

Разработан алгоритм H-метода для решения задачи оптимизации маршрутов для обслуживания множества клиентов с учетом временных ограничений несколькими гомогенными транспортными средствами с заданными емкостными ограничениями (задача VRPTW). Приведены результаты вычислительного эксперимента по сравнению предложенного алгоритма с рядом известных методов комбинаторной оптимизации, таких как детерминированный локальный поиск, имитационный отжиг, G-алгоритм.

© Л.Ф. Гуляницкий, А.В. Самусь,
2012
УДК 519.21

Л.Ф. ГУЛЯНИЦКИЙ, А.В. САМУСЬ

РЕШЕНИЕ H-МЕТОДОМ ЗАДАЧИ ОПТИМИЗАЦИИ МАРШРУТОВ ТРАНСПОРТНЫХ СРЕДСТВ С ВРЕМЕННЫМИ ОКНАМИ

Введение. Рассмотрена проблема обслуживания множества клиентов с учетом временных ограничений, выдвигаемых клиентами. Математическая модель – задача построения маршрута для транспортных средств (ТС) с временными окнами и заданными емкостными ограничениями, называемая Vehicle Routing Problem with Time Windows (VRPTW) [1–6]. Данная задача – практически важная и NP-сложная задача комбинаторной оптимизации.

Следует отметить, что VRPTW – это многокритериальная задача, в которой цель не только минимизация количества ТС используемых для транспортировки, а также и времени перевозок, их стоимости или общей длины маршрута. Поэтому используется иерархическая целевая функция.

Для ее решения предлагается алгоритм H-метода, поскольку его эффективность подтверждается опытом решения ряда комбинаторных задачах оптимизации на перестановках. Для построения схемы данного алгоритма используется одномаршрутное представление VRPTW; также предложен способ построения отрезков – одна из важных составляющих частей данного метода.

Для исследования эффективности разработанного алгоритма проведен вычислитель-

ный эксперимент, в котором результаты применения H -метода сравнивались с рядом известных алгоритмов – локального поиска, имитационного отжига, G -алгоритма.

1. Математическая модель задачи

VRPTW задается парком гомогенных ТС V , множеством клиентов C и направленным графом G . Граф состоит из $|C|+2$ вершин, где клиенты обозначены $1, \dots, n$, а склады представлены вершинами 0 («склад, из которого выезжают») и $n+1$ («склад, в который возвращаются»).

Обозначим множество $N=\{0,1,\dots,n+1\}$. Множество дуг (обозначим A) отвечает связям (расстояниям) между складом и клиентами, а также связям между клиентами. Никакая дуга не заканчивается в вершине 0 и никакая дуга не выходит из вершины $n+1$. С каждой дугой (i, j) , $i \neq j$, связывается стоимость c_{ij} и время t_{ij} , которое может включать время обслуживания клиента i .

Пусть каждое ТС имеет вместимость Q , и у каждого клиента i имеем спрос D_i . Пусть также каждый клиент i имеет временное окно $[a_i, b_i]$. ТС должно приехать к клиенту не позже b_i . Оно может приехать ранее a_i , но обслуживание клиента не будет начинаться раньше. Склад также имеет временное окно $[a_0, b_0]$ (предполагается, что временные окна для обоих складов идентичны): промежуток $[a_0, b_0]$ называется горизонтом планирования. ТС не могут выезжать из склада ранее a_0 и должны вернуться раньше или во время b_0 . Количество ТС (маршрутов) используемых в решении обозначим m .

Маршрут i -го ТС обозначает R_i – перестановка подмножества вершин, посещающих ТС. Время посещения k -го клиента i -м ТС обозначим s_{ik} . В случае, если ТС k не обслуживает клиента i , то s_{ik} ничего не означает. Принимается, что $a_0 = 0$, и поэтому $s_{0k} = 0$ для всех k .

Предполагается, что Q , a_i , b_i , D_i , c_{ij} – неотрицательные целые числа, тогда как t_{ij} считаются положительными числами. При применении точных методов обычно также принимается, что неравенство треугольника выполняется и для c_{ij} , и для t_{ij} [2].

