, 2010.

example, NasoNet, a system for diagnosis and prognosis of nasopharyngeal cancer [10], or a DBN for management of patients suffering from a carcinoid tumor [26]. More recently, dynamic Bayesian networks have been used in genomics and proteomics, for example, in predicting protein secondary structure [28], modeling peptide fragmentation [15] and cellular systems [9], or in identifying gene regulatory networks from time course microarray data [29].

This paper shows how dynamic Bayesian networks can be applied to risk assessment in medicine. In addition to introducing the formalism to the readers, it describes a real model, based on a DBN, originating from author's work at the University of Pittsburgh [3,4,5]. This model illustrates general principles of building DBN models and applying them to risk assessment in medicine.

The reminder of this paper is structured as follows. Section 1. provides a brief review of work focusing on temporal modeling in medicine. Sections 2. and 3. present the formalism of Bayesian networks and their temporal extension, i.e., dynamic Bayesian networks. Section 4. captures several issues related to cervical cancer screening and describes an example of a risk model based on a dynamic Bayesian network. Sections 5. concludes the paper.

2. Bayesian Networks

Bayesian networks (BNs) [21], also called belief networks or causal networks, are acyclic directed graphs modeling probabilistic influences among variables. The graphical part of a Bayesian network reflects the structure of a modeled problem, while conditional probability distributions quantify local interactions among neighboring variables. Bayesian networks have proven to be powerful tools for modeling complex problems involving uncertain knowledge. They have been practically employed in a variety of fields, including engineering, science, and medicine with some models reaching the size of hundreds or thousands of variables.

Figure 1 captures a simple BN model. This example model includes four risk factors and one effect of breast cancer. Each arc of this graph represents a probabilistic relationship and it is quantified by a conditional probability distribution. For example, the arc between the variables *Family history* and *Breast Cancer* tells that family history of cancer impacts a risk of developing a breast cancer.

3. Dynamic Bayesian Networks

Dynamic Bayesian networks (DBNs) are a temporal extension of Bayesian networks that allows to model dynamic processes. The hidden Markov model [22] is considered

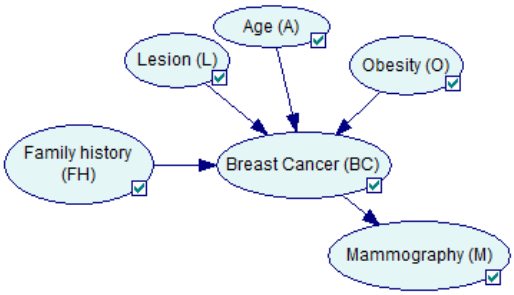

Fig. 1. Example of a BN model

to be the simplest dynamic Bayesian network. While Bayesian networks (BNs) have been used as modeling tools for over two decades, their temporal extension, dynamic Bayesian networks, found their way into medical modeling only in the last decade. Figure 2 captures an example of a dynamic Bayesian network model, an extension of the model presented in Figure 1. The graphical structure of the DBN model is similar to its static version, although there are additional arcs that quantify temporal relationships between neighboring variables.

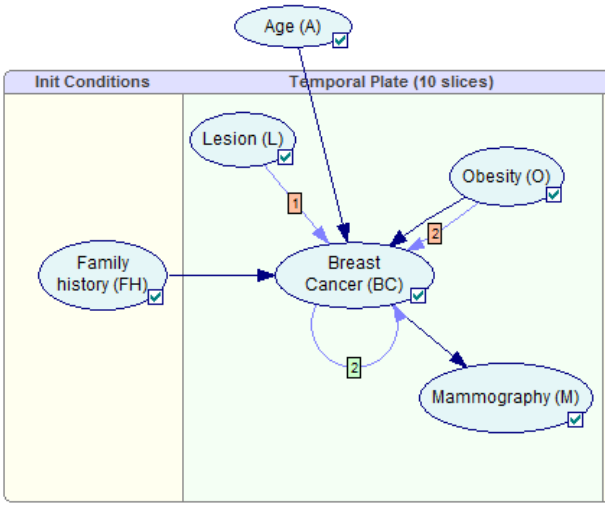

Fig. 2. Example of a DBN model

3.1 Temporal aspects of a DBN model

The dynamic arcs included in the example model, presented in Figure 2, represent changes over time among the variables. The single digit numbers on the arcs denote the temporal delay of influence. An arc labeled as *1* between the variables *Lesion (L)* and *Breast Cancer (BC)*, for example, denotes an influence that takes one time step, while an arc labeled as *2* between the variables *Obesity (O)* and *Breast cancer (BC)* denotes an influence that takes two time steps. Effectively, the model encodes the following conditional distribution over the variable *Breast Cancer (BC)*:

$$P(BC_t|A,FH,O_t,L_{t-1},O_{t-2},BC_{t-2}). \tag{1}$$

In other words, conditional probability distribution for *Breast Cancer (BC)* depends on a patient *Age (A)*, *Family history (FH)*, and a current status of the variable *Obesity (O)*. Furthermore, it depends on *Lesion (L)* result in previous time step and *Obesity* result recorded two time steps ago. Finally, it also depends on *Breast Cancer* result two time steps ago. The time step that is chosen for a dynamic Bayesian model varies on a modeled problem. In this example it could be a time interval used in screening for a breast cancer.

Age (A)	yes						
Obesity (O)	yes						
Lesion (L) [t-1]	present				absent		
Obesity (O) [t-2]	yes		no		yes		n
(Self) [t-2]	yes	no	yes	no	yes	no	yes
yes	0.1	0.08	0.02	0.01	0.1	0.08	0.02
no	0.9	0.92	0.98	0.99	0.9	0.92	0.98

Fig. 3. Fragment of a conditional probability table for the variable *Breast Cancer*

Since there are three types of arcs coming into the variable *Breast Cancer* (i.e., regular arcs representing static relationships between the variables and two types of temporal arcs with labels *1* and *2*), there are three different conditional probability tables that quantify the variable *Breast Cancer*. Equations 2, 3, and 4 correspond respectively to these three conditional probability tables (i.e., regular arcs: time step $t = 0$, temporal arcs labeled as *1*: time step $t = 1$, and temporal arcs labeled as *2*: time step $t = 2$):

$$P(BC_{t=0}|A,FH,O_{t=0}) \tag{2}$$

$$P(BC_{t=1}|A,O_{t=1},L_{t=0}) \tag{3}$$

$$P(BC_{t=2}|A, O_{t=2}, L_{t=1}, O_{t=0}, BC_{t=0}). \tag{4}$$

Figure 3 shows a fragment of the conditional probability table for the variable *Breast Cancer* for time step $t = 2$ (see also Equation 4). In this case a conditional probability distribution for *Breast Cancer* depends on the variables: *Age*, *Obesity*, and *Lesion* in previous time step $t = 1$. Furthermore, this conditional probability distribution depends on the variables *Obesity* and *Breast Cancer* in time step $t = 0$.

3.2 Unrolled DBN model

Figure 4 captures three unrolled time steps of the DBN model presented in Figure 2. Four out of six variables are repeated in each time step, i.e., *Lesion*, *Obesity*, *Breast Cancer*, *Mammography*. The variable *Family history* is not repeated since it was modeled only as an initial condition and it is not changing over time. Another variable that is not repeated is *Age*, although, it impacts the variable *Breast Cancer* in each time step.

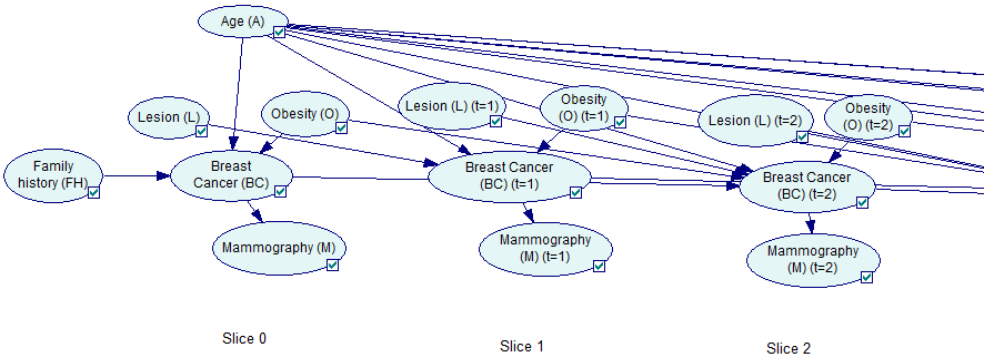

Fig. 4. Unrolled DBN model for the first 3 time steps

3.3 Dynamic evidence

Evidence can be observed for any time step implemented in the model. Figure 5 shows dynamic evidence for the variable *Mammography*. The model has 10 time steps (see Figure 2), therefore, there is a possibility of observing this variable for 10 different time steps. At time step *0* the result of the variable *Mammography* has been

observed normal, at time step *1* normal, there is no observation at time step *2* and an abnormal mammography was observed at time step *3*.

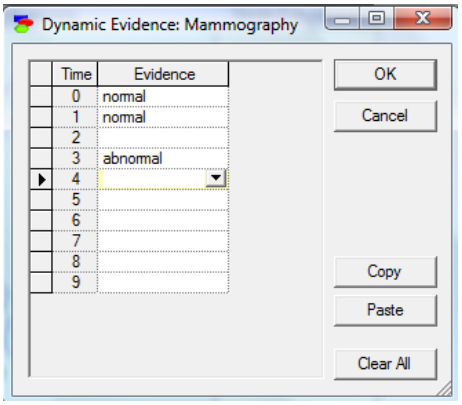

Fig. 5. Entering dynamic evidence for the variable *Mammography*

3.4 Risk assessment

Given observed dynamic evidence, the model can derive the probability distribution over a variable in question (in this case, the variable *Breast Cancer*). For example, the model will calculate the following probability:

$$P(BC(present)|E), \tag{5}$$

where

$$E = A(55), O_t(present), L_{t-1}(present), M_{t-1}(abnormal). \tag{6}$$

In this case, the model calculates a risk of developing a breast cancer for a 55 old, obese woman with a lesion and an abnormal mammography result in a previous time step. Figure 6 shows the probability of developing a breast cancer given this dynamic evidence, i.e., $P(BC(present)|E)$. This plot shows the risk of developing breast cancer over 10 time steps. It can be used to estimate the optimal time for follow-up medical tests and procedures.

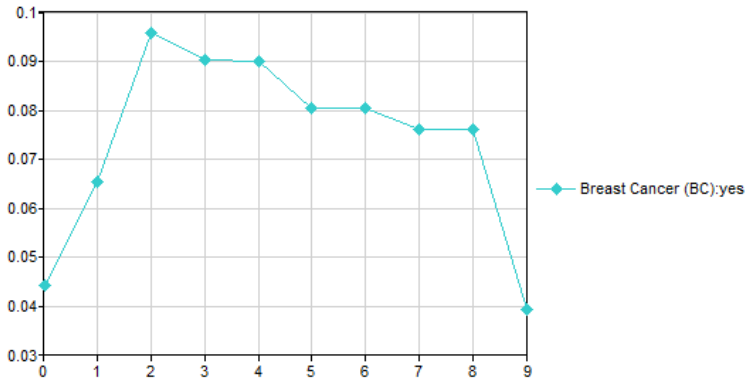

Fig. 6. Risk of a breast cancer over time

3.5 Challenges

The most challenging task in building a dynamic model are missing data, since often there is no complete follow-up of a patient case. A patient may show up for a test and then skip a year or never come back. There are several ways of dealing with this problem, one of which is representing missing values as an additional state [20]. Reasoning algorithms for Bayesian networks do not require complete information on a patient case. This means that the posterior probability distribution over a variable in question can be derived given any subset of possible observations.

4. Cervical Cancer Screening

DBNs are especially suitable for modeling screening data where there are temporal dependencies among variables. In this section, I will present an example of a medical problem, cervical cancer screening, in which DBNs have proven invaluable.

4.1 The problem of cervical cancer

Cervical cancer is the fifth most deadly cancer in women worldwide.[1] The introduction of the Papanicolaou test (also called PAP smear or PAP test) for cervical cancer screening has dramatically reduced the incidence and mortality of cervical

[1] World Health Organization, Fact sheet No. 297, Cancer, February 2006 (http://www.who.int/mediacentre/factsheets/fs297/en/index.html

cancer. Abnormal PAP test result suggests the presence of potentially premalignant or malignant changes in the cervix. PAP test allows for an examination and possible preventive treatment. Recommendations for how often a PAP test should be performed vary, depending on a screening program, between once a year and once every five years. The most important risk factor in the development of cervical cancer is infection with a high-risk strain of human papillomavirus (hrHPV). The virus works by triggering alterations in the cells of the cervix, which can lead further to the development of precancer, which can further result in cancer.

There have been several computer-based tools implemented to assist cervical cancer screening, diagnosis, and treatment decisions. These tools include computer-based systems to assist cytotechnologists and cytopathologists in the interpretation of PAP test slides. For example, an automated cervical precancerous diagnostic system extracts features from PAP test slides and then based on an artificial neural network predicts the cervical precancerous stage [17]. Another tool, developed a decade ago, is the PAPNET system [19]. The PAPNET system is also based on the neural network approach and assists rescreening of PAP test slides in order to identify cervical abnormalities that were not identified by a manual rescreening.

Cantor et al. [7] presented several decision-analytic and cost-effectiveness models that could be applied to guide cervical cancer screening, diagnosis, and treatment decisions. One of the decision-analytic models was a Markov model for the natural history of HPV infection and cervical carcinogenesis [18]. The model assesses life-time risk of cervical cancer as well as approximates the age-specific incidence of cervical cancer. Similar model was built for the German population [24]. The model was a Markov model for evaluating a life-time risk and life-time mortality of cervical cancer. Another group of tools for cervical cancer screening are cost-effectiveness models. Most of these cost-effectiveness models refer to investigation of an optimal scenario for cervical cancer screening based on two tests: PAP test and testing for the presence of hrHPV, e.g., [6,11,14].

There are many published studies that report risk assessments for cervical precancer and invasive cervical cancer, e.g., [8,13,23]. All these approaches have a major weakness, i.e., to my knowledge, all of these studies assess the risk based on the current state of a patient and do not include any history record. Many of these studies are based on cross-sectional data or on data coming from clinical trials. The strength of graphical models, such as DBNs is that they can easily combine information originating from history records and other sources.

4.2 The Pittsburgh Cervical Cancer Screening Model

The risk model presented in this paper is called *Pittsburgh Cervical Cancer Screening Model (*PCCSM). The model was built in Pittsburgh (Pennsylvania, USA) and the data that quantified it, reflect greater Pittsburgh population. The model is a dynamic Bayesian network that consists of 19 variables including cytological and histopathological data, and hrHPV test results. It also includes patient history data, such as history of infections, history of cancer, history of contraception, history of abnormal cytology, menstrual history, and demographics, i.e., age and race. One of the unique features of the PCCSM is the fact that risk assessments are generated not only based on a current state of a patient case, but also on a history record. Another advantage of the model is its sound quantification. All numerical parameters of the model were assessed based on a hospital data set coming from one population of patients. The model was parametrized by means of data collected during four years (2005-2008) and consisting of 393,531 patient records with PAP test result. The data were collected at Magee-Womens Hospital of the University of Pittsburgh Medical Center. More details on the model can be found in [5].

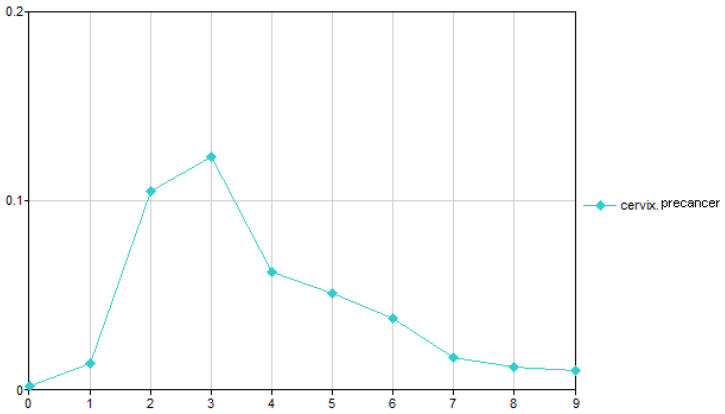

Fig. 7. Temporal beliefs

The PCCSM generates risk assessments for cervical precancer and invasive cervical cancer over time. Figure 7 captures quantitative risk assessments of precancer over the time period of 15 years for a single example patient case. It shows that this patient will run the highest risk of cervical precancer between the first and third year after the initial test. The dip in the third year is due to a delay in the effect of an

hrHPV virus infection. This risk will decrease after the fourth year. The reason for this shape of the curve were abnormal observations for *t=1* and *t=2* (abnormal PAP test results and positive hrHPV test results, respectively).

