On the Constrained Longest Common Subsequence Problem

Anna Gorbenko

***Abstract*—The problem of the longest common subsequence is a classical distance measure for strings. There have been several attempts to accommodate longest common subsequences along with some other distance measures. There are a large number of different variants of the problem. In this paper, we consider the constrained longest common subsequence problem for two strings and arbitrary number of constraints. In particular, we consider an explicit reduction from the problem to the satisfiability problem and present experimental results for different satisfiability algorithms. It should be noted that different regularities in experimentally obtained data reveal important information about the underlying physical system. In this paper, we consider the problem of systematic monitoring of passenger flows. In particular, we use constrained longest common subsequences for tracking the image features.**

***Index Terms*—longest common subsequence, satisfiability problem, feature tracking, genetic algorithms.**

I. Introduction

VARIOUS algorithms on sequences of symbols have been studied for a long time and now form a fundamental part of computer science (see e.g. [1]–[3]). One of the most important problems in analysis of sequences is the longest common subsequence problem. This problem has been studied extensively over the last thirty years (see [4]–[8]). There are a large number of applications of different variants of this problem (see e.g. [9]–[11]). In particular, we can mention robot self-awareness (see e.g. [12]–[19]), mining for interesting patterns (see e.g. [20], [21]), and automatic generation of recognition modules (see e.g. [22]).

In this paper, we consider the constrained longest common subsequence problem that was proposed in [23]. It should be noted that there are a number of efficient algorithms for the constrained longest common subsequence problem for two strings and one constraint (see e.g. [24]–[31]).

However, in general case, the constrained longest common subsequence problem is **NP**-hard [32], [33]. In particular, the **NP**-hardness and inapproximability of the constrained longest common subsequence problem for two strings and arbitrary number of constraints was proved in [32]. This paper is devoted to the consideration of efficient algorithms for the constrained longest common subsequence problem.

II. Preliminaries

Let $\Sigma = \{a_1, a_2, \ldots, a_m\}$ be a fixed alphabet. Given two strings S and T over Σ, the string T is a subsequence of S if T can be obtained from S by deleting some letters from S. Note that the order of the remaining letters of S should be preserved. The length of a string S is the number of letters in it. The length of a string S is denoted as $|S|$. For simplicity, we use $S[i]$ to denote the ith letter in the string S, and $S[i, j]$ to denote the substring of S consisting of the ith letter through the jth letter.

Given two strings S_1 and S_2, the classic longest common subsequence problem asks for a longest string T that is a subsequence of both S_1 and S_2. The decision version of the constrained longest common subsequence problem for two strings and arbitrary number of constraints can be formulated as following.

Constrained longest common subsequence problem (C-LCS-D):

Instance: *Two strings S_1 and S_2 over Σ, a set*

$$\{T_1, T_2, \ldots, T_n\}$$

of strings over Σ, a positive integer k.

Question: *Is there a string T over Σ such that*

- *$|T| \geq k$;*
- *T is a common subsequence of S_1 and S_2;*
- *T_i is a subsequence of T, for all $1 \leq i \leq n$?*

III. An explicit reduction from C-LCS-D to the satisfiability problem

The satisfiability problem was the first known **NP**-complete problem. Different variants of the satisfiability problem were considered. In particular, the 3-satisfiability problem (3SAT) is the problem of determining if the variables of a given boolean function in conjunctive normal form with 3 variables per clause (3-CNF) can be assigned in such a way as to make the formula evaluate to true (see e.g. [34]).

Note that 3SAT is **NP**-complete. However, there are a large number of different efficient satisfiability algorithms. Encoding various hard problems as instances of the satisfiability problem and solving them with efficient satisfiability algorithms has caused considerable interest (see e.g. [35]–[38]). In this paper, we consider an explicit reduction from C-LCS-D to the satisfiability problem.

Let

$$\varphi[p,1] = \bigwedge_{1 \leq i \leq k} \bigvee_{1 \leq j \leq |S_p|} x[p,i,j],$$

$$\varphi[p,2] = \bigwedge_{\substack{1 \leq i \leq k, \\ 1 \leq j[1] < j[2] \leq |S_p|}} (\neg x[p,i,j[1]] \vee$$

$$\neg x[p,i,j[2]]),$$

Ural Federal University, Department of Intelligent Systems and Robotics of Mathematics and Computer Science Institute, 620083 Ekaterinburg, Russian Federation. Email: gorbenko.aa@gmail.com

The work was partially supported by Analytical Departmental Program "Developing the scientific potential of high school", RFBR, research project No. 13-01-00048 a, and Ural Federal University development program with the financial support of young scientists.