Решением задачи является разбиение множества V на подмножества (маршруты) и задание порядка обхода вершин на каждом подмножестве (перестановка вершин маршрута). Решение задачи является допустимым, если все маршруты удовлетворяют дополнительным ограничениям задачи. Целевая функция – суммарная стоимость маршрутов в задаче [5, 6]:

$$F_{VRPTW} = \sum_{i=1}^m C(R_i) \quad (1)$$

при ограничениях

$$Q(R_i) \leq Q, \quad i = \overline{1, m} \quad (2)$$

$$a_i \leq s_{ik} \leq b_i, \quad \forall i \in N, \forall k \in V, \quad (3)$$

где $C(R_i)$ – стоимость маршрута R_i i -го ТС, а $Q(R_i)$ – вместительность ТС, необходимого для данного маршрута.

Условие (2) ограничивает грузоподъемность ТС, а (3) означает, что ТС обслуживается в заданном временном интервале. Остальные условия VRPTW удовлетворяются по построению маршрутов.

Для того чтобы сделать целевую функцию иерархической, т. е. чтобы сначала уменьшалось количество ТС, а затем расстояние, достаточно ко всем значениям c_{0j} , $j \in C$, прибавить достаточно большое $h_0 > 0$, что отвечает стоимости использования ТС.

Чтобы найти решение с заранее определенной нижней границей m для количества маршрутов (или эквивалентно – ТС), достаточно прибавить к множеству клиентов C и графу G фиктивных клиентов F (например, $C_{n+2}, \dots, C_{n+m+1}$), имеющих одинаковые временные окна $[a, b]$, $a \geq a_0$, $b > a_0$, и не имеющих спроса ($D_j = 0$, $j \in F$), а также положить

$$c_{ij} = c_{0j}, t_{ij} = t_{0j} + (b - a_0), \forall i \in F, j \in C, \quad (4)$$

$$c_{0j} = 0, t_{0j} = 0, j \in F, \quad (5)$$

$$t_{ij} = M' \quad \forall i \in F, \forall j \in F, \quad (6)$$

где M' – достаточно большое скалярное значение; потом увеличить все временные окна клиентов из C (кроме значения a_0) и ассоциируемые времена перемещения для дуг в графе G на величину $b - a_0$. Следовательно, клиенты из подмножества F будут обязательно обслуживаться разными ТС, что и обеспечит минимальное количество ТС, не меньше чем m .

VRPTW – комплексная задача [2]. Если снять временные ограничения, то VRPTW превратится в известную задачу VRP. Если задано больше двух депо, то образовывается задача Multiple Depot VRP. Если клиенты сосредоточены у каждого депо, то задача может быть разбита на несколько независимых задач. Однако, если клиенты и депо расположены произвольным образом, то нужно искать решение задачи маршрутизации транспорта с несколькими депо (MDVRP), где у каждого депо свой парк ТС.

Задача маршрутизации с возможностью возврата и доставки товаров (VRP with Pick-Ups and Deliveries) расширяет стандартную VRP тем, что клиенты в точках доставки могут возвращать некоторое количество товара обратно в депо. Таким образом, нужно быть уверенным, что товары, которые хочет вернуть клиент, не будут перегружать ТС. Данное ограничение делает планирование задачи более сложным и может привести к неэффективному использованию транспорта, увеличению стоимости маршрутов и общего количества ТС в депо.

VRPTW содержит как частные случаи несколько NP-сложных задач комбинаторной оптимизации (задача коммивояжера, задача о рюкзаке), из чего следует, что она также – NP-сложная. Более того, нахождение допустимого решения для задачи коммивояжера с временными окнами (TSPTW) является NP-полным [3]. Нахождение допустимого решения VRPTW с фиксированным количеством ТС – NP-сложное в сильном смысле [6]. Если количество доступных ТС неограничено, то проверка допустимости варианта решения, что состоит, например, из маршрутов «склад-и-склад», $\forall i \in C$, – простая задача, которая может быть решена за время $O(n)$.

2. Алгоритмы решения

Вычислительные схемы алгоритмов, основанных на локальном поиске, используют понятие окрестности в пространстве вариантов решений задачи X . Вариант решения состоит из множества маршрутов – последовательности номеров клиентов, посещаемых первым, вторым и т. д. ТС, каждый из которых ограничивается складом в начале и в конце. В работе [7] определено, что наиболее эффективными способами модификаций маршрутов оказались оператор замены двух ребер (названный далее оператором 2^* -замены) и оператор перемещения (relocate), впервые предложенный в [3]. Согласно этому будем различать и два способа порождения окрестностей $N(x)$ произвольной точки $x \in X$.