The PCCSM model allowed for identifying those risk categories that are crucial for follow-up planning, e.g., patients that are at higher risk for cervical cancer should be screened more often than patients that are at lower risk. Figure 8 presents risk assessments generated by the PCCSM model and stratified by the outputs of two variables: PAP and HPV tests. The chart captures average two years risk assessments for over 40,000 patient cases tested with the PCCSM model. It is evident from Figure 8 that a combination of *HSIL+* PAP test result with a positive HPV test result indicated the highest risk group for cervical precancer and cervical cancer. On the other hand a positive HPV test result does not by itself put a patient in a high risk group if the PAP test result is negative.

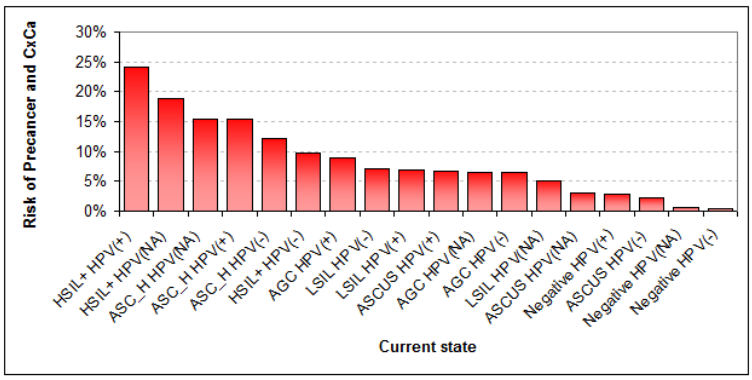

Fig. 8. PCCSM risk assessments for cervical precancer and cervical cancer (CxCa) stratified by the outputs of PAP and HPV test results

The PCCSM model allows for individualized management of patients and computes patient-specific risk based on the patients characteristics, history data, and test results. Figure 9 captures the PCCSM risk assessments given different patient history record. For example, a patient with all negative PAP test results in the past (last bar on the chart) is at different risk category than a patient having at least one *ASCUS* result (one of the abnormal PAP test results) in the past (the category *Any-ASCUS*). From the chart we can see that a risk assessment for the latter category doubles.

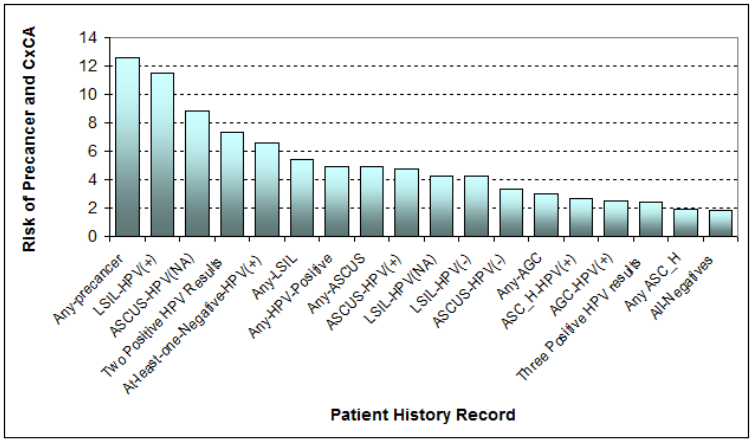

Fig. 9. PCCSM risk assessments for cervical precancer and cervical cancer (CxCa)

The PCCSM model is going to be used in Magee-Womens Hospital in the routine practice of identifying high risk patients. We are in the process of building a web-based graphical interface that will help to interact with the model.

5. Conclusions

Dynamic Bayesian networks are capable to model temporal relationships in medicine. They allow for computing quantitative risk assessments given observed variables. Dynamic Bayesian network models offer looking at risk assessments from different perspectives. They allow to identify groups of patients that are at higher risk of developing a disease. These models generate risk assessments over time. Furthermore, they quantify risk given patient history record. These quantitative risk assessments can be helpful in establishing the optimal timing of follow-up screening and can increase the accuracy of risk estimates. This can have a noticeable effect on the quality of medical care.

Acknowledgments

I would like to thank my collaborators at Magee Womens Hospital, University of Pittsburgh Medical Center: prof. R. Marshall Austin, for his expertise in cytopathology, his invaluable guidance, and his constant support and encouragement, Karen Lassige for her invaluable help in retrieving the data from the hospital database.

The models presented in this paper were created and tested using SMILE, an inference engine, and GeNIe, a development environment for reasoning in graphical probabilistic models, both developed at the Decision Systems Laboratory, University of Pittsburgh and available at http://genie.sis.pitt.edu/.

References

[1] Klaus-Peter Adlassnig, Carlo Combi, Amar K. Das, Elpida T. Keravnou, and Giuseppe Pozzi. Temporal representation and reasoning in medicine: Research directions and challenges. *Artificial Intelligence in Medicine*, 38:101–113, 2006.

[2] Juan Carlos Augusto. Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33:1–24, 2005.

[3] R. Marshall Austin, Agnieszka Oniśko, and Marek J. Druzdzel. Bayesian network model analysis as a quality control and risk assessment tool in cervical cancer screening. *Journal of Lower Genital Tract Disease*, 12:153–179, 2008.

[4] R. Marshall Austin, Agnieszka Oniśko, and Marek J. Druzdzel. The Pittsburgh Cervical Cancer Screening Model. *Cancer Cytopathology*, 114:345, 2008.

[5] R. Marshall Austin, Agnieszka Oniśko, and Marek J. Druzdzel. The Pittsburgh Cervical Cancer Screening Model. A Risk Assessment Tool. *Arch Pathol Lab Med*, 134:744–750, 2010.

[6] Michael A. Bidus, G. Larry Maxwell, Shalini Kulasingam, G. Scott Rose, John C. Elkas, Mildred Chernofsky, and Evan R. Myers. Cost-effectiveness analysis of liquid-based cytology and human papillomavirus testing in cervical cancer screening. *Obstetricians and Gynecologists*, 107(5):997–1005, 2006.

[7] Scott B. Cantor, Marianne C. Fahs, Jeanne S. Mandelblatt, Evan R. Myers, and Gillian D. Sanders. Decision science and cervical cancer. *Cancer*, 98(9):2003–2008, 2003.

[8] Philip E. Castle, Mario Sideri, Jose Jeronimo, Diane Solomon, and Mark Schiffman. Risk assessmenmt to guide the prevention of cervical cancer. *American Journal of Obstetrics and Gynecology*, 197:356.e1–356.e6, 2007.

[9] Fulvia Ferrazzi, Paola Sebastiani, Isaac S. Kohane, Marco Ramoni, and Riccardo Bellazzi. Dynamic Bayesian networks in modelling cellular systems: A critical appraisal on simulated data. In *Proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2006)*, pages 544–549, Salt Lake City, Utah, USA, 22-23 June 2006.

[10] S. F. Galan, F. Aguado, F. J. Díez, and J. Mira. NasoNet, modeling the spread of nasopharyngeal cancer with networks of probabilistic events in discrete time. *Artificial Intelligence in Medicine*, 25:247–264, 2002.

[11] Sue J. Goldie, Jane J. Kim, and Thomas C. Wright. Cost-effectiveness of human papillomavirus DNA testing for cervical cancer screening in women aged 30 years or more. *Obstetricians and Gynecologists*, 103(4):619–631, 2004.

[12] D. Harmanec, T. Y. Leong, S. Sundaresh, K. L. Poh, T. T. Yeo, I. Ng, and T. W. K Lew. Decision analytic approach to severe head injury management. In *Proceedings of the 1999 Annual Meeting of the American Medical Informatics Association (AMIA–99)*, pages 271–275, Washington, D.C., November 6–10 1999.

[13] Michelle J. Khan, Philip E. Castle, Attila T. Lorincz, Sholom Wacholder, Mark Sherman, David R. Scott, Brenda B. Rush, Andrew G. Glass, and Mark Schiffman. The elevated 10-year risk of cervical precancer and cancer in women with human papillomavirus (HPV) type 16 or 18 and the possible utility of type-specific hpv testing in clinical practice. *Journal of the National Cancer Institute*, 97(14):1072–79, 2005.

[14] Jane J. Kim, Thomas C. Wright, and Sue J. Goldie. Cost-effectiveness of alternative triage strategies for atypical squamous cells of undetermined significance. *JAMA*, 287:2382–90, 2002.

[15] Aaron A. Klammer, Sheila M. Reynolds, Jeff A. Bilmes, Michael J. MacCoss, and William Stafford Noble. Modeling peptide fragmentation with dynamic Bayesian networks for peptide identification. *Bioinformatics*, 24:i348–i356, 2008.

[16] Tze-Yun Leong. Multiple perspective dynamic decision making. *Artificial Intelligence*, 105:209–261, 1998.

[17] Nor Ashidi Mat-Isa, Mohd Yusoff Mashor, and Nor Hayati Othman. An automated cervical pre-cancerous diagnostic system. *Artificial Intelligence in Medicine*, 42:1–11, 2008.

[18] Evan R. Myers, Douglas C. McCrory, Kavita Nanda, Lori Bastian, and David B. Matchar. Mathematical model for the natural history of human papillomavirus infection and cervical carcinogenesis. *American Journal of Epidemiology*, 151:1158–1171, 2000.

[19] Timothy J. O'Leary, Miguel Tellado, Sally-Beth Buckner, Izzat S. Ali, Angelica Stevens, and Curtis W. Ollayos. PAPNET-assisted rescreening of cervical smears. Cost and accuracy compared with a 100 manual rescreening strategy. *JAMA*, 279(3):1–11, 1998.

[20] Agnieszka Oniśko, Marek J. Druzdzel, and Hanna Wasyluk. An experimental comparison of methods for handling incomplete data in learning parameters of Bayesian networks. In S.T. Wierzchoń M. Kłopotek, M. Michalewicz,

editor, *Intelligent Information Systems,* Advances in Soft Computing *Series*, Heidelberg, 2002. Physica-Verlag (A Springer-Verlag Company). 351–360.

[21] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[22] L. R. Rabiner. A tutorial in Hidden Markov Models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–289, 1989.

[23] Guglielmo Ronco, Nereo Segnan, Paolo Giorgi-Rossi, Marco Zappa, and et. al. Human papillomavirus testing and liquid-based cytology: Results at rucruitment from the new technologies for cervical cancer randomized controlled trial. *Journal of the National Cancer Institute*, 98(11):765–774, 2006.

[24] Uwe Siebert, Gaby Sroczynska, Peter Hillemanns, Jutta Engel, Roland Stabenow, Christa Stegmaier, Kerstin Voigt, Bernhard Gibis, Dieter Hölzel, and Sue J. Goldie. The German cervical cancer screening model: development and validation of a decision-analytic model for cervical cancer screening in Germany. *The European Journal of Public Health*, 16(2):185–192, 2006.

[25] Frank A. Sonnenberg and J. Robert Beck. Markov models in medical decision making: A practical guide. *Medical Decision Making*, 13:322–338, 1993.

[26] Marcel A. J. van Gerven, Babs G. Taal, and Peter J. F. Lucas. Dynamic Bayesian neworks as prognostic models for clinical patient management. *Journal of Biomedical Informatics*, 41:515–529, 2008.

[27] Yanping Xiang and Kim-Leng Poh. Time-critical dynamic decision modeling in medicine. *Computers in Biology and Medicine*, 32:85–97, 2002.

[28] Xin-Qiu Yao, Huaiqiu Zhu, and Zhen-Su She. A dynamic Bayesian network approach to protein secondary structure prediction. *BMC Bioinformatics*, 9:49–61, 2008.

[29] Min Zou and Suzanne D. Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–79, 2005.

ZASTOSOWANIE DYNAMICZNYCH SIECI BAYESOWSKICH W WYZNACZANIU RYZYKA W MEDYCYNIE

Streszczenie Dynamiczne sieci bayesowskie (DBNs) pozwalają na modelowanie zależności czasowych. Modele te są niejednokrotnie używane w prognostyce. Na przykład w medycynie, jako narzędzia do prognozowania czy też planowania terapii. Dynamiczne sieci

bayesowskie są też szeroko stosowane w genomice oraz w proteomice.
Atrykuł ten opisuje, w jaki sposób dynamiczne sieci bayesowskie mogą być zastosowane w wyznaczaniu ryzyka w medycynie. W pracy przedstawiono przykład zastosowania dynamicznych sieci bayesowskich w profilaktyce raka szyjki macicy. Prezentowany model został zbudowany w oparciu o dwa źródła wiedzy: opinie eksperta oraz dane medyczne. Model ten pozwala na wyznaczanie ryzyka zachorowania na raka szyjki macicy. Wartości ryzyka wyznaczane przez model pozwalają na określenie optymalnego czasu wykonania kolejnych badań przesiewowych oraz na zindywidualizowanie procesu profilaktyki.

Słowa kluczowe: dynamiczne sieci bayesowskie, wyznaczanie ryzyka w medycynie

GENETIC ALGORITHM FINDS ROUTES IN TRAVELLING SALESMAN PROBLEM WITH PROFITS

Anna Piwońska[1]

[1]Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Travelling salesman problem with profits is a version of a classic travelling salesman problem where it is not necessary to visit all vertices. Instead of it, with each vertex a number meaning a profit is associated. The problem is to find a cycle in a graph which maximizes collected profit but does not exceed a given cost constraint. This problem is NP-hard. Additional assumptions to this problem were proposed in the paper. We assumed that a graph may not be a complete graph. Moreover, repeated visiting of a given vertex is allowed, however with an assumption that a profit is realized only during first visiting. With these additional assumptions, the problem is more real-life and could have applications in logistics and shipping. To solve the problem, a genetic algorithm with special operators was proposed. The algorithm was tested on networks of cities in some voivodeships of Poland, obtaining very good results.

Keywords: travelling salesman problem with profits, genetic algorithm

1. Introduction

Travelling salesman problem (TSP) is a well known combinatorial optimization problem, studied in operational research and computer science. The problem is formulated as follows. Given the set of n cities and distances between each pair of them, find a closed tour (a cycle) through all cities that visits each city only once and is of minimum length. The problem is known to be NP-hard, therefore many heuristics have been proposed to find near-optimal solutions [9].

While in a classic TSP a salesman needs to visit all cities, some variant problems enforce to visit only selected ones, depending on a profit gained during visiting. This feature gives rise to a number of problems which are called in the literature travelling salesman problem with profits (TSPwP) [2,7]. In this group of problems,

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 51-65, 2010.

usually one of n cities has a special meaning - it is considered as a depot. In a one version of TSPwP described in the literature, the problem is to find an elementary cycle (i.e., a cycle such that each vertex is visited at most once) starting from a depot, that maximizes collected profit such that the tour length does not exceed a given constraint. This problem is also known in the literature under the name "the orienteering problem" (OP) [15] or "the selective TSP" [14] and will be considered in the paper. Like TSP, TSPwP belongs to the class of NP-hard problems. Due to high time complexity of the TSPwP, many metaheuristic approaches have been proposed in the literature, such as tabu search [5,13,3], ant colony optimization [10], genetic algorithms [1,7], neural networks [17] and harmony search [4].

In this paper, additional assumptions to this problem were proposed. Firstly, we assumed that a graph may not be a complete: not every pair of vertices must be connected by an edge. Looking at a map one can see that in the real world cities in some region are not all connected to each other. Despite of the fact that we can transform such a not complete graph in a complete one by introducing dummy edges, such an approach seems to be ineffective. It would result in a lot of unnecessary data introduced to the problem.

The second assumption is that we allow repeated visiting of a given vertex: a cycle we are looking for may not be an elementary one. This assumption results from the fact that a graph is not complete. However, while a salesman can be in a given city more than once, a profit is realized only during first visiting. This assumption prevents from finding routes in which a city with a highest profit is continually visited while others are not. With these additional assumptions, the problem is more real-life and could have applications in logistics and shipping. To find routes with optimal profit, a genetic algorithm (GA) with special operators was used. The method was implemented and tested on networks of cities in some voivodeships of Poland. Obtained results were encouraging.

The paper is organized as follows. Next section includes formal definition of TSPwP. Section 3 describes in details the GA. Experimental results are presented in Section 4. The paper ends with some remarks about future work.