(Advance online publication: 29 November 2013)

$$\varphi[p,3] = \bigwedge_{\substack{1\le i[1]<i[2]\le k,\\ 1\le j[2]<j[1]\le |S_p|}} (\neg x[p,i[1],j[1]] \vee \neg x[p,i[2],j[2]]),$$

$$\psi = \bigwedge_{\substack{1\le i\le k,\\ 1\le j[1]\le |S_1|,\\ 1\le j[2]\le |S_2|,\\ S_1[j[1]]\ne S_2[j[2]],}} (\neg x[1,i,j[1]]\vee \neg x[2,i,j[2]]),$$

$$\rho[q,1] = \bigwedge_{1\le i\le |T_q|} \bigvee_{1\le j\le k} y[q,i,j],$$

$$\rho[q,2] = \bigwedge_{\substack{1\le i\le |T_q|,\\ 1\le j[1]<j[2]\le k}} (\neg y[q,i,j[1]]\vee \neg y[q,i,j[2]]),$$

$$\rho[q,3] = \bigwedge_{\substack{1\le i[1]<i[2]\le |T_q|,\\ 1\le j[2]<j[1]\le k}} (\neg y[q,i[1],j[1]]\vee \neg y[q,i[2],j[2]]),$$

$$\tau[q] = \bigwedge_{\substack{1\le i\le |T_q|,\\ 1\le j\le k,\\ 1\le t\le |S_1|,\\ T_q[i]\ne S_1[t],}} (\neg y[q,i,j]\vee \neg x[1,j,t]),$$

$$\xi = (\bigwedge_{\substack{1\le i\le 2,\\ 1\le j\le 3}} \varphi[i,j]) \wedge \psi \wedge (\bigwedge_{\substack{1\le i\le n,\\ 1\le j\le 3}} \rho[i,j]) \wedge (\bigwedge_{1\le i\le n} \tau[i]).$$

It is not hard to verify that there is a string T over Σ such that $|T| \ge k$, T is a common subsequence of S_1 and S_2, and T_i is a subsequence of T, for all $1 \le i \le n$, if and only if ξ is satisfiable. It is clear that ξ is a CNF. Using standard transformations (see e.g. [39]), we can obtain an explicit transformation ξ into ζ such that $\xi \Leftrightarrow \zeta$ and ζ is a 3-CNF. Clearly, ζ gives us an explicit reduction from C-LCS-D to 3SAT.

Fig. 1. A typical example of data for monitoring of passenger flows.

IV. Monitoring of passenger flows

In this section, we consider the problem of systematic monitoring of passenger flows. In general, we can apply various face and body detectors to images for solution of this problem. However, low quality of data (see e.g. Figure 1) makes this task very difficult. To simplify this task, it is natural to use some method of tracking the image features. In particular, we can represent a sequence of features as a string.

We can consider strings of features of current and previous images and use longest common subsequence to establish a feature correspondence. However, successful feature tracking have different values for different types of features. In particular, features that extracted from the images of passengers have critical importance for solution of the problem of systematic monitoring of passenger flows. If we use classic longest common subsequences, then we may lose some important features (see e.g. Figure 2). In case of Figure 2, if we consider a classic longest common subsequence, then subsequence of features, which extracted from the back of the chair (white area), can absorb features of passenger. In this case, we lose corresponding passenger. Therefore, we use constrained longest common subsequences.

We consider image corners (see [40]), vertical edges, and color features (see [41]) as the set of features of the environment. Let

$$B_1 = \{b_{1,1}, b_{1,2}, \ldots, b_{1,\beta_1}\}$$

be an alphabet of image corners. Let

$$B_2 = \{b_{2,1}, b_{2,2}, \ldots, b_{2,\beta_2}\}$$

be an alphabet of vertical edges. Let

$$B_3 = \{b_{3,1}, b_{3,2}, \ldots, b_{3,\beta_3}\}$$

be an alphabet of color features. In this case,

$$B_1 \cup B_2 \cup B_3$$

is the alphabet of features of the environment.

We use Haar cascades (see e.g. [42], [43]) for initial detection of passengers. Haar cascades allow us to obtain a set of various features, parts of faces, parts of bodies, pieces of clothing and so on. We classify these features based on their motion. This classification allows us to select areas of

Fig. 2. An example of a loss of information.

Fig. 3. Areas of interest.

interest and identify these areas or sets of these areas as passengers (see e.g. Figure 3). After classification, we use unusual patterns and passenger color features as features for tracking.

Let

$$C_1 = \{c_{1,1}, c_{1,2}, \ldots, c_{1,\gamma_1}\}$$

be an alphabet of unusual patterns. Let

$$C_2 = \{c_{2,1}, c_{2,2}, \ldots, c_{2,\gamma_2}\}$$

be an alphabet of passenger color features.