Для задач маршрутизации ТС возможны различные варианты представления решения.

1. Одномаршрутное представление.
2. Многомаршрутное представление.

Для примера пусть наше решение имеет два ТС. Первый посещает сначала клиента № 1, затем клиента № 3 и возвращается в депо, второй – клиент № 4, клиент № 2, клиент № 5 и возвращается в депо. Тогда одно маршрутное представление будет иметь вид: $\{1, 3, 0, 4, 2, 5\}$, а многомаршрутное представление:

$$\left\{ \left\{ 0, 1, 3, 0^* \right\} \right. \\ \left. \left\{ 0, 4, 2, 5, 0^* \right\} \right\}.$$

Представление второго вида удобно использовать для применения операторов 2^* -замены и оператора перемещения, построения решения с помощью конструктивных алгоритмов, для поиска решений имитационными алгоритмами.

В свою очередь представление первого типа приближает постановку задачи маршрутизации ТС к задаче коммивояжера и работой с перестановками. В данном случае есть возможность применить транспозиционную метрику и все наработки, касающиеся его.

***H*-метод.** Основная идея *H*-метода [8] заключается в том, что формируется начальная популяция – некоторое множество решений (рис. 1). С данного множества определенным образом выбираются несколько пар точек. Для каждой из них строится полуинтервал в пространстве вариантов решения задачи, который проходит через выбранные точки и на нем отыскивается лучшая по значению целевой функции точка. Найденный вариант передается как начальное приближение для улучшения траекторным алгоритмом (в нашем случае это *G*-алгоритм [9]). В результате таких действий находится совокупность точек, улучшающих решение. Данные точки включаются в очередную популяцию вместо худших по выбранному критерию точек и описанная процедура повторяется до выполнения выбранного критерия завершения.

```

procedure  $H$ -Algorithm( $x$ ) ;
  begin
     $h := 0; P^0 \neq \emptyset;$ 
    for  $j := 1$  to  $m$  do
       $x :=$  ГенерацияДопустимогоВарианта;
       $x := G\_Search(x);$ 
       $P^h := P^h \cup x;$ 
    endfor; { формирование начальной популяции}

  repeat
     $P := P^h;$ 
    for  $i := 1$  to  $k$  do
      ОтборДляВариации ( $x, y \in P: f(x) > f(y)$ );
       $z := \arg \min \{f(u) : u \in < x, x^\infty \setminus L(y)\}, y \in < x, x^\infty /$ ;
       $z := G\_Search(z);$ 
       $P := P^h \cup z;$ 
    endfor; { сформирована временная популяция из  $(m + k)$  точек}
    for  $i := 1$  to  $l$  do
      ОтборДляМутации ( $x \in P$ );
       $x :=$  Мутация ( $x$ );
       $x := G\_Search(x);$ 
       $P^h := P^h \cup x;$ 
    endfor; { сформирована временная популяция из  $m + k + l$  точек}
     $P^h :=$  ОтборПопуляции ( $P$ );
     $h := h + 1;$ 
  until условие завершения;
   $x := \arg \min \{f(y) : y \in P\};$ 
  return  $x$ ;
end

```

РИС. 1. Общая схема H -метода

Для построения полуинтервалов $< x, x^\infty /$ в рассматриваемой задаче маршрутизации ТС используется одномаршрутное представление и транспозиционная метрика. Отличие от представленного в [6] способа состоит в том, что использовать транспозицию для двух разделителей не разрешается.

G-алгоритм [9]. Принадлежит к классу алгоритмов стохастического локального поиска [10, 11] и имеет некоторые аналогии с методом имитационного отжига, отличаясь типом применяемой вероятностной машины (рис. 2).

```

procedure G(x)
begin
   $\mu := \mu_0$ ;
  while условия завершения не выполняются do
    y:= очередная точка окрестности N(x);
     $\Delta := f(x) - f(y)$ ;
    if  $\Delta > 0$  then x:= y
      else принять y в качестве x с вероятностью  $p(\mu, \Delta)$ 
    end if
    изменить  $\mu$ 
  end while
  return x
end

```

РИС. 2. Вычислительная схема G-алгоритма

3. Вычислительный эксперимент

Поскольку получение пригодных для практики оценок эффективности алгоритмов решения является проблематичным, обычно апробацию новых алгоритмов осуществляют путем решения задач VRPTW из известной библиотеки, условия и результаты решения которых представлены в [12].