2. Problem definition

A network of cities in our model is represented by a weighted, undirected graph $G = \langle V, E \rangle$. $V = \{1, 2, ..., n\}$ is a set of n vertices and E is a set of edges. Each node in G corresponds to a city in a network. Vertex 1 has a special meaning and is interpreted as the depot. An undirected edge $\{i, j\} \in E$ is an element of the set E and means that there is a possibility to travel from the city i to the city j (and vice versa). The

weight a_{ij} for an undirected edge $\{i, j\}$ denotes a distance between cities i and j. Additionally, with each vertex a non-negative number meaning a profit is associated. Let $P = \{p_1, p_2, ..., p_n\}$ be a vector of profits for all vertices. An important assumption is that a profit is realized only during first visiting of a given vertex.

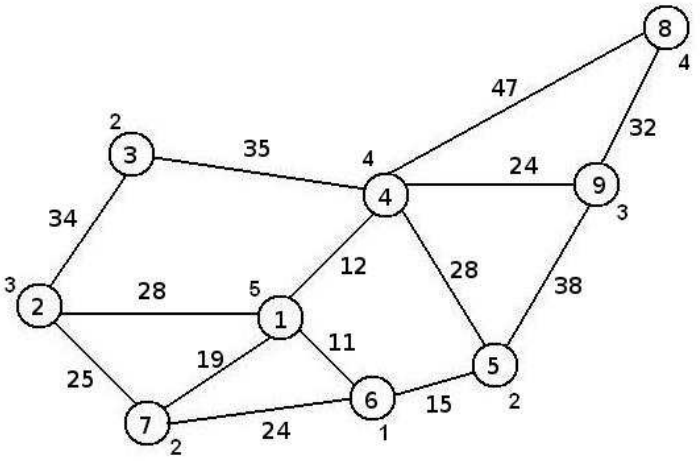

Fig. 1. A graph representation of a network of cities

A graph representation of an exemplary network of cities is shown in Fig. 1. It is a simple example of the network which includes nine cities. The a_{ij} values are marked on the edges and the p_i values are: {5, 3, 2, 4, 2, 1, 2, 4, 3} (marked beside vertices). One can see that the highest profit equals to 5 can be gained during visiting the depot.

The TSPwP can be formulated as follows. The goal is to find a cycle starting from the depot that maximizes collected profit such that the tour length does not exceed a given constraint c_{max}.

Assuming $c_{max} = 100$, for the graph presented in Fig. 1, one possible solution could be: 1 - 4 - 9 - 5 - 6 - 1. In this case the tour length equals to 100 and the collected profit equals to 15.

3. GA for discovering routes

GAs are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and heredity. First introduced by John Holland in the 60s, GAs has been widely studied, experimented and successfully applied in many fields [6,11].

In a typical GA, a population of candidate solutions (called chromosomes) is evolved during artificial evolution. Traditionally, solutions are represented as binary strings, but other encodings are also possible. The GA starts from a population of randomly generated individuals. In each generation, the fitness of every individual is evaluated. Based on the fitness, individuals are stochastically selected from the current population to the next one. This step is called a selection. Individuals in the new population undergo genetic operators: crossover and mutation. The new population is then used in the next iteration of the algorithm. The algorithm usually terminates when a maximum number of generations *ng* has been reached [12].

The pseudocode of the GA described in this paper is presented below.

```
  Genetic algorithm
  Begin
    generate an initial population of individuals of size P;
    compute fitness function for each individual;
    for i:=1 to ng do
      Begin
        select the population i from the population i-1 by means of tournament
selection
        with the group size equals to t\_size;
        divide population into disjont pairs;
        cross each pair if possible;
        mutate each individual if possible;
      End;
    choose the best individual from the final population as the result;
  End;
```

3.1 Genetic reprezentation and initial popualtion generating

The first step in the GA is encoding a solution into a chromosome. We use the path representation which is the most natural for this problem [11]. In this approach, a tour is encoded as a sequence of vertices. For example, the tour 1 - 4 - 5 - 6 - 1 is represented by the sequence 1 - 4 - 5 - 6 - 1, as was described in the Section 2.

The GA starts with a population of *P* solutions of TSPwP. The initial population is generated in a special way. Starting at the depot, with equal probability we choose

a city to which we can travel from the depot. We add the distance between the depot and the chosen city to the current tour length. If the current tour length is not greater than $c_{max}/2$, we continue, but instead of starting at the depot, we start at the chosen city. We again randomly select a city, but this time we exclude from the set of possible cities the city from which we have just arrived (the last city in a partial tour). This assumption prevents from continual visiting a given city but is relaxed if there is no possibility to choose another city.

If the current tour length is greater than $c_{max}/2$, we reject the last city and return to the depot the same way. In this case the tour length does not exceed c_{max} therefore the constraint imposed by the problem is preserved. One can see that such an idea of generating the initial population causes that individuals are symmetrical in respect of the middle city in the tour. However, experiments show that the GA quickly breaks these symmetries.

Let us construct an individual for the problem presented in Fig. 1 with the assumption that $c_{max} = 150$. We start at the node 1 and have to choose one node from the set {2, 4, 6, 7}. Let us assume that node 2 was selected. Since the distance between 1 and 2 equals to 28, the current tour length equals to 28 (and is not greater then $c_{max}/2$). The partial tour is 1 - 2. Starting at the node 2, we can select the node 3 or the node 7, with equal probability (we exclude node 1). Let us assume that the node 3 was selected. The current tour is now 1 - 2 - 3 with the length equal to 62. Starting from the node 3 we can only select the node 4 but this situation will cause crossing the treshold value $c_{max}/2$. We must reject the node 4 and return to the depot the same way. Our complete tour is 1 - 2 - 3 - 2 - 1 and has the length equal to 124.

3.2 Fitness computing

The next step is to evaluate individuals in the initial population by means of the fitness function. The fitness of a given individual is equal to collected profit under the assumption that a profit is realized only during first visiting of a given vertex. For example, the fitness of the individual represented by the chromosome: 1 - 2 - 3 - 2 - 1 equals to 10.

3.3 Selection

Once we have the fitness function computed, the GA starts to improve the initial population through repetitive application of selection, crossover and mutation. In our experiments we use tournament selection: we select t_{size} individuals from the current population and determine the best one from the group. The winner is copied to the

next population and the whole tournament group is returned to the old population (randomizing with returns). This step is repeated P times. The parameter t_{size} should be carefully set because the higher t_{size}, the faster convergence of the GA.

3.4 Crossover

We present a new heuristic crossover operator adjusted to our problem. In the first step, individuals are randomly coupled. Then, each couple is tested if crossover can take place. If two parents do not have at least one common gene (with the exception of the depot), crossover can not be done and parents remain unchanged. Crossover is implemented in the following way. First we randomly choose one common gene from the set of common genes in both parents (we exclude the depot from this set). This gene will be the crossing point. Then we exchange fragments of tours from the crossing point to the end of the chromosome in two parent individuals. If offspring individuals preserve the constraint c_{max}, they replace in the new population their parents. If one offspring individual does not preserve the constraint c_{max}, its position in the new population is occupied by better (more fitter) parent. If both children do not preserve the constraint c_{max}, they are replaced by their parents in the new population. The example of the crossover is presented in Fig. 2. This example concerns Fig. 1 with the assumption that $c_{max} = 200$.

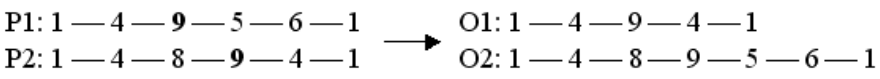

Fig. 2. The example of the crossover operator

The length of the tours represented by offsppring are equal to 72 and 155, respectively. Since both offspring individuals preserve the constraint c_{max}, they replace in the new population their parents.

3.5 Mutation

The last genetic operator is a mutation. Each individual in the current population undergo mutation. It is performed in the following way. First we randomly select a position in a chromosome where a mutation will be performed. Then we try to insert a city (from the set of possible cities) at this position. If inserting a city do not violate the constraint c_{max}, we keep this new city in a tour otherwise it is rejected.

For example, let us look at the individual O2 in Fig. 3. Let us assume that this individual is to be muated and randomly selected position in the chromosome is marked with an arrow. The only city we can insert between the cities 6 and 1 is the city 7. Inserting this city will result in the tour length equal to 198. Since $c_{max} = 200$, we keep the city 7 on its position. The new mutated individual O2' replaces O2 in the population.

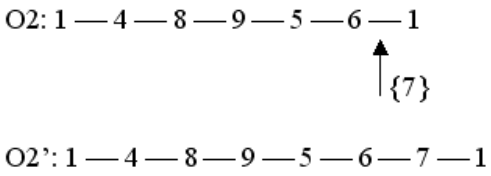

Fig. 3. The example of the mutation operator

4. Experimental results

We conducted experiments on networks of cities in the following voivodeships of Poland: podlaskie, mazowieckie and warminsko-mazurskie. From each voivodeship, twenty cities were selected. A network was created from a real map, by including to a graph main segments of roads. Profits associated with a given city were determined according to a number of inhabitants in a given city. The more inhabitants, the higher profit associated with a given city. These rules are presented in a Tab. 1.

Table 1. Profits associated with a given city

number of inhabitants	profit
<= 10000	1
(10000, 20000>	2
(20000, 30000>	3
(30000, 40000>	4
> 40000	5

The data for each voivodeship (profits and distances between cities) are presented in Appendix.

The parameters for the GA were determined through series of experiments described in [8]. Based on results of those experiments we set $P = 200$, $t_{size} = 3$ and $ng = 100$. Tests were performed for three c_{max} values: 300, 500 and 700. Results of experiments are presented in later subsections.

4.1 Voivodeship podlaskie

The best results from ten GA runs are presented in Tab. 2 and 3.

Table 2. The best results for voivodeship podlaskie

	$c_{max} = 300$	$c_{max} = 500$	$c_{max} = 700$
profit	25	37	46
distance	298	498	697

Table 3. Chromosomes of the best individuals for voivodeship podlaskie

	chromosome
$c_{max} = 300$	[1,9,6,19,8,2,14,13,17,15,1]
$c_{max} = 500$	[1,7,5,17,15,13,14,2,8,19,9,6,4,12,1]
$c_{max} = 700$	[1,7,18,11,5,17,15,13,14,2,19,8,19,6,9,6,4,3,4,12,20,1]

4.2 Voivodeship mazowieckie

The best results from ten GA runs are presented in Tab. 4 and 5.

Table 4. The best results for voivodeship mazowieckie

	$c_{max} = 300$	$c_{max} = 500$	$c_{max} = 700$
profit	23	41	50
distance	291	494	700

4.3 Voivodeship warminsko-mazurskie

The best results from ten GA runs are presented in Tab. 6 and 7.

Table 5. Chromosomes of the best individuals for voivodeship mazowieckie

	chromosome
$c_{max} = 300$	[1,10,5,12,6,11,15,1]
$c_{max} = 500$	[1,11,6,12,5,2,3,4,9,14,13,16,1]
$c_{max} = 700$	[1,13,14,9,8,7,8,4,3,2,5,6,12,6,11,15,16,1,10,1]

Table 6. The best results for voivodeship warminsko-mazurskie

	$c_{max} = 300$	$c_{max} = 500$	$c_{max} = 700$
profit	48	57	59
distance	295	500	700

Let us look closely at the results obtained for voivodeship warminsko-mazurskie. Fig. 4 presents the best of ten GA runs for each c_{max}. On each plot we can see the profit of the best individual in a given generation. For all plots presented in this figure, the GA quickly (before 15th generation) finds the optimal (or suboptimal) solutions. Further generations do not bring any improvement.

For $c_{max} = 500$, the profit of the best individual equals to 57. The chromosome of this individual consists of 25 cities and has the tour length equal to 500. However, for $c_{max} = 700$, the profit of the best individual is only 2 greater than for $c_{max} = 500$ and the chromosome length equals to 44 genes (the tour length equals to 700). This problem is explained below.

For voivodeship warminsko-mazurskie, the maximal profit which can be gained equals to 59 for a tour length equal to 435. A solution with the best possible profit is found quickly - in 11th generation. However, the length of the solution (with the maximal profit) from the final generation equals to 700 (Tab. 6). The main reason is the mutation operator. In spite of the fact that a current solution can not be improved (because it is the global maximum), the mutation causes inserting to the current tour cities which increase the total tour length but can not increase the collected profit. This situation is clear when looking at the chromosome of the best individual for $c_{max} = 700$ from Tab. 7. One can see that some citities e.g. 17, 14, 19 repeat in the

Table 7. Chromosomes of the best individuals for voivodeship warminsko-mazurskie

	chromosome
$c_{max} = 300$	[1,5,17,19,14,9,7,10,6,20,6,3,12,8,16,11,4,5,1]
$c_{max} = 500$	[1,16,11,4,5,17,14,9,13,2,13,9,7,10,8,12,3,15,18,20,6,10,7,10,1]
$c_{max} = 700$	[1,5,17,19,17,19,14,17,19,17,14,17,19,17,19,14,19,17,14,17,14,19, 2,13,9,7,10,6,20,18,15,3,12,8,16,11,4,11,16,11,16,11,16,1]

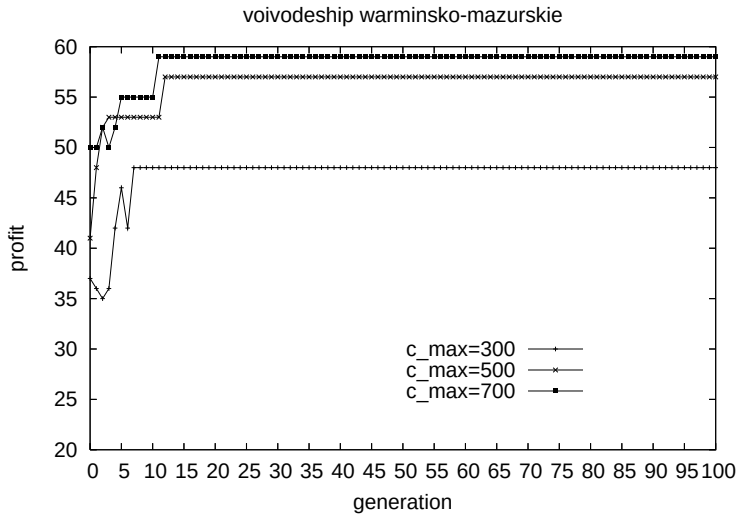

Fig. 4. The GA run for voivodeship warminsko-mazurskie

chromosome but they do not influence fitness value and can be removed from the chromosome. This way of action of the mutation needs to be improved. The new cities should be inserted to the tour only if they cause an improvement of the fitness function, otherwise the tour length is unnecessary incremented.

To verify this hypothesis, we implemented improved version of mutation operator. In this case, a new city is inserted to the tour only if it is not present in a tour yet. Otherwise, mutation is not performed. The new mutation was tested on the same data as the ordinary mutation. Differences were most evidently observed for $c_{max} = 700$ and are presented in Tab. 8.

Table 8. Results of the improved mutation, $c_{max} = 700$

voivodeship	chromosome	profit	distance
podlaskie	[1,7,18,5,11,16,17,15,13,14,2,8,19,6,4,3,10,4,12,20,1]	46	677
mazowieckie	[1,11,6,10,5,2,3,4,7,8,9,14,13,16,18,19,17,15,1]	53	691
warminsko-mazurskie	[1,5,4,11,16,8,12,3,15,18,20,6,10,7,9,13,2,19,14,17,5,1]	59	435

One can see that results obtained with the improved mutation are better in the context of collected profit (mazowieckie) and also in the context of total distance (podlaskie, mazowieckie, warminsko-mazurskie). The problem of repeated cities was

majorly eliminated. Of course, repeated cities may again appear in a solution due to crossover operator.

5. Conclusions

In this paper we presented a version of TSPwP. Additional assumptions to this problem were proposed in the paper which make the problem more real-life. The aim of the work was designing a GA to deal with this problem. The GA proposed in the paper was tested on some voivodeships in Poland, obatining satisfactory results. However, these results should be compared with results obtained by the other heuristic algorithms.

Another issue which must be carefully studied is the muation operator. Results of experiments have shown that this operator has significant role in the quality of obtained solutions. Two versions of muation inserting a city to a tour were implemented and tested. However, another kind of mutation can also be considered, for example inserting a fragment of tour or exchanging cities in a tour.

The results presented in this paper are results of preliminary experiments. The future work will be focused on testing the improved version of the GA on bigger and more dense networks, for example for cities from the whole Poland.