Let f be a feature. The set of pixels of the feature f at time t we denote by $S_f(t)$. We consider

$$\big(\max_{(x,y)\in S_f(t)} x, \min_{(\max_{(x,y)\in S_f(t)} x, y)\in S_f(t)} y\big)$$

as the coordinates of the feature f at time t. Let

$$f\langle x(t), y(t)\rangle$$

be a feature f with coordinates

$$(x(t), y(t))$$

at time t. We assume that

$$f_1\langle x_1(t), y_1(t)\rangle < f_2\langle x_2(t), y_2(t)\rangle$$

if and only if

$$(x_1(t) < x_2(t)) \vee ((x_1(t) = x_2(t)) \wedge (y_1(t) > y_2(t))),$$

for $f_1, f_2 \in B_2 \cup C_1$. If $f_1 \notin B_2 \cup C_1$ or $f_2 \notin B_2 \cup C_1$, then

$$f_1\langle x_1(t), y_1(t)\rangle < f_2\langle x_2(t), y_2(t)\rangle$$

$$\Leftrightarrow$$

$$\begin{cases} (x_1(t-1) < x_2(t-1))\wedge \\ (x_1(t) < x_2(t) + \varepsilon), \\ ((x_1(t-1) = x_2(t-1))\wedge \\ (y_1(t-1) > y_2(t-1)))\wedge \\ (x_1(t) < x_2(t) + \varepsilon), \\ (x_2(t) \geq x_1(t) + \varepsilon), \end{cases}$$

where ε is a constant that depends on the resolution of the images. Under this assumption, we can construct the string

$$F(t) = f_{t,1}\langle x_{t,1}(t), y_{t,1}(t)\rangle f_{t,2}\langle x_{t,2}(t), y_{t,2}(t)\rangle \ldots$$

$$f_{t,r_t}\langle x_{t,r_t}(t), y_{t,r_t}(t)\rangle$$

of all features at time t.

TABLE I
THE AVERAGE NUMBER OF FRAMES DURING TRACKING

i	1	2	3	4	5	6	7	8	9	10	11	12
$nf_{LCS}(Set[i])$	19	16	10	7	4	3	24	20	11	8	5	4
$nf_{C-LCS}(Set[i])$	304	112	94	57	36	27	743	481	366	154	217	125

Let

$$P(t) = \{P_1(t), P_2(t), \ldots, P_\alpha(t)\}$$

be a set of passengers at time t. We create a set of strings of features of passengers. In particular, we assume that the string Z_j of features of $P_j(t)$ is the longest subsequence of $F(0)$ such that $F(0)[i] \in P_j(t)$, for all i and j.

We assume that

$$f_1\langle x_1(t), y_1(t)\rangle = f_2\langle x_2(t), y_2(t)\rangle$$

if and only if $f_1 = f_2$. For feature tracking, we consider strings $F(t-1)$ and $F(t)$ and the set of constraints

$$\{Z_j \mid 1 \leq j \leq \alpha\}.$$

If we can solve the constrained longest common subsequence problem, then we use constrained longest common subsequence to localize passengers. If we can not solve the problem, then we again use Haar cascades for initial detection of passengers and restart the process. Usage of constrained longest common subsequences allows us to minimize number of runs of very complicated process of classification.

In our experiments, we consider video files that have been received from one bus camera. We have considered the following parameters: resolution; infrared video and color video; number of passengers. We have created following data sets:

- $Set[1]$: resolution 1280×800,
infrared video,
number of passengers < 5;
- $Set[2]$: resolution 640×400,
infrared video,
number of passengers < 5;
- $Set[3]$: resolution 1280×800,
infrared video,
$5 \leq$ number of passengers < 10;
- $Set[4]$: resolution 640×400,
infrared video,
$5 \leq$ number of passengers < 10;
- $Set[5]$: resolution 1280×800,
infrared video,
number of passengers < 15;
- $Set[6]$: resolution 640×400,
infrared video,
number of passengers < 15;
- $Set[7]$: resolution 1280×800,
color video,
number of passengers < 5;
- $Set[8]$: resolution 640×400,
color video,
number of passengers < 5;
- $Set[9]$: resolution 1280×800,
color video,
$5 \leq$ number of passengers < 10;
- $Set[10]$: resolution 640×400,
color video,
$5 \leq$ number of passengers < 10;
- $Set[11]$: resolution 1280×800,
color video,
number of passengers < 15;
- $Set[12]$: resolution 640×400,
color video,
number of passengers < 15.

For any data set $Set[i]$, let $nf_{LCS}(Set[i])$ be the average number of frames before the loss of first passenger during longest common subsequence tracking, $nf_{C-LCS}(Set[i])$ be the average number of frames before the loss of first passenger during constrained longest common subsequence tracking. Selected experimental results are given in Table I.