Ввиду того, что *H*-метод относится к популяционным алгоритмам, то для него формировалась начальная популяция, а для всех остальных итерационных алгоритмов использовалась процедура рестарта, в которой в качестве начального приближения использовались указанная популяция. Данные варианты формировались двумя способами: путем применения конструктивного алгоритма эвристической вставки и с помощью датчика случайных чисел, с помощью которого строились допустимые решения. Популяция в *H*-методе состояла из 80 вариантов решения, а критерием останова было порождение 50 популяций. Целевая функция иерархическая: алгоритмы минимизировали количество ТС, а затем суммарную длину всех маршрутов.

Обобщенные результаты эксперимента приведены в таблице, где для каждой задачи показана (абсолютная) разность количества ТС лучшего варианта решения, найденного соответствующим алгоритмом в ходе эксперимента, а через знак "/" – относительная разность суммарного расстояния маршрутов (в %). Данная величина рассчитывалась так: $(f_a - f_{лучш}) / f_{лучш} \cdot 100$, где f_a – найденное соответствующим алгоритмом решение, $f_{лучш}$ – лучшее из известных приближенных решений поименованных в первом столбце задач из [12], поэтому найденные варианты решения, улучшающие длину маршрута имеют отрицательное значение. Сравнение осуществлялось с лучшим из известных решений, поскольку данные о точном решении были не всегда (см. вторую

колонку таблицы). Дополнительные обозначения: *Имя* – это идентификатор задачи из [12], *TP* – точное решение (если оно неизвестно, стоит прочерк), LS_{2^*} *замены* – локальный поиск с операцией 2^* -замены для построения окрестности, $LS_{relocate}$ – локальный поиск с операцией relocate для построения окрестности, *SA* – имитационный отжиг, *H* – *H*-метод, а *G* – *G*-алгоритм. В скобках указан способ порождения начальных приближений: *алг.* – с помощью эвристического алгоритма, *случ.* – с помощью генератора случайных вариантов. Жирным шрифтом выделены результаты либо совпадающие с точным решением, либо улучшающие длину наилучшего из известных вариантов.