6. Appendix

The data for voivodeship podlaskie are presented below.

Voivodeship podlaskie
1 (Bialystok 5) **2** 80 **4** 90 **5** 49 **7** 39 **9** 42 **12** 40 **14** 68 **15** 28 **20** 47
2 (Lomza 5) **1** 80 **8** 32 **14** 24 **19** 25
3 (Suwalki 5) **4** 34 **10** 31
4 (Augustow 4) **1** 90 **3** 34 **6** 42 **10** 43 **12** 76
5 (Bielsk Podlaski 3) **1** 49 **7** 28 **11** 30 **17** 25 **18** 25
6 (Grajewo 3) **4** 42 **9** 37 **19** 38
7 (Hajnowka 3) **1** 44 **5** 28 **18** 27
8 (Kolno 2) **2** 32 **19** 17
9 (Monki 2) **1** 42 **6** 37 **19** 54
10 (Sejny 1) **3** 31 **4** 43
11 (Siemiatycze 2) **5** 30 **16** 38 **18** 38
12 (Sokolka 2) **1** 40 **4** 76 **20** 26
13 (Wysokie Mazowieckie 1) **14** 21 **15** 31 **17** 27

14 (Zambrow 3) **1** 68 **2** 24 **13** 21 **16** 43
15 (Lapy 2) **1** 28 **13** 31 **17** 32
16 (Ciechanowiec 1) **11** 38 **14** 43 **17** 25
17 (Bransk 1) **5** 25 **13** 27 **15** 32 **16** 25
18 (Kleszczele 1) **5** 24 **7** 27 **11** 38
19 (Stawiski 1) **2** 25 **6** 38 **8** 17 **9** 54
20 (Krynki 1) **1** 47 **12** 26

Each city has an unique number. The capital of the voivodeship has the number 1 (this rule is also applied to other voivodeships). Profits associated with cities are given in the brackets. In each line the distances between a given city and other cities are presented (in pairs: a number of a city and a distance). For example let us look at the first line. One can see that the distance between Bialystok and Lomza (number 2) equals to 80, the distance between Bialystok and Augustow (number 4) equals to 90 etc.

The data for voivodeship mazowieckie are presented below.

Voivodeship mazowieckie
1 (Warszawa 5) **3** 81 **7** 58 **10** 37 **11** 54 **13** 40 **16** 34 **15** 46
2 (Ciechanow 5) **12** 82 **5** 36 **3** 37
3 (Makow Mazowiecki 2) **2** 37 **1** 81 **4** 56
4 (Ostrow Mazowiecka 3) **3** 56 **7** 42 **8** 37 **9** 55
5 (Plonsk 3) **12** 49 **2** 36 **6** 31 **10** 33
6 (Wyszogrod 1) **12** 38 **11** 24 **5** 31 **10** 37
7 (Wyszkow 3) **4** 42 **8** 18 **1** 58
8 (Lochow 1) **9** 46 **13** 43 **7** 18 **4** 37
9 (Sokolow Podlaski 2) **4** 55 **8** 46 **14** 30
10 (Nowy Dwor Mazowiecki 3) **5** 33 **6** 37 **1** 37
11 (Sochaczew 4) **6** 24 **1** 54 **15** 64
12 (Plock 5) **2** 82 **6** 38 **5** 49
13 (Minsk Mazowiecki 4) **1** 40 **8** 43 **14** 50 **16** 41
14 (Siedlce 5) **13** 40 **9** 30
15 (Grojec 2) **11** 64 **1** 46 **16** 29 **17** 27
16 (Gora Kalwaria 2) **1** 34 **15** 29 **13** 41 **18** 52
17 (Bialobrzegi 1) **15** 27 **18** 50 **19** 35
18 (Kozienice 2) **16** 52 **17** 50 **19** 37 **20** 28
19 (Radom 5) **17** 35 **18** 37 **20** 33

20 (Zwolen 1) **19** 33 **18** 28

The data for voivodeship warminsko-mazurskie are presented below.

Voivodeship warminsko-mazurskie
1 (Olsztyn 5) **5** 20 **14** 40 **10** 35 **16** 10
2 (Elblag 5) **13** 40 **19** 50
3 (Elk 5) **6** 20 **15** 10 **12** 20
4 (Ilawa 4) **5** 30 **11** 35
5 (Ostroda 4) **4** 30 **19** 15 **17** 10 **1** 20
6 (Gizycko 3) **10** 10 **20** 5 **3** 20
7 (Ketrzyn 3) **9** 20 **10** 5
8 (Szczytno 3) **16** 15 **10** 25 **12** 30
9 (Bartoszyce 3) **13** 50 **7** 20 **14** 10
10 (Mragowo 3) **7** 5 **6** 10 **12** 15 **8** 25 **1** 35
11 (Dzialdowo 3) **4** 35 **16** 20
12 (Pisz 2) **10** 15 **3** 20 **8** 30
13 (Braniewo 2) **2** 40 **9** 50
14 (Lidzbark Warminski 2) **9** 10 **1** 40 **17** 10 **19** 10
15 (Olecko 2) **18** 5 **3** 10
16 (Nidzica 2) **1** 10 **8** 15 **11** 20
17 (Morag 2) **19** 10 **14** 10 **5** 10
18 (Goldap 2) **20** 20 **15** 5
19 (Paslek 2) **2** 50 **14** 10 **17** 10 **5** 15
20 (Wegorzewo 2) **6** 5 **18** 20

References

[1] Fatih Tasgetiren, M.: A Genetic Algorithm with an Adaptive Penalty Function for the Orienteering Problem. Journal of Economic and Social Research, vol. 4 (2), pp. 1-26, 2002.

[2] Feillet, D., Dejax, P., Gendreau, M.: Traveling Salesman Problems with Profits, Transportation Science, vol. 39 (2), pp. 188-205, 2005.

[3] Fischetti, M., Salazar González, J. J., Toth, P.: Solving the orienteering problem through branch-and-cut. INFORMS Journal on Computing, vol. 10 (2), pp. 133-148, 1998.

[4] Geem, Z. W., Tseng, Ch.-L., Park, Y.: Harmony Search for Generalized Orienteering Problem: Best Touring in China. LNCS, vol. 3612, Springer, pp. 741-750, 2005.
[5] Gendreau, M., Laporte, G., Semet, F.: A tabu search heuristic for the undirected selective travelling salesman problem. European Journal of Operational Research, vol. 106 (2-3), Elsevier, pp. 539-545, 1998.
[6] Goldberg, D. E.: Genetic algorithms and their applications. WNT, Warsaw, 1995.
[7] Jozefowiez, N., Glover F., Laguna, M.: Multi-objective Meta-heuristics for the Traveling Salesman Problem with Profits. Journal of Mathematical Modelling and Algorithms, vol. 7 (2), pp. 177-195, 2008.
[8] Koszelew, J., Piwonska, A.: Tunning Parameters of Evolutionary Algorithm in Travelling Salesman Problem with Profits and Returns. Archives of Transport System Telematics, to appear (2010).
[9] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley & Sons, 1985.
[10] Liang, Y.-C., Smith, A. E.: An ant colony approach to the orienteering problem. Technical report. Department of Industrial and Systems Engineering, Auburn University, Auburn, USA, 2001.
[11] Michalewicz, Z.: Genetic Algorithms+Data Structures=Evolution Programs. WNT, Warsaw, 1996.
[12] Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, 1998.
[13] Ramesh, R., Brown, K. M.: An efficient four-phase heuristic for the generalized orienteering problem. Computers & Operations Research, vol. 18 (2), Elsevier, pp. 151-165, 1991.
[14] Qin, H., Lim, A., Xu, D.: The Selective Traveling Salesman Problem with Regular Working Time Windows. Studies in Computational Intelligence, vol. 214, pp. 291-296, 2009.
[15] Sevkli, Z., Sevilgen, F. E.: Variable Neighborhood Search for the Orienteering Problem. LNCS, vol. 4263, Springer, pp. 134-143, 2006.
[16] Souffriau, W., Vansteenwegen, P., Berghe, G. V., Oudheusden, D. V.: A Greedy Randomised Adaptive Search Procedure for the Team Orienteering Problem. In Proceedings of EU/MEeting 2008 - Troyes, France, 2008.
[17] Wang, Q., Sun, X., Golden, B. L., Jia, J.: Using artificial neural networks to solve the orienteering problem. Annals of Operations Research, vol. 61, Springer, pp. 111-120, 1995.

ALGORYTM GENETYCZNY ODNAJDUJE TRASY W PROBLEMIE KOMIWOJAŻERA Z ZYSKAMI

Streszczenie Problem komiwojażera z zyskami (ang. TSP with profits) jest pewną wersją klasycznego problemu komiwojażera, w której nie jest konieczne odwiedzenie wszystkich wierzchołków grafu. Zamiast tego, z każdym wierzchołkiem związana jest pewna liczba oznaczająca zysk. Problem polega na znalezieniu cyklu w grafie, który maksymalizuje zysk, ale którego koszt nie przekracza zadanego ograniczenia. Problem ten jest problemem NP-trudnym. Do tak postawionego problemu, w pracy zaproponowano dodatkowe założenia. Przyjęto mianowicie, że graf nie musi być pełny. Ponadto dopuszczona jest możliwość powrotów, czyli ponownego odwiedzenia danego wierzchołka, przy założeniu jednak, iż zysk realizowany jest tylko podczas pierwszego odwiedzenia. Przy tych dodatkowych założeniach problem jest bardziej realny i może mieć konkretne zastosowania w logistyce i spedycji. Do rozwiązania problemu zaproponowano algorytm genetyczny, uzyskując bardzo dobre wyniki.

Słowa kluczowe: problem komiwojażera z zyskami, algorytm genetyczny

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/2008

USER ACTIVITY DETECTION IN COMPUTER SYSTEMS BY MEANS OF RECURRENCE PLOT ANALYSIS

Tomasz Rybak[1], Romuald Mosdorf[1,2]

[1]Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland
[2]Faculty of Mechanical Engineering, Bialystok University of Technology, Białystok, Poland

Abstract: As computers are getting faster and disks are growing bigger more data describing user behaviour can be gathered. These data can be analysed to gain insight into user behaviour and then to detect user traits. Currently many different methods are used to analyse data — and there is still no one best method for analysing different parameters of computer systems. Computer systems behave non-linearly because they run many programs on multi-user operating systems; this causes inter-program dependencies requiring non-linear methods to analyse gathered data.
The aim of the article is to present how non-linear methods are able to detect subtle changes introduced into system by user's actions. Number of interrupts per second was chosen as variable describing system's behaviour. Analysis presented in this article focuses on idle system and system busy accessing hardware. Article shows that using recurrence plot can reveal similarities in behaviour of the system running different programs, and therefore can be used to detect similarities and differences in users behaviour.
This article presents analysis of system activity through usage of series of recurrence plots to detect changes introduced by user actions. Analysis of lengths of horizontal and vertical lines on recurrence plots allows for describing periodicity of the system. This allows for gaining insight into behaviour of entire computing environment. Article shows that different tasks (such as network transmission, writing or reading from CD-ROM, compressing data) result in different recurrence plots; at the same time changes introduced by those tasks are hard to detect without analysis data. This means that usage of recurrence plot is crucial in detecting changes introduced in system by user's actions.

Keywords: user behaviour analysis, fractal analysis, recurrence plot

1. Introduction

Computers are typically driven by their users. Even if a computer is running programs that were not chosen directly by user, this is done to aid user: anti-virus program

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 67-85, 2010.

protects against harmful software, firewall reduces risk of attacks from the network, indexing system scans all documents to allow for faster finding interesting phrases in the files. Programs mentioned above run all the time, so their characteristics can be treated as background noise while analysing the system's behaviour. Most of the characteristics of a running system come from programs that were directly run by the user. Some users tend to run many programs at the same time and switch between them; others perform tasks inside one application, close it when job inside this program is done, and only then start a new program. Some noise can be introduced by the fact that some programs use other programs — e.g. text processor can call spreadsheet or graphics program to compute results and generate graphs. Even in the case of a single user, her behaviour can depend on the mood, eagerness to work, degree of tiredness, time of day, etc. Therefore, the view of entire system, and change of it over time, can be used to characterise behaviour of the user.

As noted by Rolia et al. [7], knowing state of system, their capacities, possible bottlenecks, and current load allows for reconfiguration to avoid unnecessary overloading of some machines by routing load to other ones. Porter and Katz [5] are measuring details of behaviour of systems to have roughly the same load on all servers so none is over- or under-utilised. Matthew Garret [1] notes that to be able to reduce power consumption, and thus ecological footprint of computers, we need to know details of behaviour of individual programs and entire system. Most common ones are: how many operations involving disk are executed in one second (pointing whether the disk can slow down), how often CPU needs to check state of peripheral devices (disallowing switching to the lower power consumption modes), and whether any operations can be grouped together so there are longer gaps between operations allowing for disabling of some hardware.

Hoffman [2] and Rogers [6] describe monitoring of server farms serving Hotmail mail and Microsoft pages respectively. They note that to be able to manage large groups of machines and to be able to detect anomalies one needs to gather details of their work. On the other hand, no human can cope with such amounts of data so one needs to use statistics to detect trends and base decisions on those aggregate results. Hoffman also notes that it is impossible and pointless to try to come with artificial test cases when managing large groups of machines. Generating large tests requires large amounts of work and thinking about many different scenarios. To be able to execute test cases one also needs need fair amount of servers — which would be used solely for testing and not for usage by users. The biggest disadvantage of using dedicated testing machines would be inability to come with all of scenarios that users can generate. There is too many users, possible software configurations, and so on. He claims that it is better to just observe and analyse behaviour of real, life system. This

requires ability of accessing necessary data and very fast responses to any problems. The most important in such a case is disallowing for analysing software not to cause performance cost on the systems it is running on.

Oskin [4] claims that increasing number of cores in CPU forces programmers to analyse data describing execution details so he is able to check if multi-threaded programs behave correctly and to correctly manage large number of virtual processors. George Neville-Neil [3] observers that current operating systems provide programmer with access to many internal hardware counters that can show how well program is executing and if its performance suffers on particular hardware platform. Shaw et al. [9] use specialised hardware with good amount of monitoring to make sure that massive parallelism is well used and no chip is using power without doing useful work. Counters in such systems include cache hits and misses, number of context switches, number of correctly and incorrectly predicted branches, page faults, etc. Although they are most useful for operating system programmers, they point why computer is behaving slowly and thus are valuable resource of information about internals of computer system behaviour. But again they generate vast amounts of data which is very hard to analyse by human being.

As can be seen from cited literature there is still no consensus over which methods are the best and which ones in the best way represents behaviour of programs, especially when different aspect of behaviour are analysed. This article presents usage of non-linear methods to detect characteristics of user's behaviour. Analysed activities were using hardware present in the computer system: transmission over network, burning files to CD and reading files from CD, so number of hardware interrupts per second was used as the best descriptor of the system behaviour. Interrupt usually occurs as result of hardware event, like keyboard or mouse activity, disk or network transmission, comes from hardware clock, etc. The more interrupts, the more activity comes from hardware, which means that programs are intensively communicating with the outside environment.

Because of inter-program dependencies we assume that computer systems are non-linear dynamic systems. The aim of the article is to present how non-linear methods are able to detect subtle changes introduced into computer system by user actions. The remainder of the article is structured as follows. The next section describes procedure used to collect data, including hardware and software configuration. Section 3. describes theory of mathematical means used to analyse gathered data. Section 4. describes and analyses obtained results and their meaning. The two last sections summarize paper and present possible future research.

2. Methodology of data collection

Data was gathered on the AMD Duron 1.3GHz with 768MB or RAM and single IDE 7200RPM hard drive. Debian Linux system (version named Sid) was used as operating system for this computer. System had 1GB of active swap partition; kernel was 32-bit 2.6.26 with Debian patches. System was not upgraded (neither manually nor automatically) during entire course of experiment to avoid changes in the environment that could influence process of gathering data or the data itself.

System had public IP address, so it was possible to connect to it from the Internet. Network connection was not disabled, because doing so would not resemble normal mode of operation that would be later examined. Some run-levels, however, have not started any servers listening on the network which allowed for comparison of situations with and without public services available. We claim that having active network card does not invalidate experiment results.

Data was gathered by running the computer on different run-levels (set of running system processes) on consecutive days. To collect numerical data vmstat program was used. Detailed description of characteristics of Unix-like operating systems, as well as details of process of collecting data can be found in our previous paper [8].