V. MINING FOR INTERESTING PATTERNS

It is well-known that feature selection is one of the most important problems of image processing (see e.g. [44], [45]). A common technique for feature selection is the discovery of frequent patterns.

Note that we can use fluents [46] to express temporal patterns. This approach allow us to consider different string problems to mine interesting patterns. Since different versions of the longest common subsequence problem frequently used to mine interesting patterns (see e.g. [9], [11], [47]–[49]), it is natural to use C-LCS to mine interesting patterns.

Mining for interesting patterns has a number of applications in robot self-awareness (see e.g. [9], [11]). In particular, we need some system of prediction of collisions to build robot with ability to anticipate the motions (see e.g. [16], [50], [51]).

The c-fragment longest arc-preserving common subsequence problem (C-FLCS) and the problem of the longest common subsequence over the set (LCSS) were used to create sets of interesting patterns for prediction of collisions (see [9], [11]). These sets were used by recurrent neural network for prediction of collisions of mobile robot. It is clear that we can apply C-LCS to create a set of interesting patterns for prediction of collisions. Let t be the size of training set. Selected experimental results are shown in Table II.

TABLE II
THE QUALITY OF PREDICTION

t	10^2	10^3	10^4	10^5	10^6	10^7
C-LCS	95 %	98 %	98.9 %	99.2 %	99.5 %	99.6 %
C-FLCS	91 %	96 %	97 %	98 %	98.1 %	98.1 %
LCSS	76 %	83 %	88 %	96 %	96 %	96 %

TABLE III
EXPERIMENTAL RESULTS FOR DIFFERENT TEST SETS FOR MONITORING OF PASSENGER FLOWS

solver	test	average time	max time	best time
OA[1]	$Test[1]$	1.13 sec	2.19 min	0.04 sec
OA[2]	$Test[1]$	1.38 sec	1.28 min	0.06 sec
OA[3]	$Test[1]$	0.27 sec	42.14 sec	0.02 sec
OA[4]	$Test[1]$	0.08 sec	6.15 sec	0.012 sec
OA[5]	$Test[1]$	0.03 sec	3.19 sec	0.007 sec
GSAT	$Test[1]$	0.54 sec	1.17 min	0.05 sec
OA[1]	$Test[2]$	1.28 sec	2.43 min	0.12 sec
OA[2]	$Test[2]$	1.54 sec	1.57 min	0.09 sec
OA[3]	$Test[2]$	0.35 sec	47.32 sec	0.03 sec
OA[4]	$Test[2]$	0.24 sec	33.2 sec	0.021 sec
OA[5]	$Test[2]$	0.13 sec	19.8 sec	0.014 sec
GSAT	$Test[2]$	0.87 sec	59.13 sec	0.086 sec
OA[1]	$Test[3]$	9.19 sec	6.58 min	0.36 sec
OA[2]	$Test[3]$	12.63 sec	9.13 min	0.28 sec
OA[3]	$Test[3]$	7.14 sec	4.68 min	0.19 sec
OA[4]	$Test[3]$	3.13 sec	2.05 min	0.043 sec
OA[5]	$Test[3]$	2.52 sec	1.44 min	0.022 sec
GSAT	$Test[3]$	8.43 sec	5.12 min	0.121 sec
OA[1]	$Test[4]$	2.12 min	22.23 min	1.29 sec
OA[2]	$Test[4]$	4.44 min	27.15 min	2.16 sec
OA[3]	$Test[4]$	1.62 min	16.97 min	1.08 sec
OA[4]	$Test[4]$	56.2 sec	8.49 min	0.064 sec
OA[5]	$Test[4]$	42.8 sec	6.27 min	0.043 sec
GSAT	$Test[4]$	1.83 min	19.73 min	1.15 sec
OA[1]	$Test[5]$	25.02 min	4.82 hr	2.06 min
OA[2]	$Test[5]$	31.49 min	6.39 hr	38.77 sec
OA[3]	$Test[5]$	14.12 min	2.14 hr	19.7 sec
OA[4]	$Test[5]$	3.58 min	56.77 min	6.09 sec
OA[5]	$Test[5]$	2.16 min	47.2 min	0.6 sec
GSAT	$Test[5]$	15.88 min	3.03 hr	28.5 sec
OA[1]	$Test[6]$	4.16 hr	31.78 hr	6.91 min
OA[2]	$Test[6]$	1.92 hr	17.09 hr	8.11 min
OA[3]	$Test[6]$	9.18 hr	43.52 hr	53.69 sec
OA[4]	$Test[6]$	6.43 min	1.09 hr	18.05 sec
OA[5]	$Test[6]$	3.17 min	53.1 min	1.129 sec
GSAT	$Test[6]$	47.3 min	3.69 hr	1.1 min

VI. SAT SOLVERS FOR C-LCS-D

We use genetic algorithms OA[1] (see [52]), OA[2] (see [53]), OA[3] (see [54]), OA[4] (see [55]), and OA[5] (see [56]) for the satisfiability problem to obtain optimal solutions of C-LCS-D. Also, we have considered GSAT with adaptive score function (see [57]).