ТАБЛИЦА

Имя	TP	LS_{2^*opt}		$LS_{relocate}$		SA		G	H	
		алг.	случ.	алг.	случ.	алг.	случ.	случ.	алг.	случ.
R101	1/–0.5	1/0.1	2/2.7	1/5.5	2/8.7	1/0.0	1/3.8	1/0.1	0/0.3	0/0.3
R102	1/–1.3	1/5.4	2/4.0	1/11.6	2/10.4	1/1.8	1/2.4	1/– 0.7	1/– 0.9	1/– 0.8
R103	1/–6.5	2/3.9	3/6.3	2/12.0	2/15.3	1/– 0.7	2/– 1.5	1/– 2.2	1/– 6.0	1/– 5.6
R104	2/–3.5	3/16.9	3/20.8	3/21.3	3/27.8	2/8.1	3/7.3	2/9.7	1/– 0.9	1/0.0
R105	1/–1.6	1/5.3	3/6.5	1/15.2	3/13.4	1/3.7	3/3.7	1/4.5	0/0.9	1/– 1.2
R106	1/–1.4	2/10.8	3/9.9	2/11.9	3/13.9	2/6.4	2/7.3	2/4.1	1/– 0.2	1/– 0.2
R107	1/–3.6	2/13.0	3/10.2	2/15.2	3/24.7	2/5.7	2/8.2	2/4.2	1/– 1.9	1/– 1.2
R108	–	2/22.1	3/15.9	2/29.0	2/34.4	1/11.7	2/4.0	2/6.5	1/– 1.2	1/– 1.0
R109	2/–4.0	2/9.9	3/9.6	2/19.7	4/14.0	2/4.7	3/3.4	2/0.4	1/– 3.4	1/– 2.8
R110	2/–4.5	3/13.0	4/14.2	3/18.7	4/24.2	2/6.7	2/10.8	2/3.8	2/– 2.2	1/– 0.9
R111	2/–4.4	2/18.0	3/15.7	2/37.4	3/16.2	2/5.2	3/6.1	2/4.1	1/– 0.7	1/– 1.7
R112	–	2/13.6	3/19.2	2/19.0	3/23.9	2/8.4	2/8.1	2/7.0	1/– 1.2	2/– 0.7
R201	4/–8.7	0/25.9	1/21.1	0/25.2	1/37.3	0/10.9	1/2.3	2/– 2.3	0/4.4	1/1.5
R202	–	1/9.2	2/13.0	1/13.9	2/22.3	1/– 3.2	2/– 5.5	2/– 9.0	1/– 8.1	2/– 12
R203	–	1/28.7	1/54.3	1/37.5	1/49.6	1/1.5	1/7.2	1/– 2.1	1/– 5.0	1/– 4.0
R204	–	1/22.4	2/21.4	1/28.3	2/36.7	1/1.6	1/9.3	1/– 5.1	1/– 7.5	1/– 7.7
R205	–	0/47.0	2/22.7	0/49.2	2/33.6	0/23.3	1/6.3	2/– 0.6	0/50	1/– 3.5
R206	–	0/27.3	1/32.9	0/30.1	1/59.8	0/11.2	1/6.7	1/9.1	0/11	1/– 1.8
R207	–	1/23.5	2/25.5	1/24.4	2/29.1	1/0.9	2/1.3	2/– 4.8	1/– 6.1	1/– 4.6
R208	–	1/25.0	1/41.6	1/31.6	2/42.1	1/3.2	1/6.5	1/3.1	0/7.3	1/– 2.8
R209	–	0/48.9	2/32.3	0/58.0	2/35.1	0/18.3	1/13.6	1/3.9	0/23	1/– 2.8
R210	–	1/23.3	1/52.5	1/31.7	1/53.0	1/5.0	1/12.1	1/4.2	1/– 1.5	1/– 0.7
R211	–	1/12.6	2/12.4	1/13.1	2/23.9	1/– 4.1	2/– 6.1	2/– 9.2	1/– 12	1/– 12
RC101	1/–4.5	3/2.5	4/4.7	2/3.6	4/15.0	1/0.1	3/0.9	3/2.4	1/– 2.4	2/– 2.0
RC102	2/–6.3	3/5.2	4/6.9	3/9.5	4/16.9	2/– 1.8	3/– 1.1	2/1.6	2/– 4.6	2/– 4.4
RC103	0/–0.3	2/15.8	3/19.7	2/17.7	3/32.7	2/10.3	2/10.6	1/11.2	1/2.0	1/4.8
RC104	–	1/32.1	2/22.4	2/22.6	3/30.3	1/12.6	1/12.8	2/7.9	1/1.3	1/1.4

Окончание таблицы

Имя	TP	LS _{2*opt}		LS _{relocate}		SA		G	H	
		алг.	случ.	алг.	случ.	алг.	случ.	случ.	алг.	случ.
RC105	2/-7.1	2/6.5	4/3.8	3/9.0	4/10.1	2/1.7	3/-1.2	3/-0.4	2/-2.6	3/-3.8
RC106	–	2/5.4	3/9.3	2/11.0	4/17.3	1/0.7	3/4.1	3/2.2	1/-0.5	2/-2.4
RC107	1/-1.8	2/14.2	2/20.4	2/20.0	3/41.5	1/8.7	1/6.1	2/8.2	1/-0.1	1/0.4
RC108	1/-2.2	2/20.3	3/22.4	2/21.9	3/30.1	2/8.3	2/10.8	2/10.4	1/-0.4	1/0.0
RC201	5/-10	0/25.1	2/10.9	0/30.2	2/22.4	0/18.3	2/-1.1	2/-5.6	0/38	1/-5.5
RC202	5/-20	1/18.2	2/27.3	1/21.2	2/54.3	1/-3.9	2/-5.6	2/-15	1/-13	2/-18
RC203	–	1/25.2	2/32.5	1/34.9	1/91.3	1/-0.4	2/-6.2	2/-9.3	1/-6.5	2/-10
RC204	–	0/56.0	1/37.4	0/66.2	1/86.2	0/25.9	1/5.1	1/4.6	0/0.2	1/-1.0
RC205	3/-11	0/43.0	2/17.3	0/49.1	2/25.9	0/27.1	2/-3.7	3/-8.0	0/63	2/-7.9
RC206	–	1/22.6	2/33.2	1/27.6	2/52.0	1/4.2	2/5.2	2/-2.7	1/-5.6	1/0.0
RC207	–	1/31.2	2/32.4	1/39.9	2/48.2	1/4.3	2/3.9	2/0.6	1/-5.2	1/-4.8

Как следует из результатов эксперимента, разработанный алгоритм *H*-метода продемонстрировал повышенные показатели точности по сравнению с алгоритмами как детерминированного локального поиска, так и алгоритмами имитационного отжига и *G*-алгоритма.