As mentioned in Introduction, number of interrupts per second was chosen as a measure of system's activity. Data was gathered using vmstat program once a second for about 90 minutes, depending on the observed run-level. Data from all possible run-levels was gathered. We decided to use four cases: one from a system in which only a bare necessity of programs needed to run the operating system is run (Single mode, raw data in Figure 3 a)), one from a system in which all programs except graphical environment are running (level 2, raw data on Figure 4 a)) one from a system with active graphical environment (level 5, raw data on Figure 5 a)) and finally from a system in which user is logged in, and he was transferring data over network, copying it, burning to CD drive and reading data from CD (raw data on Figure 6 a)).

3. Non-linear signal analysis

Recurrence plot was chosen to analyse the gathered data describing characteristics of the Linux system.

Fractal analysis of an one-dimensional signal assumes that all important dynamic variables present in system influence these time series. To perform non-linear (fractal) analysis of the signal we need to transform this signal into one describing point in a high-dimensional phase space. To do it we treat few consecutive values

as coordinates of one point in the phase space. Usually all values are normalised at the very beginning of analysis to simplify computations. Number of values taken from a stream and treated as coordinates of points depend on the dimensionality of phase space. Usage of all points that were captured in such a way results in attractor reconstruction. In many cases not all values are used — this is called "stroboscopic coordinates". To avoid visual clutter only one of every N points is drawn. Number of non-drawn points is determined by parameter τ, called time delay; it is multiply of time between points of original time series. Dimension and Lapunov coordinates of original attractor and attractor reconstructed using stroboscopic coordinates and using all points from original signal are the same.

At the same time choosing proper value of time delay τ influences our ability to analyse the signal. If τ is too large input points lie too far away from each another to provide enough information. If it is too small, input points lie too close which may suggest presence of non-existent linear dependency in the signal. One of the possible methods to find proper value of time delay is to find period (even non-exact one) and to choose value slightly smaller than found period. If there is no visible period, one can use autocorrelation (and take half of maximum value) or mutual information. In case of mutual information we take value of the first minimum of this function.

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

This function is constant or oscillates while τ increases when data is generated by periodical systems. In chaotic system value of this function decreases rapidly when τ increases. This function is therefore useful for determining whether underlying mechanism is periodic or chaotic.

Recurrence plot is based on idea of calculating attractors. It is used to compare all possible states that are represented by trajectories of points in the high-dimensional phase space. If such trajectory goes through region that is close to previous one, it is matched as recurrent. Recurrence plot is the chart showing all periods when dynamic system's state is repeating. Usually phase space has dimension much larger than possible to visualise and understand by human. Recurrence plot (proposed by Eckmann in 1987) allows to show on 2D chart all repeating states of system and is based on matrix of similarity. Positive Lapunov points are matched by diagonal lines' lengths.

Recurrence plot can be defined as Haeviside function over difference of distance of points is space (over some metrics) and the threshold. Its main three parameters are τ, dimension and threshold ε. Too small threshold means that some points that are far away will be taken as close ones; such situation can occur in the system where

much of the values are very small and from time to time there is large spike, like ECG signal.

$$R_{ij} = \Theta(\varepsilon - \|x_i - x_j\|) \tag{2}$$

Single recurrence plot is 2D matrix of values from set of $\{0,1\}$. Recurrence plot is symmetrical amongst diagonal. It can be plotted on the screen or the paper. Black dot (value of 1) at coordinates (i,j) means that on system at time i and j was in similar state, because its attractor was represented as points that were close together (their distance was less than the chosen threshold). This means that dot is plotted if two sequences coming from input data are similar (their product is larger than threshold). This allows for visually analysing similarity of signal at different scales. Similar techniques are used in analysis of gene sequences (FASTA, BLAST) to find similar gene sequences. This technique requires large amounts of memory and long processing.

Recurrence plot can be used as a mean for visual analysis of the self-similarity of signal, but also to find numerical characteristics of analysed dynamic system. The most important parameter used in analysis described in this paper is laminarity showing how stable the system is. This can be determined by measuring length of horizontal (or vertical, as recurrence plot is the symmetrical matrix) lines. According to the web page http://recurrence-plot.tk/ horizontal (and vertical) lines point to the periods where system does not change much. Another important factor is divergence, pointed by length of diagonal lines. Diagonal lines point to the states where system is oscillating and trajectory returns to the close subspace. It can be connected to positive Lapunov exponents and point where signal is repeating itself.

Dividing entire signal into parts and generating recurrence plots for each of them results in series of recurrence plots. This allows for temporal analysis of lengths of lines present in the plot. Technique used in this article creates many recurrence plots, each starting one point later than the previous one. For each of calculated plots lengths of horizontal and diagonal lines are calculated; then maximum lengths of appropriate lines create charts used in analysis of system (and user) behaviour.

4. Characteristics of gathered data

Figure 1 shows the first 25 components of FFT signal. Single mode (shown in part b) of the figure) has the lowest values of power, hence it is presented on a separate chart. Its values are about 500 times smaller than those for other situations. At the same time this chart shows the largest variation of frequencies present in signal. This

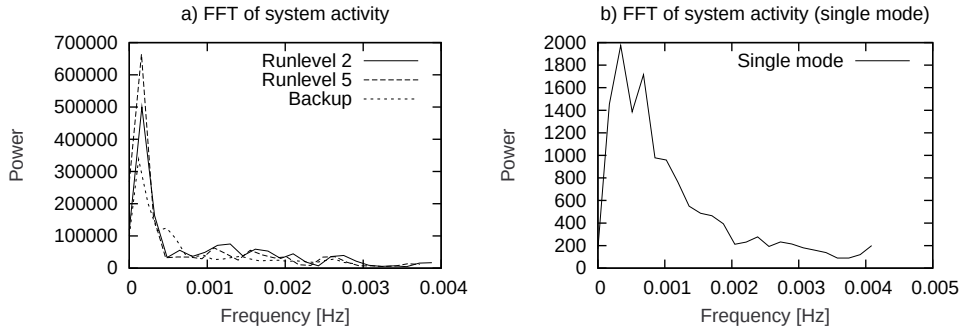

Fig. 1. Power spectrum of analysed signal. a) Run-level 2, Run-level 5, and user activity; b) Run-level Single

means that in case of Single run-level noise is much more visible. This is similar to the so-called pink noise.

Other run-levels' frequencies are in similar range, although they differ in exact values of power present in signal. All of those signals (shown in part a) of Figure 1) have very distinctive main frequency. This was caused by disk activity during scanning disk in search of suspicious files. In the case of the last data set, presenting system with running user-initiated programs, it generates second frequency in the chart. The large first part of tail in the case of backup is caused by inter-connection of programs that were being run during backup process. Usage of the results from one program by another one can cause resonance visible in the figure.

Figure 2 shows interdependence of the original signal and signal after some time. The least amount of mutual information in the signal in in the Single mode, which.is to be expected, as the smallest number of programs were running there. The amount of mutual information in signal from Single mode does not decrease over time. Run-levels 2 and 5 contain more information. They have similar overall shapes of curves but chart of run-level 2 is more smooth. This comes from fact that in this mode less programs are running (no digital clock, no screen-saver, no other graphical utilities), so there is less interaction. But similar shape means that changes caused by interactions between those new programs do not have long-lasting consequences. This suggests that those programs do not run for long time and also one program do not cause effects that affect running of other programs. Signals coming from run-level 2 and 5 show small decrease in value of mutual information. At the same time mutual information in the case of user interaction with the system is much larger, and decreases very rapidly.

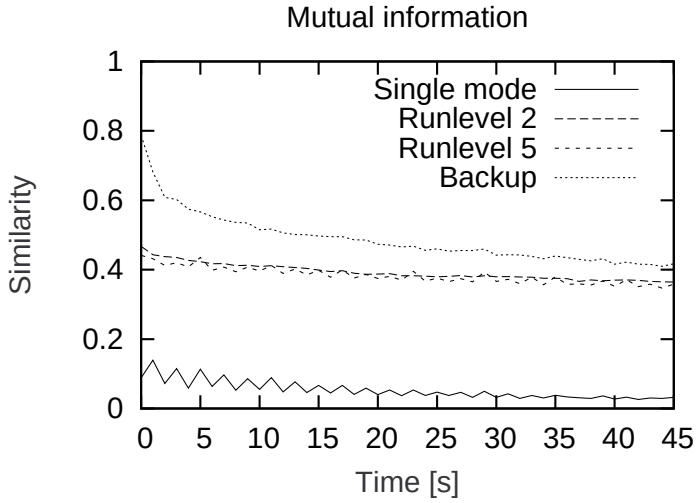

Fig. 2. Mutual information of signal compared to itself after time

Recurrence plots was created with following values of parameters: $\tau = 3$, dimension $m = 2$ and threshold $\varepsilon = 0.02$. Detailed analysis of single recurrence plots was described in [8]. This article is extension of previous work.

Window of size 150s was chosen for analysing series of recurrence plots. Analysis for windows of size 60, 90, 120, 150, 180, and 240s was performed initially. For short windows images were too noisy, and longer windows resulted in disappearance of fine details. Hence our decision of size of window 150s. Of course for different situations (especially different types of collected data) it might be necessary to chose different size of window.

N-150 recurrence plots were generated for signal of length N. Then, for each of recurrence plots from the window the longest diagonal and horizontal lines were found. Following charts show how much those maximum lengths were changing over time of experiment.

Figure 3 a) shows shows number of interrupts per second in single mode, in which almost no service was active; Figures 3 b), c), and d) show sample recurrence plots that were generated from this data. Although mean activity was constant, one can see changes in the signal. In the beginning (first 3 minutes) signal shows great variability. This is visible on the recurrence plot Figure 3 b) and is caused by post-startup activity of the system. This is rather early phase of running of the system

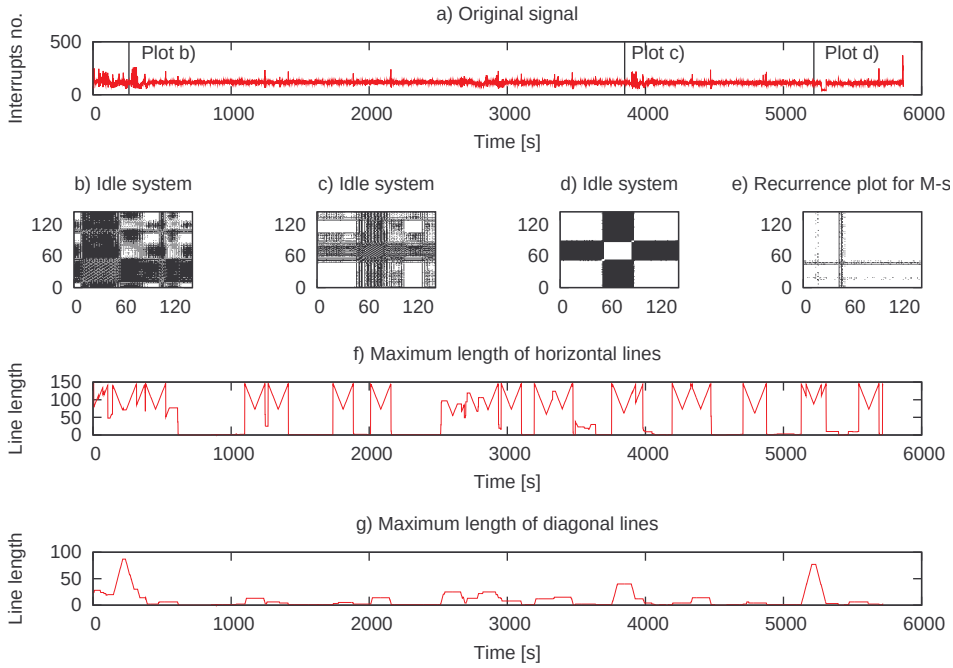

Fig. 3. Recurrence Plot for Single run-level. a) Original signal; b)-e) recurrence plots; f) maximum lengths of horizontal lines g) maximum lengths of vertical lines

which means that some programs are still initialising itself. In later chart signal is much more nice. Only part shown in Figure 3 c) shows increased activity; because, as mentioned in Section 2. system was connected to the internet, it is probable that it was someone trying to connect to the system.

Figure 3 f) shows maximum length of horizontal lines As mentioned in section describing recurrence plots (Section 3.), horizontal (and vertical) lines point to stationary parts of the system, when system is laminar (it does not change or changes very slowly). One can see that there are periods when horizontal lines are large, and when they are short. In most situations we can observe "M-shaped" structures. At the beginning there is no horizontal line, then there is jump to maximum length, then some slight but steady decrease of length, but not much (less than 50%) and again growth, and then almost immediate drop to zero. Those occurrences are connected to the spikes in the signal, and they can be used to detect such rapid changes in values in signal.

Diagonal likes (Figure 3 g)) are not long, as system is not changing much, and long diagonal lines point times when system is rapidly changing.

Figure 4 shows number of interrupts in run-level 2, without active graphical environment. Figure 4 b) shows recurrence plot from the period of disk-scan, and part c) situation when there was not activity. Situation seen in the latter figure is not interesting, and is similar to situation from run-level Single. This plot differs from the one shown in [8] as it does not show any details. This is due to choosing threshold which meant that for the idle period no points were printed. On the other hand choosing lower threshold would mean that recurrence plot in the period of activity would be almost entirely black; values differ over ten times between those cases. Changing threshold in course of analysis is something to investigate but it could taint data and results. Normalisation of data can also influence shape of recurrence plot. We are not yet sure how to deal with comparing raw recurrence plots from highly variable signals.

Figure 4 b) shows fragment of run-level 2 during high activity (disk scan, Phase 1). Signal changes but some self-similarity can be seen. Figure 4 c) is also run-level 2, but this is fragment after disk scanning, in the idle mode (Phase 2). Here recurrence plot is almost entirely black: this means that there is much similarity in signal. This also means that there is no long-lasting changes in signal, at least not enough to be detected by the plot. This situation differs from from the Single run-level. Additional daemons (programs constantly running in the background) caused change in activity and system switches more often to serve them, which results in more variable signal.

Analysis of idle period did not result in any points in recurrence plots. The only non-zero parts in graphs of maximum lengths of horizontal and diagonal lines,

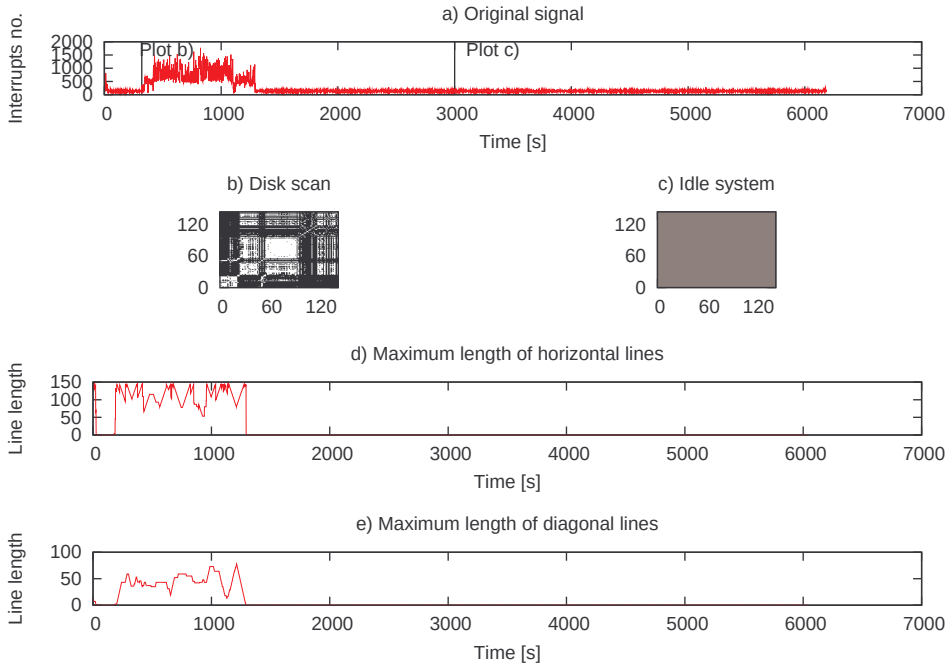

Fig. 4. Recurrence Plot for Run-level 2. a) Original signal; b) and c) recurrence plots; d) maximum lengths of horizontal lines e) maximum lengths of vertical lines

showing stability (Figure 4 d)) and divergence (Figure 4 c)) of system are those from the full disk scan.