We have used heterogeneous cluster (500 calculation nodes, Intel Core i7). Each test was runned on a cluster of at least 100 nodes. Note that due to restrictions on computation time (20 hours) we used savepoints.

In our experiments, we use real world data for monitoring of passenger flows. In particular, we consider two test sets,

- $Test[1]$: average length of strings = 150,
  average number of constraints = 7;
- $Test[2]$: average length of strings = 200,
  average number of constraints = 15.

Also, we consider four synthetic test sets for monitoring of passenger flows,

- $Test[3]$: average length of strings = 150,
  average number of constraints = 7;
- $Test[4]$: average length of strings = 200,
  average number of constraints = 15;
- $Test[5]$: average length of strings = 1000,
  average number of constraints = 100;
- $Test[6]$: average length of strings = 6000,
  average number of constraints = 200.

Selected experimental results are given in Table III.

We have considered real world data for mining for interesting patterns (see [9]). Selected experimental results are given in Table IV.

TABLE IV
EXPERIMENTAL RESULTS FOR MINING FOR INTERESTING PATTERNS

solver	average time	max time	best time
OA[1]	47 sec	1.67 hr	4.8 sec
OA[2]	51 sec	1.83 hr	3.91 sec
OA[3]	45 sec	2.29 hr	7.53 sec
OA[4]	12 sec	19 sec	1.23 sec
OA[5]	4.2 sec	14.7 sec	3.6 sec
GSAT	49 sec	3.25 hr	1.82 sec

VII. A TASK-LEVEL ROBOT LEARNING FROM DEMONSTRATION

Robot task learning has received significant attention recently (see e.g. [58]). In particular, the longest common

Fig. 4. Robot Neato XV-11 with an onboard computer and a camera.

subsequence of the state sequences can be used for task generalization (see e.g. [59]). We can assume that the longest common subsequence of two demonstrations constitute the generalized task model. The other actions can be considered as alternative paths, noise, or alternative tasks. Also, the longest common subsequence of the state sequences can be used for task learning from demonstration. In particular, we can consider task learning with one training example prepared by a human. In this case, the robot's state sequences are processed to evaluate the robot's performance given the specific training example prepared by a human (see e.g. [60]). It is natural to use demonstrations of different simple tasks to learn a complex task. In this case, we need a common subsequence of two demonstrations of the complex task such that task models of simple tasks are subsequences of the common subsequence of two demonstrations. It is clear that we can use the constrained longest common subsequence for solution of this problem.

In our experiments, we consider Neato XV-11 [61] with an onboard computer and a camera (see Figure 4). We consider a simple genetic algorithm that evolves a population of sequences of motor primitives and tries to obtain a sequence of motor primitives for given trajectory. At first, we assume that we have only one human training example of some trajectory H. We consider the robot's state sequence R and use the length of the longest common subsequence of H and R as the value of the fitness function for R. This genetic algorithm we denote by $\mathcal{T}_1$. Also we consider genetic algorithm $\mathcal{T}_2$ where we assume that we have human training example of some trajectory H and two human training examples H_1 and H_2 of some parts of the trajectory. We consider the robot's state sequences R, R_1, and R_2 for H, H_1, and H_2. Let T_i be the longest common subsequence of H_i and R_i where $i \in \{1, 2\}$. Let T be the constrained longest common subsequence of H and R for $\{T_1, T_2\}$. In $\mathcal{T}_2$, we use the length of T as the value of the fitness function for R.

Let $N_i(n)$ be the average number of generations of $\mathcal{T}_i$ that needed to obtain $H = R$ for $|H| = n$. It is clear that we can use

$$N(n) = \frac{N_2(n)}{N_1(n)}$$

as a measure of the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$. Selected experimental results are given in Table V.