Л.Ф. Гуляницкий, О.В. Самусь

РОЗВ'ЯЗАННЯ *H*-МЕТОДОМ ЗАДАЧІ ОПТИМІЗАЦІЇ МАРШРУТІВ ТРАНСПОРТНИХ ЗАСОБІВ З ЧАСОВИМИ ВІКНАМИ

Розроблено алгоритм *H*-методу для розв'язання задачі оптимізації маршрутів для обслуговування множини клієнтів з урахуванням часових обмежень декількома гомогенними транспортними засобами із заданими ємкісними обмеженнями (задача VRPTW). Наведено результати обчислювального експерименту в порівнянні запропонованого алгоритму з низкою відомих методів комбінаторної оптимізації, таких як детермінований локальний пошук, імітаційний відпал, *G*-алгоритм.

L.F. Hulianytskyi, O.V. Samus

SOLVING VEHICLE ROUTING PROBLEM WITH TIME WINDOWS BY *H*-METHOD

It was developed an algorithm *H*-method for solving the optimization of routes to service a variety of clients in terms of time limits by several homogeneous vehicles with specified capacitive constraints (VRPTW). The results of computer simulation of the proposed algorithm compared with a number of well-known methods of combinatorial optimization, such as local search, simulation annealing, *G*-algorithm.

1. Bräysy O., Gendreau M. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms // *Transportation Science*. – 2005. – 39(1). – P. 104–118.
2. *The Vehicle Routing Problem: Latest Advantages and New Challenges* (Eds. B.Golden, S.Raghavan, E.A.Wasil). –Springer Science+Business Media, LLC. 2008. – 599 p.

3. *Larsen J.* Parallelization of the Vehicle Routing Problem with Time Windows // Ph.D. thesis. Institute of Mathematical Modelling, Technical University of Denmark. – Lyngby, 1999.
4. *Savelsbergh M.W.P.* Local Search for Routing Problems with Time Windows // *Annals of Operations Research*. – 1985. – 4. – P. 285–305.
5. *Potvin J.-Y., Rousseau J.-M.* An Exchange Heuristic for Routing Problems with Time Windows // *J. of the Operational Research Society*. – 1995. – 46. – P. 1433–1446.
6. *Гуляницкий Л.Ф.* Проблема оптимизации маршрутов транспортных средств с временными окнами // *Компьютерная математика*. – 2007. – № 1. – С. 122–132.
7. *Kohl N.* Exact methods for Time Constrained Routing and Related Scheduling Problems // PhD thesis, Department of Mathematical Modelling. – Technical University of Denmark, 1995.
8. *Гуляницкий Л.Ф., Сергиенко И.В.* Метаэвристический метод деформированного многогранника в комбинаторной оптимизации // *Кибернетика и системный анализ*. – 2007. – № 6. – С. 70–79.
9. *Гуляницкий Л.Ф.* Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования // *Компьютерная математика*. – 2004. – № 1. – С. 64–72.
10. *Hoos H.H., Stützle T.* Stochastic Local Search: Foundations and Applications. – San Francisco: Morgan Kaufmann Publ., 2005. – 658 p.
11. *Сергиенко И.В., Гуляницкий Л.Ф., Сиренко С.И.* Классификация прикладных методов комбинаторной оптимизации // *Кибернетика и системный анализ*. – 2009. – № 5. – С. 71–83.
12. <http://w.cba.neu.edu/~msolomon/problems.htm>

Получено 20.09.2012

Об авторах:

Гуляницкий Леонид Федорович,

доктор технических наук, заведующий отделом
Института кибернетики имени В.М. Глушкова НАН Украины,
leonhul.icyb@google.com

Самусь Александр Викторович,

магистр, аспирант Института кибернетики имени В.М. Глушкова НАН Украины.
samus.aleksandr@gmail.com