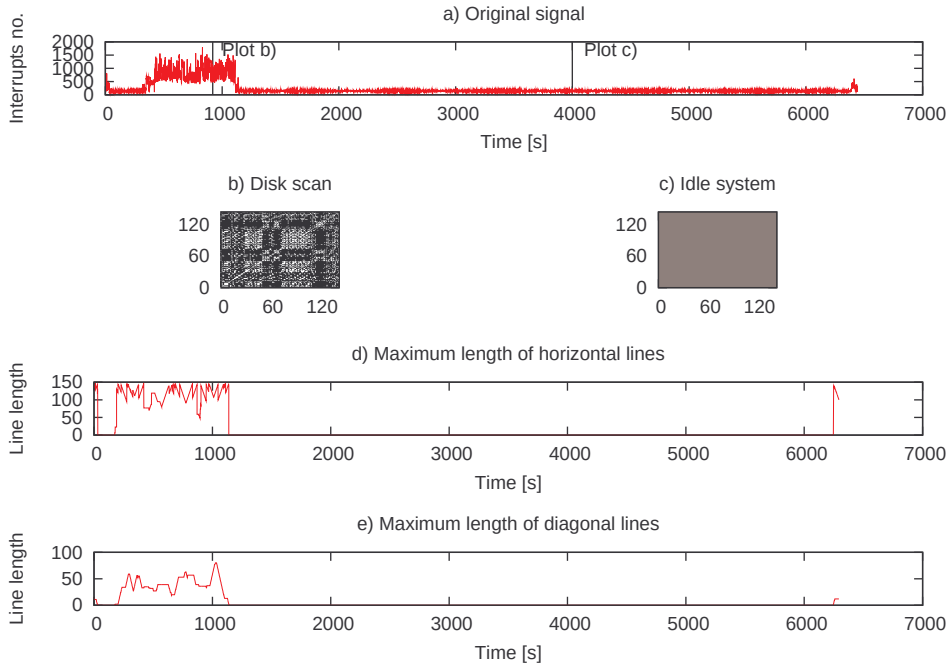

Fig. 5. Recurrence Plot for Run-level 5. a) Original signal; b) and c) recurrence plots; d) maximum lengths of horizontal lines e) maximum lengths of vertical lines

Figure 5 a) presents number of interrupts in run-level 5 with active graphical session. Here situation is very similar to situation in run-level 2, described in previous paragraphs. Similarity of those two situation was also described in previous paper [8], Both recurrence plots and graphs of stability (Figure 5 d)) and divergence (Figure 5 e)) look almost exactly the same.

For run-level 5 Figure 5 b) shows recurrence plot during disk scanning (Phase 1). This plot is similar to one seen in run-level 2 (part a of this figure). But in case of later activity, after disk scanning (Figure 5 c), image is different from run-level 2. This means that although there is self-similarity in signal, it is not on so many levels. Here again additional graphical programs cause “ripples’ that cause signal to be less smooth. The same situation was seen in information plot (Figure 2) and in FFT (Figure 1).

Figure 6 a) shows active graphical environment with user logged-in and during backup creation. We can see four different phases: transmission of files over the network, shown in Figure 6 b), compression of all files and making them ready to save to the CD (Figure 6 c)), burning data to CD (Figure 6 d)), and reading files from CD (Figure 6 e)).

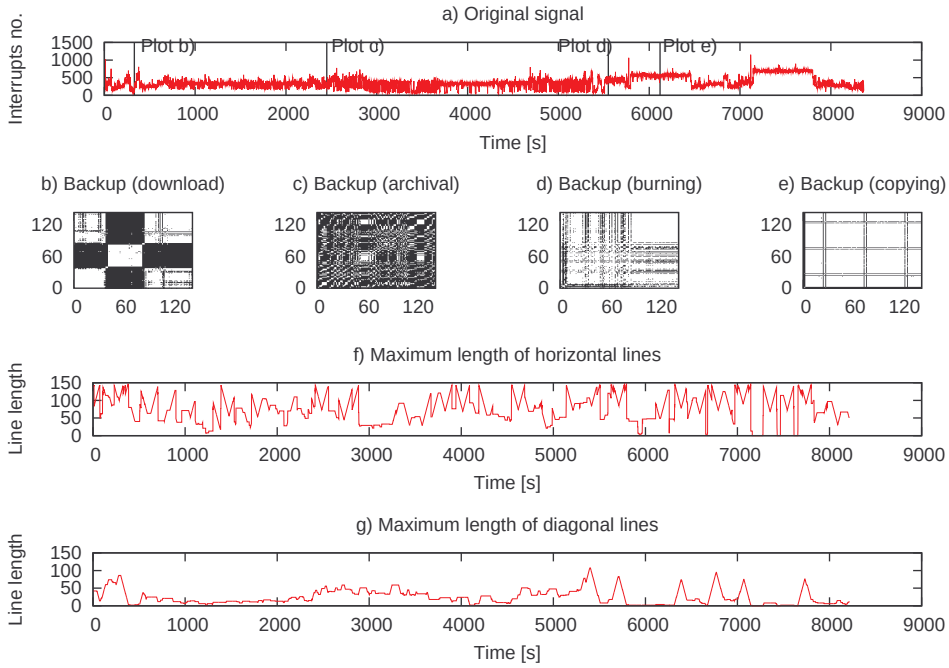

Fig. 6. Recurrence Plot for system with user activity. a) Original signal; b)-e) recurrence plots; f) maximum lengths of horizontal lines g) maximum lengths of vertical lines

5. Discussion

Figure 2 shows signal's mutual information. It shows that introducing user's behaviour changes amount of inter-dependency of signal with respect to time. Charts showing idle systems are flat; adding new programs (system daemons in case of run-level 2 and graphical environment for run-level 5) increases values but does not change shape of the system's mutual information. When user is active, however, situation changes dramatically. Mutual information is much larger and drops rapidly

after the first few seconds. This means that user's activities affects both long- and short-term memory of the system, but as the time passes traces of user's behaviour disappear and inter-dependency of signal presenting user's activity decreases to levels present in systems without user's actions.

This is the result of running programs working on the same task (creation of backup copy of entire system) so programs depend on each other as results generated by one program is input data for the next one. Running more programs causes introduction of information into the system. Further analysis of mutual information is needed to determine influence of different tasks: CPU-bound, limited by memory, disk, and other subsystems on mutual information charts.

First three figures showing recurrence plots (3, 4, and 5) show situations without any user activity thus showing idle system state. One can observe that first chart presents rather constant mean activity with small variation. Next two charts show two different stages; one (shown in Figures 4 c) and 5 c)) is similar to single run-level but with slightly larger mean value. Both of them show large activity about two minutes after system start and lasting for 10 to 15 minutes. This is caused by service monitoring changes in files on entire hard-drive checking every file on disk. This lasts and also caused running of many processes.

From analysis of system's configuration (details in [8]) it is clear that single mode (Figure 3) can be treated as entry (basic) level and presents the least sophisticated situation. Its activity changes only slightly, and can be treated as constant. This activity comes mostly from kernel responding to hardware and time events. Run-levels 2 and 5 are quite similar, but 5 has higher level of base (average) activity (when no user programs are running) because more programs are running in the background as run-level 5 activates graphical environment, as opposed to run-level 2. They both present much activity starting at 5 minutes from system start up to about 20 minutes. This is caused by cron job that is checking validity and consistency of installed packages. After finishing it no other non-standard program is running.

As expected, running more processes in the higher run-levels introduces more variability to signal. On the other hand when entire system is busy running the same task situation is similar regardless of active run-level, as can be seen on Figures 4 a) and b) and 5 a) and b), which are very similar — even if the former comes from system without any X-Window programs, and latter from active X session. But in this situation entire system is busy checking all files present on hard drive — so any differences disappear eclipsed by this activity.

As seen in Figure 6 e) showing self-similarity of signal during burning image to CD, hardware-induced events are very regular. This means that system is influenced not only by running programs and user's actions but also by external sources of

events, like network (where arrival of packet causes kernel to serve it) or hardware events (sector read from the disk, empty buffer in the CD-ROM burner, sample ready to be read from the sound card). As system has no control over external environment and only partial control over hardware used to communicate with physical environment, those events can be seen as coming from outside of the system.

Figure 6 c) presents recurrence plot of CPU-intensive task during backup procedure (compression of files). We can see that there is high level of self-similarity. Figures 6 d) and e) show signal during disk operations (burning image to CD). This signal has very distinctive structure with period of about 40 seconds. It is caused by buffering: kernel transmit data to DVD drive until its buffers are full, and then can do switch to other tasks. When drive has no data it informs kernel which then again transmit data needed to fill-up buffers. We can estimate that drive has buffer sufficient to store data for about 40s of work.

Figure 6 g) shows behaviour of system through lengths of diagonal lines. At the very beginning length of diagonal lines grows which means that system is repetitive. After the end of disk scan maximum length of diagonal lines dropps significantly which means that system's behaviour is more chaotic. After starting backup procedure the length of diagonal lines started growing; it means that repetitive process was running again. During creation of backup maximum lengths of diagonal lines was changing: it decreased, increased, then some spikes started to show. Those changes point presence of some processes that were reoccurring at very few occasions.

Figures 4 d) and 5 d) show maximum lengths of horizontal lines in recurrence plots for run-levels 2 and 5. In those run-levels for the first twenty minutes was busy checking all files on the hard drive, and then went idle waiting for user interaction. The final minute of activity should be not taken into consideration as it shows activity during system shutdown, as described in section 2.. In both of those cases system is showing stationary and recurrent behaviours only during disk-scanning phase. Idle system does not show any recurrent or stationary behaviour which means that idle system presents chaotic behaviour. Programs run by user can thus be seen as behaviours that introduce order into initially chaotic system. This is confirmed by Figure 6 f) which shows increased lengths of horizontal lines. It means that running programs increase amount of stability in the system.

"M-shapes" appear again on Figure 6 f); similar shapes were present on Figure 3 f). They are again caused by spikes present in the original activity chart. Figure 3 e) shows horizontal lines that span through almost entire chart. Horizontal (and vertical) lines do not come through entire plot but are separated at the crossing. There is small gap in the lines where they cross; it is not visible in the chart with the naked eye, but

it exists and limits length of the line in the chart. This means that when the crossing is in the middle of the plot the line is the shortest. The line gets longer when lines move away from the middle, as more of the line is not separated. The horizontal line is thus the longest when gap is at the border of chart. When this gap moves to the center of the plot, length of horizontal line decreases. This is the first part of the "M-shape"; then gap moves away from the chart, length of line grows and second part the this shape begins. The exact source of system's activity causing occurrence of spikes in the signal is currently unknown. Because the main purpose of presented research was detection of changes caused by user's actions, network was not disabled to limit changes introduced to system by process of conducting experiment. This could be caused by some network activity, like ARP requests, tries of infecting machine, etc. Those "M-shapes" on chart showing maximum length of horizontal lines can be used to detect such rapid changes in signal. This behaviour is not visible on other run-levels (shown in Figures 4 b) and 5 b)), though. Horizontal lines on the system with user's requested actions are much less regular, as can be seen in Figure 6 b). This additional source of events introduced by user's actions is more powerful and can occlude subtle differences introduced by external network packets coming to the system.

Lengths of diagonal lines in Single run-level recurrence plot is smaller than for other run levels (Figure 3 c)). Charts showing maximum lengths of diagonal lines in run-levels 2 and 5 (Figures 4 c) and 5 c) respectively) look similar to charts showing horizontal lines from those recurrence plots (Figures 4 b) and 5 b)).

6. Summary

This article presents analysis of system activity using series of recurrence plots and lengths of lines present on those plots. Number of interrupts fired in each second was measured and treated as a signal describing the system activity. Data from four different configurations of the system was gathered, with and without user activity. We have shown that system activity can be investigated and explained using non-linear methods.

Signal was analysed using a mutual information and recurrence plots. We have shown that introducing user to the system increases amount of information present in the system and the lone value of mutual information can be used to determine whether the system is idle or if user is active. Shape of mutual information plot can be used to determine whether the system is running short-living programs, or whether it is busy with long-lasting, intensive tasks.

The majority of our investigation focused on using of recurrence plots as means of analysing changes in the signal and detecting trends. Human user is operating on

time scale of minutes so gathering signal once every second and analysing trends over many seconds is a good compromise between detecting events in the system and limiting amount of data to analyse.

Research described in this paper focused on using a number of interrupts as a measure of the system activity. Number of interrupts is an important variable in cases with small sets of running programs and when running programs are dependent on hardware (network transmission, reading and writing CD). Context switch means stopping one program and starting another. It is used in modern operating systems to achieve multiprocessing, or at least illusion that many programs are running at the same time. The larger number of context switches, the more frequently kernel switches tasks. A large number of contest switches means that there are many programs running on the system, and all of them are waiting for CPU (i.e. not many are waiting for some external event). Number of context switches, as noted in Section 2., can be useful in analysing systems in which many programs are running. A difference between those variables and their interdependencies call for further research.

As noted in Section 3. many parameters can influence shape of recurrence plot. Further research is needed to investigate influence of parameters (like threshold ε, or time delay τ) on results that can be obtained during analysis of data. Differences between charts presented in previous ([8]) and current article show that it is crucial to use proper values of parameters. Improper values can cause disappearance of important details in the noise. Automatic procedure of generating proper values of parameters can be very helpful with analysis of user's behaviour.

Like Hoffman [2] we claim that observing real system is crucial; this is why we use existing tools that do not influence system much. We also intend to find ways of real-time automatic analysis of gathered signal. As noted in a comment to CACM article[1]: "Another [usage of growing computing capabilities] might be defensive computer security, analyzing past and current patterns of activity on the machine, communicating with other machines, and working to prevent malicious activity."

We were able to show that analysis of computer system activity allows to detect changes introduced by user's behaviour. User's actions change state of the system through running programs and requiring processing of data. This changes state of the system; but those changes are not always visible in the signal without performing analysis on it. Presented analysis using recurrence plots generates chart which allow easily to detect user's behaviour, such as transferring data over the network, writing

[1] http://cacm.acm.org/blogs/blog-cacm/23833-what-to-do-with-those-idle-cores/fulltext

data to compact disk, compressing data, or reading data from the external storage. By analysing characteristics of the recurrence plots and periodicity of the data we were able to detect rapid changes in the signal and distinguish between different signal types, which leads to detecting user activity that generated particular signal shape.

Acknowledgements

We would like to thank the anonymous reviewer for thoughtful and valuable comments that improved the article, and to our colleagues that helped with the language mistakes.

References

[1] Matthew Garrett. Powering down. *Communications of ACM*, 51(9):42–46, 2008.
[2] Bill Hoffman. Monitoring, at your service. *Queue*, 3(10):34–43, 2005.
[3] George V. Neville-Neil. Kode vicious beautiful code exists, if you know where to look. *Communications of ACM*, 51(7):23–25, 2008.
[4] Mark Oskin. The revolution inside the box. *Communications of ACM*, 51(7):70–78, 2008.
[5] George Porter and Randy H. Katz. Effective web service load balancing through statistical monitoring. *Communications of ACM*, 49(3):48–54, 2006.
[6] Daniel Rogers. Lessons from the floor. *Queue*, 3(10):26–32, 2005.
[7] Jerry Rolia, Ludmila Cherkasova, Martin Arlitt, and Vijay Machiraju. Supporting application quality of service in shared resource pools. *Communications of ACM*, 49(3):55–60, 2006.
[8] Tomasz Rybak and Romuald Mosdorf. Computer users activity analysis using recurrence plot. In *International Conference on Biometrics and Kansei Engineering*, Cieszyn, Poland, 2009. AGH.
[9] David E. Shaw, Martin M. Deneroff, Ron O. Dror, Jeffrey S. Kuskin, Richard H. Larson, John K. Salmon, Cliff Young, Brannon Batson, Kevin J. Bowers, Jack C. Chao, Michael P. Eastwood, Joseph Gagliardo, J. P. Grossman, C. Richard Ho, Douglas J. Ierardi, István Kolossváry, John L. Klepeis, Timothy Layman, Christine McLeavey, Mark A. Moraes, Rolf Mueller, Edward C. Priest, Yibing Shan, Jochen Spengler, Michael Theobald, Brian Towles, and Stanley C. Wang. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of ACM*, 51(7):91–97, 2008.