TABLE V
EXPERIMENTAL RESULTS FOR $\mathcal{T}_1$ AND $\mathcal{T}_2$

n	10^2	10^3	10^4	10^5
$N(n)$	0.43	0.37	0.12	0.03

It is easy to see that $\mathcal{T}_2$ gives us better results. However, for $\mathcal{T}_2$ we need additional human training examples. Now we consider the following genetic algorithm $\mathcal{T}_3$. We consider human training examples $H^1, H^2, \ldots, H^k$ for different tasks. We assume that $\mathcal{T}_3$ evolves a population of sequences of motor primitives and tries to obtain a set of sequences of motor primitives for trajectories $H^1, H^2, \ldots, H^k$. Let R^i be the robot's state sequence for the trajectory H^i. Let $T_{i,j}$ be the longest common subsequence of H^i, R^i, and H^j. Let T^j be the constrained longest common subsequence of H^j and R^j for $\{T_{i,j} \mid i \neq j\}$. In $\mathcal{T}_3$, use the length of T^j as the value of the fitness function for R^j. Let

$$M(n) = \frac{N_3(n)}{N_1(n)}$$

where $N_3(n)$ be the average number of generations of $\mathcal{T}_3$ that needed to obtain $H^j = R^j$ for $|H| = n$, $1 \leq j \leq k$. Selected experimental results are given in Table VI.

TABLE VI
EXPERIMENTAL RESULTS FOR $\mathcal{T}_1$ AND $\mathcal{T}_3$

n	10^2	10^3	10^4	10^5
$M(n)$	0.56	0.18	0.041	0.0082

It is clear that $\mathcal{T}_3$ demonstrates good performance and does not require additional human training examples. However, $\mathcal{T}_3$ can be used only in the case when we have many learning tasks.

VIII. Conclusion

In this paper, we have considered the constrained longest common subsequence problem for two strings and arbitrary number of constraints. In particular, we have considered applications of the constrained longest common subsequence problem for monitoring of passenger flows and task-level robot learning from demonstration.

We have proposed an explicit reduction from the constrained longest common subsequence problem to the satisfiability problem. Also, we have presented experimental results for different satisfiability algorithms. In particular, we have considered synthetic test sets and real world data for monitoring of passenger flows.

References

[1] V. Popov, "Multiple genome rearrangement by swaps and by element duplications," *Theoretical Computer Science*, vol. 385, no. 1-3, pp. 115-126, October 2007.

[2] V. Yu. Popov, "Computational complexity of problems related to DNA sequencing by hybridization," *Doklady Mathematics*, vol. 72, no. 1, pp. 642-644, July-August 2005.

[3] V. Popov, "The approximate period problem for DNA alphabet," *Theoretical Computer Science*, vol. 304, no. 1-3, pp. 443-447, July 2003.

[4] H. Bodlaender, R. Downey, M. Fellows, and H. Wareham, "The parameterized complexity of sequence alignment and consensus," *Theoretical Computer Science*, vol. 147, no. 1-2, pp. 31-54, August 1995.