WYKRYWANIE AKTYWNOŚCI UŻYTKOWNIKA PRZY UŻYCIU ANALIZY RECURRENCE PLOT

Strzeszczenie: Dzięki nieustannemu wzrostowi wydajności systemów komputerowych możemy gromadzić coraz więcej danych opisujących aktywność systemów. Dane te mogą być analizowane aby zyskać wgląd w zachowanie użytkowników. Uważamy że systemy komputerowe, z racji działania w nich wielu programów które wpływają wzajemnie na siebie, majaą charakter nieliniowy. Dlatego też spośród wielu istniejących metod analizy dużych zbiorów danych zdecydowaliśmy się na użycie nieliniowych metod analizy.
Artykuł przedstawia wykorzystanie nieliniowych metod w celu wykrycia subtelnych zmian wprowadzonych do systemu poprzez działanie użytkownika. Analiza skupia się na porównaniu systemu bezczynnego i takiego w który działają programy uruchomione przez użytkownika. Jako zmienna najlepiej charakteryzująca system została wybrana liczba przerwań na sekundę. Artykuł przedstawia użycie wykresu recurrence plot w celu wykrycia podobieństw w zachowaniu systemu, a przez to w działaniu użytkownika.
Badanie systemu wykorzystuje serię wykresów aby wykryć charakter zmian wprowadzonych przez użytkownika. Analiza długości pionowych i ukośnych linii pozwala na wykrycie okresowych zachowań komputera, a tym samym na lepsze zrozumienie procesów zachodzących w całym systemie. Pokazane zostało że różne zadania (transmisja danych przy użyciu sieci komputerowej, nagrywanie plików na dysk CD, odczyt plików z dysku DVD, kompresja danych) generują różne wykresy recurrence plot. Ponieważ zmiany stanu systemu nie znajdują odzwierciedlenia w sygnale przedstawiajaącym liczbę przerwań na sekundę, użycie recurrence plot jest kluczowe do wykrycia zmian spowodowanych przez użytkownika.

Słowa kluczowe: badanie aktywności, aktywność systemu, analiza fraktalna, recurrence plot

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/09.

METHODS AND TOOLS FOR HIGHER EDUCATION SERVICE QUALITY ASSESSMENT (SURVEY)

Hanna Shauchenka[1], Eugenia Busłowska[2]

[1]School of Computing and Mathematics, University of Plymouth, Plymouth
[2]Faculty of Computer Science, Białystok University of Technology, Białystok

Abstract: The concept of quality of education in education, a number of years ago and is associated with the achievement of agreed standards, as well as consistency between the objectives of the program and the competence of graduates. There is no uniform terminology and definition of ”quality education” is understood as in any field. By E. Deming, ”Quality is what satisfies and even delights the customer”. Customer in college are students who express an opinion on the services received.

Keywords: quality in education, Total Quality Management, service quality

1. Introduction

The concept of quality in education is quite new and until now not a well developed field of study. There is no unified terminology and the term ”quality of education” is understood in different ways by different authors. All authors, however, adapt the concept of quality of education from industry, as in the following definitions: excellence in education [1] value addition in education [2] fitness of educational outcome and experience for use [3]; defect avoidance in the education process [4]; meeting or exceeding customer's expectations of education [5]. Liberalization and economization; growing competitiveness are the most significant reasons for growing importance of the quality concept in higher education.

The application of Total Quality Management (TQM) philosophy and methodology in the context of higher education is fully acknowledged and widely used today [6, 7, 8]. The necessity to use the TQM philosophy is aimed at providing educational services and giving results of academic and research activities. TQM methods are also implemented in the Quality Standards ISO 9001-2000, also known as the basic quality principles for Higher Education Institutions by Bologna Process.

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 87-102, 2010.

On the international political level the necessity of Quality Management in Higher Education is formulated and conceptualized in the mentioned Bologna Process and in the variety of complementary communiqués. There are following historical steps in recognition and discussion on Higher Education Quality importance.

- 1999: 29 European countries signed the so called Bologna Declaration. The main important issue of this declaration in the context of this research brings the principles of ISO 9001-2000 in the sector of education, which according to this declaration have a crucial impact on the educational sector. The quality model in ISO 9001:2000 is quite different in comparison with 9001-1994. It is now based upon a Process Model that any Higher Educational Institution can use.
- 2001: Prague Communiqué with its key issue that quality management systems should assure a high level of quality in higher education and provide the comparability of qualifications within the EU.
- 2001: Bergen Communiqué with its key issues about further development concerning the participation of students in a quality management process and international co-operation.
- 2003: Berlin Communiqué that declares the importance of effective quality policy and the development of quality criteria and methodology on different levels.
- 2005: Helsinki Standards and Guidelines for Quality Assurance in the European Higher Education Area declared by the European Association for Quality Assurance in Higher Education. Key issues here underline the role of students in quality management process and necessity of permanent quality monitoring.
- 2009: Leuven and Louvain-la-Neuve, The European Higher Education Area in the new decade Communiqué of the Conference of European Ministers Responsible for Higher Education. Key issues: underlining life-long learning, student-centred learning, statement of the goals of European Higher Education, and others.

2. The analysis of service quality measurement instrument

As it has been pointed out by different authors, quality is only measured at the end of the process, that is, when the service has been concluded, and there is no way to change the client perception regarding the service received to meet his expectation. According to previous research regardless of the type of service, consumers basically use the same criteria to assess quality. Service quality is a general opinion the client forms regarding service delivery, which is constituted by a series of successful or unsuccessful experiences [9-14]. Two arguments must be taken into consideration

to assess this category, namely the customer's perception and his initial expectation regarding the service received.

2.1 SERVQUAL

"SERVQUAL" is one of the most extensively used service quality measurement instrument because of its easiness to use, possession of a simple structure and capability of generalization [9-11]. According to Zeithaml, Parasuraman and Berry, SERVQUAL is a universal method and can be applied to any service organization to assess the quality of services provided [9]. Regardless of the type of service, consumers basically use the same criteria to assess quality. Service quality is a general opinion the client forms regarding service delivery, which is constituted by a series of successful or unsuccessful experiences. Conceptual model of the SERVQUAL is based on the assessment if satisfaction is found in situations where perceptions of service quality meets or exceeds consumer expectations. The client satisfaction is a result of the difference between the expectations and performance obtained. In other words Service Quality is evaluated by comparison of customer perception with expectation $(SQ = P - E)$. The SERVQUAL scale compares consumers' perceptions of twenty-two aspects of service quality with their rating of each factor's importance (expected service quality) [9, 12, 13, 14]. In their initial study Parasuraman and associates found that there were ten determinants which characterize customers' perceptions of the service provided. However, as a result of a later study they reduced the ten determinants of service quality to five. They were able to identify the following five dimensions of service quality: reliability, tangibility, responsibility, security and empathy [11, 14, 15]. These dimensions are briefly commented below [14, 15]. Reliability is the most important dimension for the consumer of services. This dimension expresses the accuracy and dependability with which the company provides its services and allows getting the answer to the following questions: Is the company reliable in providing the service? Does it provide as promised? Tangibility concerns the service provider's physical installations, equipment, staff and any materials associated with service delivery. Since there is no physical elements to be assessed in services, clients often trust the tangible evidence when making their assessment. Responsibility is the demonstration of the company employee's capabilities of providing the best service for the customer. This dimension is responsible for measuring company and employee receptiveness towards clients. Security encompasses the company's competence, courtesy and precision in providing their service. This dimension allows getting the answer to the following question: Are employees well-informed, educated, competent and trustworthy? Empathy is the

capacity to experience another person's feelings. It can be formulated as the following question: Does the service company provide careful and personalized attention? SERVQUAL instrument was developed on the basis of these five dimensions, using twenty-two aspects (questions) of service quality to their rating. The SERVQUAL scale (questionnaire) has two sections: one to map client expectations in relation to a service and the other to map perception in relation to a certain service company. However, as suggested later the twenty-two attributes of the original SERVQUAL instrument, as well as five dimensions do not always accurately describe all aspects of a given service [16-18]. An adapted version of the SERVQUAL scale for Higher education services was proposed through a review of literature in [15]. Table 1 shows the adapted questionnaire model that was used to conduct the quality expectations and perceptions survey for the Production Engineering program at UNESP/Bauru by its students [15].

Table 1: The questionnaire for the High Education Service

Dimension	Expectation (E)	Perception (P)
Reliability	1. When excellent institutions of Higher education promise to do something in a certain time, they must do so. 2. When a student has a problem, excellent institutions of Higher education demonstrate sincere interest in solving it. 3. Excellent of institutions of Higher education will do the job right the first time and will persist in doing it without error.	1. When your institution of Higher education promises to do something in a certain time, it does so. 2. When you have a problem, your institution of Higher education demonstrates sincere interest in solving it. 3. Your institution of Higher education will do the job right the first time and will persist in doing it without error.
Tangibility	1. Excellent Higher education institutions must have modern equipment, such as laboratories. 2. Higher education institution installations must be well conserved.	1. Your Higher education institution has modern equipment, such as laboratories. 2. Your Higher education institution installations are well conserved.

Table 1: The questionnaire for the High Education Service

Dimension	Expectation (E)	Perception (P)
	3. Employees and teachers at excellent institutions of Higher education must present themselves (clothes, cleanliness, etc.) in an appropriate manner for their position.	3. The employees and teachers at your institution of Higher education present themselves (clothes, cleanliness, etc.) in an appropriate manner for their position.
	4. The material associated with the service provided in excellent institutions of Higher education, such as journals, printed matter, must have a good visual appearance and be up to date.	4. The material associated with the service provided in your institution of Higher education, such as journals, printed matter, has a good visual appearance and is up to date.
Responsibility	1. Employees and teachers at excellent institutions of Higher education promise their clients the services within deadlines they are able to meet.	1. Employees and professors at your institution of Higher education promise you the services within deadlines they are able to meet.
	2. The employees and teachers at excellent institutions of Higher education are willing and available during service providing.	2. The employees and teachers at your institution of Higher education are willing and available during service providing.
	3. The employees and teachers at excellent institutions of Higher education will always show good will in helping their students.	3. The employees and teachers at your institution of Higher education always show good will in helping.
	4. The employees at excellent institutions of Higher education are always willing to explain doubts their students may have.	4. The employees and teachers at your institution of Higher education are always willing to explain your doubts.
Security	1. The behavior of employees and teachers at excellent institutions of Higher education must inspire confidence in the students.	1. The behavior of employees and teachers at your institution of Higher education inspire confidence.

Table 1: The questionnaire for the High Education Service

Dimension	Expectation (E)	Perception (P)
	2. Students at excellent institutions of Higher education feel safe in their transactions with the institution.	2. You feel safe in your transactions with your institution of Higher education.
	3. The employees and teachers at excellent institutions of Higher education must be polite to the students.	3. The employees and teachers at your institution of Higher education are polite.
	4. The employees and teacher at excellent institutions of Higher education must have the knowledge needed to answer student questions.	4. The employees and teachers at your institution of Higher education have the knowledge needed to answer your questions.
Empathy	1. Excellent institutions of Higher education must have convenient business hours for all students.	1. Your institution of Higher education has convenient business hours for all students.
	2. Excellent institutions of Higher education must have employees and teachers who provide individual attention to each student.	2. Your institution of Higher education has employees and teachers who provide individual attention to each student.
	3. Excellent institutions of Higher education must be focused on the best service for their students.	3. Your institution of Higher education is focused on the best service for its students.
	4. Excellent institutions of Higher education must understand the specific needs of their students.	4. Your institution of Higher education understands the specific needs of its students.

These questions should be scored on a Likert scale from 1 (strongly disagree) to 7 (strongly agree). The scored results E and P from the two sections (Perceptions and Expectations) of Table 1 are compared to reach a parameter (difference) for

each of the questions, that is, the final score is generated by P–E. A negative result indicates that the perceptions are below expectations, revealing the service failures that generate an unsatisfactory result for the client. A positive score indicates that the service provider offers a better service than expected. The main idea of the SERVQUAL authors was that differences between the perceived performance and expected performance determine overall service quality and can be evaluated with the following P–E measurement model [19].

$$SQ_i = w_j(P_{ij} - E_{ij}) \tag{1}$$

Where SQ_i is the overall perceived service quality of stimulus i; k – number of attributes; $P_i j$ is the performance perception of stimulus i with respect to attribute j; $E_i j$ represents service quality expectation for attribute j that is the relevant norm for stimulus i; and w_j is the weighted factor if attributes have different weights [20, 21]. SERVQUAL has been criticized for not being applicable to all services without modification as the dimensions are dependent on the type of service [22-26]. Moreover, some authors described that the Service Quality differs from industry to another industry of services [22, 27]. New factors should be added and taken into account based on generic dimensions and appropriateness of services sectors. Further research has been conducted and new results show that perceived quality alone correlates better with service quality than does the SERVQUAL gap analysis of the difference between the perceived and expected quality [23]. According to the research that has been presented in [23] the conventional disconfirmation model has conceptual, theoretical and measurement problems. Some of the issues are pointed out within the framework of the service quality measurement by SERVQUAL [11]. Because of these problems, the following evaluated performance perceived model is developed addressing the ideal point problem [28, 29, 30] by formally incorporating the classic ideal point concept into the perceived quality model [24]. Within this research the service quality has been used according to Monroe and Krishnan's perceived product quality definition as "the perceived ability of a product to provide satisfaction relative to available alternatives" [31]. On the basis of this definition and the assumption that the perceived ability of the product to deliver satisfaction can be conceptualized as the product's relative congruence with the consumer's ideal product features the following probabilistic model of perceived quality has been proposed [24].

$$Q_i = -1(\sqrt{r\sum_{j=1}^{m} w_j \sum_{k=1}^{n_j} P_{ij_k}|A_{j_k} - I_j|^r}) \tag{2}$$

Where Q_i is the individual's perceived quality of object i; w_i - importance of attribute j as a determinant of perceived quality; $P_i jk$ - the perceived probability that object i has amount k of attributes j; $A_j k$ - amount k of attributes j; I_j - ideal amount of attribute j as conceptualized in classical ideal point attitudinal models; m - number of attributes; n_j - number of amount categories of attribute j; r - Minkowski space parameter. Multiplication the right side of the equation by -1 results in larger values of Q_i being associated with higher level of perceived quality [24, 32].

The proposed perceived quality model (2) is a general model allowing for several alternative perceived quality concepts and measures derived from (2) and a simplified version of this model for Minkowski distance space parameter r=1. For example, if it is assumed that the individual evaluates object i with perceived certainty and that object i has a constant amount of each attributes the next deterministic model of perceived quality for r=1 can be derived [23, 24, 33].

$$Q_i = -1(\sum_{j=1}^{m} w_j|(A_{ij} - I_j)) \tag{3}$$

Where Q_i, w_i and I_j are defined in equation (2). $A_i j$ equals the individual's perceived amount of attribute j possessed by object i. This model is Manhattan Distance, or City Block Distance metric for ideal point model [32].

With an assumption that the perceived ability of the product to deliver satisfaction can be conceptualized as the product's relative congruence with the consumer's ideal product features. If the object i is defined as the excellence norm that is the focus of revised SERVQUAL concept, the above metrics can be used to define the perceived quality of excellence norm Q_e in terms of similarity between the excellence norm and the ideal object with respect to m attributes. The quality of another object i, Q_i relative to the quality of excellence norm then normed quality (NQ) is [24, 19].

$$NQ_i = (Q_i - Q_e) \tag{4}$$

Where NQ_i is the normed quality index for object i; Q_e represents the individual's perceived quality of the excellence norm object and Q_i is defined in equation (2). If the excellence norm is equal to the ideal or perfect object ($Q_e = 0$) then normed quality $NQ_i = Q_i$.

Last equations (3) and (4) can be used to derive the following modified SERVQUAL model that addresses the ideal point problem by incorporating the ideal point concept [24].

$$NQ_i = -1(\sum_{j=1}^{m} w_j|(|A_{ij} - I_j| - |A_{ej} - I_j|)|) \tag{5}$$

Where NQ_i is the normed quality index for object i; $A_e j$ represents the individual's perceived amount of attribute j possessed by excellence norm. The meaning w_i and I_j are the same as in equation (2) and $A_i j$ is defined in (3).

For infinite ideal points, normed quality is [24, 19].

$$NQ_i = \sum_{j=1}^{m} w_j (A_{ij} - A_{ej}) \tag{6}$$

Last equation is similar to the original SERVQUAL model described in (1). Two assumptions are used in the equation (6), namely all the m attributes have infinite classic ideal points and that the SERVQUAL normative expectations concepts is redefined as the excellence norm specified in (5).

Poor reliability and inter-factor correlations of SERVQUAL leads to proposing SERVPERF (perception-only model) and HEdPERF (Higher Education PERFormance) for efficient measurement of service quality [21, 34, 35].

2.2 SERVPERF

Due to the controversy relating to the SERVQUAL instrument, a more direct approach to the measurement of service quality has been proposed [26]. New approach was developed as the measurement instrument called SERVPERF which is used for the service quality assessment. The SERVPERF instrument like the SERVQUAL uses an attribute approach. But in comparison with SERVQUAL the SERVPERF tool measures only customers' experiences of the service. This instrument makes use of the original SERVQUAL scales. It also requires the consumer to rate the provider's service performance on a seven point scale. Comparing with SERVQUAL the SERVPERF uses a single set of questions concerning post consumption perceptions of service quality and does not seek to measure expectations [26].

It was illustrated that service quality is a form of a consumer attitude. Therefore, measuring only performance of service quality is an enhanced means of measuring service quality [10, 34]. According to this research service quality can be conceptualized as an attitude and can be regarded as the adequacy-importance model. Thus, service quality is evaluated by perceptions only without expectations and importance weights as follows [19, 21].

$$SQ_i = \sum_{j=1}^{k} P_{ji} \tag{7}$$

Where SQ is the overall service quality of object i; k – number of attributes; $P_i j$ is the performance perception of stimulus i with respect to attribute j [21].

2.3 HEdPERF

More recently, a new industry-scale called HedPERF (Higher Education PERFormance) has been developed comprising a set of 41 items [36]. This instrument aims at considering not only academic components but also aspects of the total service environment as experienced by the student. The author identified five dimensions of the service quality concept. Non-academic aspects. This dimension includes items that are essential to enable students to fulfill their study obligations, and relates to duties carried out by non-academic staff. Academic dimension. These are responsibilities of the academics. Reputation. Responsibility of higher learning institutions to project a professional image. Access dimension. This dimension includes such issues as approachability, ease of contact, availability and convenience. Programme issues. This aspect concerns the importance of offering a wide ranging and reputable academic programmes/specializations with flexible structure. The SERVPERF and HedPERF scales were compared in terms of reliability and validity. Consequently, the superiority of the new purposed measurement instrument [37] was concluded.

2.4 FM-SERVQUAL

FM-SERVQUAL was developed on the basis of original SERVQUAL i.e. through mechanism of comparison between customers' perception of the services provided by the local authority and expectations of services desired by customers [11]. It includes the use of Integrated Facilities Management Framework, combination of perception and expectation statements, using positive wording solely to avoid the confusion over the development of measurement element according to appropriateness of rule and function services of the local authority to the community.

FM-SERVQUAL instrument is able to measure Service Quality in the local authority delivery system [38]. FM-SERVQUAL can also serve as an essential gauge in policy formulation and future planning of an organization. FM-SERVQUAL is a tool for measuring Service Quality in local authorities through the comparison between customer perception and expectation of the quality of the services provided. The structured survey in such a design is suitable for collecting data in a big sample size for evaluating quality services in local authorities.

The process of constructing FM-SERVQUAL comprises several steps, it starts with defining the assessment of the Service Quality through the formula of SQ = P|E. The variation P|E where the perception of Service Quality received is asked with respect to the customer's expectation of what was actually received [38]. Then 90 items have been created that will characterize the concept of Service Quality based on

Integrated Facility Management Framework. The next steps deal with data collection and data analysis, service quality identification, and FM-SERVQUAL reliability and validity evaluations [38].

2.5 INTQUAL

Quality is widely studied using various adaptations of the SERVQUAL instruments as has been shown by the previously presented results. The internal service quality measures called INTQUAL were developed by Caruana and Pitt [39] as one of the SERVQUAL adaptation.

INTQUAL model is an internal service quality measure for service organization as an alternative to SERVQUAL that emphasizes customer's point of view. It is an attempt to establish the operational method for the internal service quality measurement.

INTQUAL is an adaptation of SERVQUAL model. It used by Berry and Parasuraman for service quality measures on management of expectation and service reliability as an adopted model for internal measure for service quality [40].

Frost and Kumar developed a conceptual model which they called INTSERVQUAL, based on the SERVQUAL scale proposed by Parasuraman et al. [41]. The study was conducted in a major international airline for measuring expectations and perceptions of internal customers. According to the authors, the two scales exhibited adequate validity as separate measures of front-line staff (customer-contact personnel) expectations of support services and their perceptions of the support staff's performance. The results indicated that the scales can be successfully used to assess the magnitude of the gap between front-line staff perceptions and expectations.

2.6 DL-sQUAL

DL-sQUAL was introduced as there were needs for an instrument to measure the quality of online education. Previous SERVQUAL and e-SQ models measured the quality of traditional and eCommerce services and there were no instruments available to measure the quality of distance learning services. In their research, Shaik et al., found that the DL-eSQUAL scale demonstrated psychometric properties based on the validity and reliability analysis [42]. Their findings from the exploratory research offered useful initial insights about the criteria and processes students use in evaluating distance learning services. These insights, in addition to serving as a starting point for developing a formal scale to measure perceived DL-sQUAL,

constituted a conceptual blueprint that distance learning administrators can use to qualitatively assess the potential strengths and weaknesses of their services. It also helps to target specific service elements requiring improvement, and training opportunities for staff. Analyzed at the item level, data drawn from application of the DL-sQUAL instrument have practical implications for distance learning administrators. This is an exploratory study with the goal of developing a DL-sQUAL scale. The scale should be viewed as a preliminary scale because the sample is limited to a single distance learning institution located in the south-east part of the United States and represents the service experiences of the students at that institution. Due to the limited nature of the sample, the results of this study cannot be generalized beyond the specific sample domain. The generalization of the results of this research study is also constrained by the absence of standardized data for comparison [43].

2.7 Conclusions and remarks

SERVQUAL is extensively used as a high education service quality measurement instrument due to its simple structure, generalization capability and the ease of use [21, 44, 45]. Nevertheless, since the quality of service largely depends on human behavior, the quality dimensions of the measuring instrument differ in different service settings. For example, empathy and responsiveness are more significant in the healthcare sector, whereas reliability is important in transportation [44]. That is why the SERVQUAL dimensions, and items under each dimension, are modified to suit a particular application [21, 44, 46, 47]. The more complicated modifications have been recognized as the new service quality measurement instruments: SERVPERF, HEdPERF, FM-SERVQUAL, Weighted SERVQUAL, Weighted SERVPERF and Weighted HedPERF [21, 44, 46, 47].

In the education sector, intangibility and lack of physical evidence of service makes the perceptions of service quality a complex composition and poses difficulties for analysis. The educational literature suggests how imperative it is for high education institutions to actively assess the quality of the services they offer and to commit themselves to continuous improvements of their service.

In order to evaluate the high education service quality fitting to most of the key stakeholders, a new attempt has to be made to propose a new instrument based on new approaches and techniques. At the same time the long practice and experimental application of SERVQUAL are quite important for further research. More than forty survey items relevant to high education service quality assessment compiled from various sources are considered in this study.

References

[1] Gilmore, H.L.: Product conformance Quality Progress, Vol. 7, No. 5, 1974, pp. 16-19.

[2] Brigham, S.: 25 Snapshots of a Movement: Profiles of Campuses Implementing CQI, American Association of Higher Education, Washington, 1994, DC. 187.

[3] Dorweiler, V.P., Yakhou, M.: Changes in professional degree programs in the USA: an environmental analysis of professional education requirements Vol. 13 No. 2, 1998, pp. 231-51.

[4] Crosby, P.B.: Quality is Free, McGraw-Hill, New York, 1979.

[5] Parasuraman, A., Zeithaml, V. A. and Berry, L.L.: A Conceptual Model of Service Quality and its Implication for Future Research, Journal of Marketing, Vol. 49 (Fall), 1985, pp. 41-50.

[6] Filippov, Vladimir: Defining the Principles of Cultural Heritage in the European Higher Education Area, 1 Higher Education in Europe, 2006, 31: 4, pp. 359 — 361.

[7] Sallis, E.: Total Quality Management in Education, Second Edition, Kogan Page, London, 1996.

[8] Bannister, D., Fransella, F.: The inquiring man: the theory of personal constructs, Penguin Books Ltd, England, 1971.

[9] Zeithaml, V. A., Parasuraman A., Berry L.: Delivering quality service: balancing customer perceptions and expectations, London, Macmillan, 1990.

[10] Parasuraman A., Zeithaml V. BERRY, L. A.: Conceptual model of service quality and its implications for future research, Journal of Marketing, 1985, vol. 49, p. 41-50.

[11] Parasuraman, A., Zeithaml, V., Berry, L.: SERVQUAL: A multi-item scale for measuring consumer perceptions of service quality, Journal of Retailing, 1988, vol. 64, p. 12-40.

[12] Parasuraman, A., Berry, L., Zeithaml, V.: Refinement and reassessment of the SERVQUAL scale, Journal of Retailing, 1991, vol. 67, pp. 420-450.

[13] Parasuraman, A., Zeithaml, V., Berry, L.: Reassessment of expectations as a comparison standard in measuring service quality, Journal of Marketing, 1994, vol. 58, pp. 111-124.

[14] Carl A. Ruby: Assessing Satisfaction with Selected Student Services Using SERVQUAL, a Market-Driven Model of Service Quality, NASPA Journal, 1998, vol. 35, pp. 331-341.

[15] Otávio José De Oliveira, Euriane Cristina Ferreira: Adaptation and application of the SERVQUAL scale in higher education, Proceedings of POMS 20th Annual Conference Orlando, Florida U.S.A., May 1-4, 2009.

[16] Gronroos C.: Service management and marketing: A customer relationship management approach, 2nd edition, John Wiley & Sons, West Sussex England, 2000.
[17] Cuthbert PF.: Managing service quality in HE: is SERVQUAL the answer? Part 1", Managing Service Quality, 1996, vol. 6, no.2, pp. 11-16.
[18] Cuthbert PF.: Managing service quality in HE: is SERVQUAL the answer? Part 2, Managing Service Quality, 1996, vol. 6, no. 3, pp. 31-35.
[19] Seth N., Deshmukh, S.G., Vrat P.: Service Quality Models: A Review, International Journal of Quality & Reliability Management, 2005, vol. 22, No.9, pp. 913-919.
[20] Teas R.K.: Expectations, performance evaluation and consumers' perceptions of quality, Journal of Marketing, 1993, vol. 57, pp. 18-34.
[21] Khan M.S.: Studies on some aspects of service quality evaluation with specific relevance to Indian service industries, PhD Thesis. National Institute of Technology, INDIA, April 2007.
[22] Carman J.M.: Consumer perceptions of service quality: an assessment of the SERVQUAL dimensions, Journal of Retailing, 1990, vol. 66 no. 1, pp. 33-55.
[23] Babakus, E., Boller. G.W.: An empirical assessment of the SERVQUAL scale, Journal of Business Research, 1992, vol. 24, pp. 253-268.
[24] Teas R.K.: Expectations, performance evaluation and consumers' perceptions of quality, Journal of Marketing, 1993, vol. 57, pp. 18-34.
[25] Brown T.J., G.A. Churchill Jr, and P.J. Peter: Improving the measurement of service quality, Journal of Retailing, 1993, vol. 69, pp. 127-138.
[26] Cronin, J.J. Jr, S.A. Taylor.: Measuring service quality: A re-examination and extension. Journal of Marketing, 1992, vol. 56, pp. 55-68.
[27] Taylor S., Baker T.: An Assessment of the Relationship between Service Quality and Customer Satisfaction in the Formation of Consumers' Purchase Intentions, Journal of Retailing, 1994, vol. 4, No. 2, pp. 163-178.
[28] Miller John A.: Studying Satisfaction, Modeling Models, Eliciting Expectations, Posing Problems, and Making Meaningful Measurement, in Conceptualization and Measurement of Consumer Satisfaction and Dissatisfaction, H. Keith Hunt, ed. Cambridge, NA: Marketing Science Institute, 1977, p. 72-91.
[29] Swan John E. and Trawick Frederick I.: Satisfaction related to Predictive vs. Desired Expectations, in Refining Concepts and Measure of Consumer Satisfaction and Complaining Behavior, H. Keith Hunt and Ralf L. Day, eds. Bloomington, IN: Indiana University, 1980, p. 7-12.
[30] Prakash Ved: Validity and Reliability of the confirmation of Expectations Paradigm as a Determinant of Consumer Satisfaction, Journal of the Academy of Marketing Science, 12 Fall, 1984, p. 63-76.

[31] Monroe, Kent B., and R. Krishna: The effect of Price on Subjective Product Evaluations, in Perceived Quality, Jacob Jacoby and Jerry c. Olson, eds. Lexington, MA: Lrxington Books, 1985, p. 209-232.

[32] Thompson, A.C.: Minkowski Geometry, Cambridge University Press, Cambridge 1996.

[33] Ginter James L.: An Experimental Investigation of Attitude Change and Choice of a New Brand, Journal of Marketing Research, 1974, p. 30-40.

[34] Cronin, J.J, Taylor, S.A.: SERVPERF versus SERVQUAL: Reconciling Performance-based and Perceptions-minus-Expectations Measurement of Service Quality, Journal of Marketing, 1994, Vol. 58, pp. 125-131.

[35] Abdullah F.: HEdPERF versus SERVPERF: The Quest for ideal measuring instrument of service quality in higher education sector, Quality Assurance in Education, 2005, Vol. 13, No. 4, pp. 305-328.

[36] Firdaus A.: The development of HEdPERF: a new measuring instrument of service quality for the higher education sector, International Journal of Consumer Studies, 30 (6): 569-581, 2006.

[37] Firdaus A.: Measuring service quality in higher education: three instruments compared, International Journal of Research & Method in Education, 29(1): 71-89, 2006.

[38] Wan Zahari, Wan Yusoff, Maziah Ismail: FM-SERVQUAL: A new approach of service quality measurement framework in local authorities, Pacific Rim Real Estate Society, 2008.

[39] Caruana A. and Pitt L.: INTQUAL – an internal measure of service quality and the link between service quakity and business performance, European Journal of Marketing, 1997, Vol. 31, No. 8, pp. 604-616.

[40] Prakash Ved: Validity and Reliability of the confirmation of Expectations Paradigm as a Determinant of Consumer Satisfaction, Journal of the Academy of Marketing Science, 1984, p. 63-76.

[41] Monroe Kent B. and R. Krishna: The effect of Price on Subjective Product Evaluations, in Perceived Quality, Jacob Jacoby and Jerry c. Olson, eds. Lexington, MA: Lrxington Books, p. 209-232.

[42] Shaik N., Lowe S., Pinegar K.: DL-sQUAL: A multiple-item scale for measuring service quality of online distance learning programs, Online Journal of Distance Learning Administration, IX(II), 2006.

[43] Farah Merican, Suhaiza Zailani and Yudi Fernando: Development of MBA Program-Service Quality Measurement Scale, 1. International Review of Business Research Papers. Vol. 5 No. 4 June 2009, pp.280-291.

[44] Mahapatra S.S., Khan M.S.: A framework for analysing quality in education settings, European Journal of Engineering Education, 2007, Vol. 32, No. 2, pp. 205–217.
[45] Philip G., Hazlett S.A.: The measurement of service quality: a new P-C-P attributes model, Int. J. Qual. Reliab. Mngt, 1997, 14, pp. 260–286.
[46] Saleh F., Ryan C.: Analyzing service quality in the hospitality industry using the SERVQUAL model, Service Ind. J., 1991, 11, pp. 324–343.
[47] Weitzel W., Schwarzkof A.B. and Peach, E.B.: The influence of customer service on retail store, J. Retail, 1989, 65, p. 27–39.

METODY I NARZĘDZIA OCENY JAKOŚCI KSZTAŁCENIA W UCZELNI WYŻSZEJ

Streszczenie Pojęcie jakości kształcenia w edukacji pojawiło się kilka lat temu i wiąże się z osiągnięciem przyjętych standardów, a także spójności między celami, programem i kompetencjami absolwentów. Nie ma jednolitej terminologii i określenie „jakość kształcenia" jest rozumiane tak jak w każdej dziedzinie. Wg E. Deminga „jakość jest tym, co zadowala, a nawet zachwyca klienta". Klientem w uczelni wyższej są studenci, którzy wyrażają opinię w zakresie usług otrzymanych.

Słowa kluczowe: jakość w edukacji, kompleksowe zarządzanie jakością , jakość usług

Artykuł zrealizowano w ramach pracy badawczej S/WI/5/08