[5] A. Gorbenko and V. Popov, "On the Longest Common Subsequence Problem," *Applied Mathematical Sciences*, vol. 6, no. 116, pp. 5781-5787, October 2012.
[6] D. Hirschberg, "Recent results on the complexity of common subsequence problems," in *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, 1983*, pp. 325-330.
[7] R. Irving and C. Fraser, "Two algorithms for the longest common subsequence of three (or more) strings," in *Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching, 1992*, pp. 214-229.
[8] J. Yang, Y. Xu, G. Sun, and Y. Shang, "A New Progressive Algorithm for a Multiple Longest Common Subsequences Problem and Its Efficient Parallelization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 862-870, May 2013.
[9] A. Gorbenko and V. Popov, "The c-Fragment Longest Arc-Preserving Common Subsequence Problem," *IAENG International Journal of Computer Science*, vol. 39, no. 3, pp. 231-238, August 2012.
[10] A. Gorbenko and V. Popov, "The Longest Common Parameterized Subsequence Problem," *Applied Mathematical Sciences*, vol. 6, no. 58, pp. 2851-2855, March 2012.
[11] A. Gorbenko, V. Popov, and A. Sheka, "Robot Self-Awareness: Temporal Relation Based Data Mining," *Engineering Letters*, vol. 19, no. 3, pp. 169-178, August 2011.
[12] A. Gorbenko and V. Popov, "Robot Self-Awareness: Usage of Co-training for Distance Functions for Sequences of Images," *Advanced Studies in Theoretical Physics*, vol. 6, no. 25, pp. 1243-1246, November 2012.
[13] A. Gorbenko and V. Popov, "Robot Self-Awareness: Occam's Razor for Fluents," *International Journal of Mathematical Analysis*, vol. 6, no. 30, pp. 1453-1455, March 2012.
[14] A. Gorbenko, V. Popov, and A. Sheka, "Robot Self-Awareness: Exploration of Internal States," *Applied Mathematical Sciences*, vol. 6, no. 14, pp. 675-688, January 2012.
[15] A. Gorbenko and V. Popov, "Anticipation in Simple Robot Navigation and Learning of Effects of Robot's Actions and Changes of the Environment," *International Journal of Mathematical Analysis*, vol. 6, no. 55, pp. 2747-2751, November 2012.
[16] A. Gorbenko and V. Popov, "The Force Law Design of Artificial Physics Optimization for Robot Anticipation of Motion," *Advanced Studies in Theoretical Physics*, vol. 6, no. 13, pp. 625-628, March 2012.
[17] A. Gorbenko and V. Popov, "Anticipation in Simple Robot Navigation and Finding Regularities," *Applied Mathematical Sciences*, vol. 6, no. 132, pp. 6577-6581, November 2012.
[18] A. Gorbenko and V. Popov, "Robot Self-Awareness: Formulation of Hypotheses Based on the Discovered Regularities," *Applied Mathematical Sciences*, vol. 6, no. 132, pp. 6583-6585, November 2012.
[19] A. Gorbenko and V. Popov, "Robot's Actions and Automatic Generation of Distance Functions for Sequences of Images," *Advanced Studies in Theoretical Physics*, vol. 6, no. 25, pp. 1247-1251, November 2012.
[20] G. M. Karthik and R. V. Pujeri, "Constraint Based Periodic Pattern Mining in Multiple Longest Common Subsequences," *Indian Journal of Science & Technology*, vol. 6, no. 8, pp. 5046-5055, August 2013.
[21] H. Kum, J. Chang, and W. Wang, "Sequential Pattern Mining in Multi-Databases via Multiple Alignment," *Data Mining and Knowledge Discovery*, vol. 12, no. 2-3, pp. 151-180, May 2006.
[22] A. Gorbenko and V. Popov, "Self-Learning Algorithm for Visual Recognition and Object Categorization for Autonomous Mobile Robots," *Lecture Notes in Electrical Engineering*, vol. 107, pp. 1289-1295, January 2012.
[23] Y.-T. Tsai, "The constrained longest common subsequence problem," *Information Processing Letters*, vol. 88, no. 4, pp. 173-176, November 2003.
[24] A. N. Arslan and Ö. Eğecioğlu, "Algorithms for the constrained longest common subsequence problems," *International Journal of Foundations Computer Science*, vol. 16, no. 6, pp. 1099-1109, November 2005.
[25] D. Becerra, W. Soto, L. Nino, and Y. Pinzon, "An algorithm for Constrained LCS," in *IEEE/ACS International Conference on Computer Systems and Applications, 2010*, pp. 1-7.
[26] S. Deorowicz, "Fast Algorithm for Constrained Longest Common Subsequence Problem," *Theoretical and Applied Informatics*, vol. 19, no. 2, pp. 91-102, February 2007.
[27] S. Deorowicz, "Bit-Parallel Algorithm for the Constrained Longest Common Subsequence Problem," *Fundamenta Informaticae*, vol. 99, no. 4, pp. 409-433, December 2010.
[28] S. Deorowicz and J. Obstoj, "Constrained Longest Common Subsequence Computing Algorithms in Practice," *Computing and Informatics*, vol. 29, no. 3, pp. 427-445, March 2010.
[29] Y. Francis, L. Chin, A. De Santis, A. L. Ferrara, N. L. Ho, and S. K. Kim, "A simple algorithm for the constrained sequence problems," *Information Processing Letters*, vol. 90, no. 4, pp. 175-179, May 2004.
[30] Z. Gotthilf and M. Lewenstein, "Approximating Constrained LCS," in *String Processing and Information Retrieval, 2007*, pp. 164-172.
[31] C. S. Iliopoulos and M. S. Rahman, "New efficient algorithms for the LCS and constrained LCS problems," *Information Processing Letters*, vol. 106, no. 1, pp. 13-18, March 2008.
[32] Z. Gotthilf, D. Hermelin, and M. Lewenstein, "Constrained LCS: Hardness and Approximation," in *Combinatorial Pattern Matching, 2008*, pp. 255-262.
[33] M. Jiang, "Approximability of Constrained LCS," in *Algorithms and Computation, 2010*, pp. 180-191.
[34] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.
[35] A. Gorbenko, M. Mornev, and V. Popov, "Planning a Typical Working Day for Indoor Service Robots," *IAENG International Journal of Computer Science*, vol. 38, no. 3, pp. 176-182, August 2011.
[36] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, "The Problem of Sensor Placement for Triangulation-Based Localisation," *International Journal of Automation and Control*, vol. 5, no. 3, pp. 245-253, August 2011.
[37] A. Gorbenko and V. Popov, "Programming for Modular Reconfigurable Robots," *Programming and Computer Software*, vol. 38, no. 1, pp. 13-23, January 2012.
[38] A. Gorbenko, V. Popov, and A. Sheka, "Localization on Discrete Grid Graphs," *Lecture Notes in Electrical Engineering*, vol. 107, pp. 971-978, January 2012.
[39] A. Gorbenko and V. Popov, "The set of parameterized k-covers problem," *Theoretical Computer Science*, vol. 423, no. 1, pp. 19-24, March 2012.
[40] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994*, pp. 593-600.
[41] P. Lamon, I. Nourbakhsh, B. Jensen, and R. Siegwart, "Deriving and matching image fingerprint sequences for mobile robot localization," in *IEEE International Conference on Robotics and Automation, 2001*, pp. 1609-1614.
[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001*, pp. 511-518.
[43] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *IEEE ICIP*, vol. 1, pp. 900-903, 2002.
[44] Y. Lin, M. D. McCool, and A. A. Ghorbani, "Time Series Motif Discovery and Anomaly Detection Based on Subseries Join," *IAENG International Journal of Computer Science*, vol. 37, no. 3, pp. 259-271, August 2010.
[45] P. Lin, N. Thapa, I. S. Omer, L. Liu, and J. Zhang, "Feature Selection: A Preprocess for Data Perturbation," *IAENG International Journal of Computer Science*, vol. 38, no. 2, pp. 168-175, May 2011.
[46] T. Oates and P. R. Cohen, "Learning planning operators with conditional and probabilistic effects," in *Proceedings of the AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems, 1996*, pp. 86-94.
[47] W.-H. Hsu, Y.-Y. Chiang, and J.-S. Wu, "IntegratingWeighted LCS and SVM for 3D Handwriting Recognition on Handheld Devices using Accelerometers," *WSEAS Transactions on Computers*, vol. 9, no. 3, pp. 235-251, March 2010.
[48] C. H. Park, J. W. Yoo, and A. M. Howard, "Transfer of skills between human operators through haptic training with robot coordination," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation, 2010*, pp. 229-235.
[49] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering Activities to Recognize and Track in a Smart Environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 527-539, April 2011.
[50] T. Kobayashi and S. Tsuda, "Sliding Mode Control of Space Robot for Unknown Target Capturing," *Engineering Letters*, vol. 19, no. 2, pp. 105-111, May 2011.
[51] O. A. Dahunsi and J. O. Pedro, "Neural Network-Based Identification and Approximate Predictive Control of a Servo-Hydraulic Vehicle Suspension System," *Engineering Letters*, vol. 18, no. 4, pp. 357-368, November 2010.
[52] A. Gorbenko and V. Popov, "SAT Solvers for the Problem of Sensor Placement," *Advanced Studies in Theoretical Physics*, vol. 6, no. 25, pp. 1235-1238, November 2012.
[53] A. Gorbenko and V. Popov, "Computational Experiments for the Problem of Selection of a Minimal Set of Visual Landmarks," *Applied Mathematical Sciences*, vol. 6, no. 116, pp. 5775-5780, November 2012.
[54] A. Gorbenko and V. Popov, "Task-resource Scheduling Problem," *International Journal of Automation and Computing*, vol. 9, no. 4, pp. 429-441, August 2012.

[55] V. Popov, "Genetic Algorithms with Exons and Introns for the Satisfiability Problem," *Advanced Studies in Theoretical Physics*, vol. 7, no. 8, pp. 355-358, January 2013.
[56] V. Popov, "A Genetic Algorithm with Expansion Operator for the 3-Satisfiability Problem," *Advanced Studies in Theoretical Physics*, vol. 7, no. 8, pp. 359-361, January 2013.
[57] V. Popov, "GSAT with Adaptive Score Function," *Advanced Studies in Theoretical Physics*, vol. 7, no. 8, pp. 363-366, January 2013.
[58] S. Ekvall and D. Kragic, "Robot Learning from Demonstration: A Task-level Planning Approach," *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223-234, March 2008.
[59] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: instructive demonstrations, generalization and practice," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems, 2003*, pp. 241-248.
[60] C. H. Park, J. W. Yoo, and A. M. Howard, "Transfer of Skills between Human Operators through Haptic Training with Robot Coordination," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation, 2010*, pp. 229-235.
[61] Neato Robotics web page. [Online]. Available: http://www.neatorobotics.com/

Anna Gorbenko was born on December 28, 1987. She received her M.Sc. in Computer Science from Department of Mathematics and Mechanics of Ural State University in 2011. She is currently a graduate student at Mathematics and Computer Science Institute of Ural Federal University and a researcher at Department of Intelligent Systems and Robotics of Mathematics and Computer Science Institute of Ural Federal University. She has (co-)authored 4 books and 90 papers.