

РЕАЛИЗАЦИЯ ВЕНГЕРСКОГО И ВОЛНОВОГО АЛГОРИТМА В ПРОГРАММЕ STATEFLOW

Е.Е. Ростов

e.rostov@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Описан процесс переноса венгерского алгоритма целеуказания, а также волнового алгоритма маршрутизации в дискретном пространстве в среду блочного программирования StateFlow, являющуюся инструментальным пакетом для среды разработки MATLAB. Перечислены основные шаги построения венгерского алгоритма, представлены его математическая модель, целевая функция и блок-схема работы. Приведено описание волнового алгоритма с описанием его работы, а также целевая функция алгоритма. Показаны блоки программы StateFlow, описывающие работу алгоритмов. Сделан вывод о применении выбранных алгоритмов маршрутизации и целеуказания в области групповой робототехники.

Ключевые слова: StateFlow, MATLAB, алгоритмы целеуказания, венгерский алгоритм, алгоритмы маршрутизации, волновой алгоритм, групповая робототехника

Введение. Перспективным направлением робототехники является управление группой мобильных роботов. Это связано с тем, что создание многофункционального мобильного робота технически сложно и трудоемко, причем такой робот не сможет выполнять задачи, которые требуют работы в коллективе. Группа роботов, напротив, способна выполнять как задачи, требующие высокой точности, так и задачи, разделяемые на большое количество подзадач, которые наиболее эффективно и независимо друг от друга можно выполнить несколькими юнитами [1]. Для эффективного управления такими группами используют венгерский алгоритм маршрутизации.

Описание алгоритма целеуказания. Описание каждого алгоритма начинается с описания блок-схемы переходных состояний. Венгерский алгоритм — алгоритм оптимизации, решающий задачу о назначениях за полиномиальное время (время работы зависит от размера входных данных). Работу алгоритма можно разбить на 9 шагов [2].

Шаг 1. Найти наименьший элемент строки i .

Шаг 2. Вычесть найденный наименьший элемент из каждого элемента i -й строки.

Шаг 3. Найти наименьший элемент столбца j .

Шаг 4. Вычесть найденный наименьший элемент из каждого элемента j -го столбца.

Шаг 5. В результате выполнения шагов 1, 2, 3 получается редуцированная матрица.

Шаг 6. Заполнить векторы I, J соответственными нулевыми элементами редуцированной матрицы.

Шаг 7. Если в каждом столбце и в каждой строке найден нулевой элемент, тогда найдено совершенное паросочетание.

Шаг 8. Выбрать нулевые элементы так, чтобы никакие два нуля не находились в одном и том же столбце или строке. Данные элементы входят в паросочетание.

Шаг 9. Выполнять шаги с 1 по 8, пока не выполнится шаг 6.

На вход алгоритма поступает матрица размера $i \times j$, где i — номер кандидата, j — номер вакансии.

Математическая модель задачи имеет вид

$$F = \sum \sum C_{ij} x_{ij},$$

при следующих условиях:

$$\sum x_{ij} = n, i = 1, 2, \dots, m,$$

$$\sum x_{ij} = m, j = 1, 2, \dots, n,$$

$$x_{ij} \geq 0, \text{ целые.}$$

Переменные x_{ij} принимают значения 1, если i -й кандидат занимает j -ю вакансию. Если данное условие не выполняется, то $x_{ij} = 0$ [3].

Ограничения по кандидатам:

$$x_{11} + x_{12} + x_{13} + x_{14} = 1;$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 1;$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 1;$$

$$x_{41} + x_{42} + x_{43} + x_{44} = 1.$$

Ограничения по вакансиям:

$$x_{11} + x_{21} + x_{31} + x_{41} = 1;$$

$$x_{12} + x_{22} + x_{32} + x_{42} = 1;$$

$$x_{13} + x_{23} + x_{33} + x_{43} = 1;$$

$$x_{14} + x_{24} + x_{34} + x_{44} = 1.$$

Целевая функция:

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{21} + x_{22} + x_{23} + x_{24} + x_{31} + x_{32} + x_{33} + x_{34} + x_{41} + x_{42} + x_{43} + x_{44} \rightarrow \min.$$

Описывая вышесказанное, составим блок-схему, представляющую последовательность операций при работе венгерского алгоритма (рис. 1).

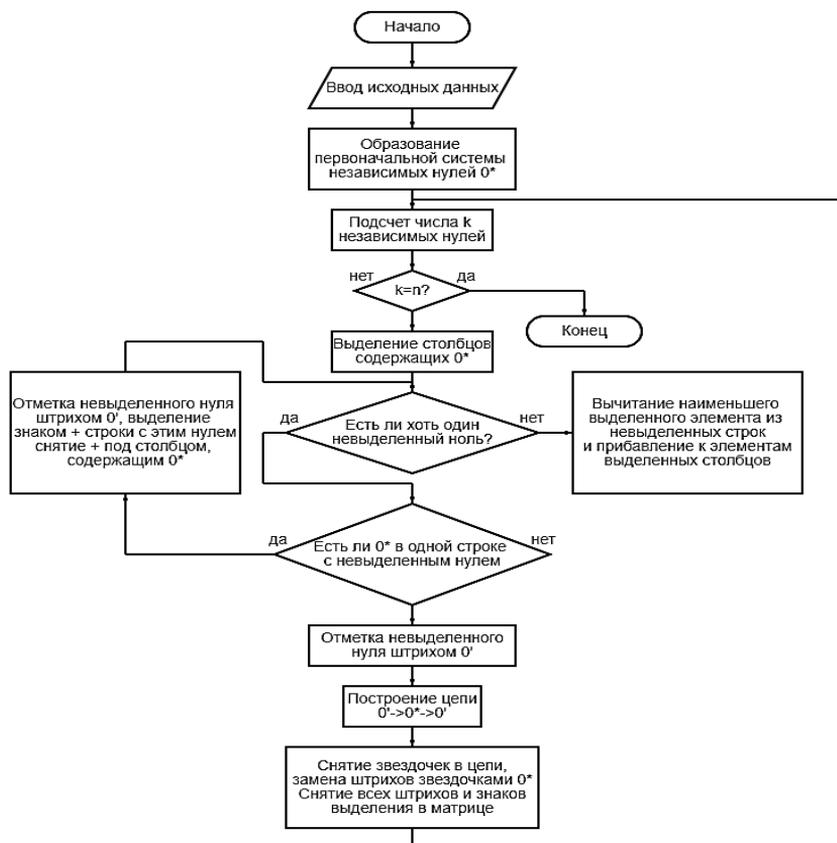


Рис. 1. Блок-схема работы венгерского алгоритма

Алгоритм выстраивает весовую матрицу таким образом, что для каждой строки кандидатов остается одно единственное решение из столбца вакансий, которое и будет являться наименее энергозатратным для исполнения именно этим роботом.

Программная реализация. Программное описание алгоритма реализуется благодаря использованию блоков пакета Stateflow. Данное расширение позволяет описать последовательность выполнения задания, а также вызвать прерывания, необходимые для корректной работы алгоритма на любом этапе выполнения задания [4]. Так, если роботы находятся на одинаковом расстоянии от выделенного им груза, для одного из роботов данная задача будет отменена и назначена другая. Также в случае, если пока участник группы пере-

мещается до цели, другой робот, выполнивший свою задачу, оказывается ближе, алгоритм передает задачу последнему. Пакет Stateflow делает возможным вызов подпрограмм, записанных внутри блока, в любой момент выполнения программы, что делает выполнение алгоритма «бесшовным». Описательные блоки Stateflow показаны на рис. 2.

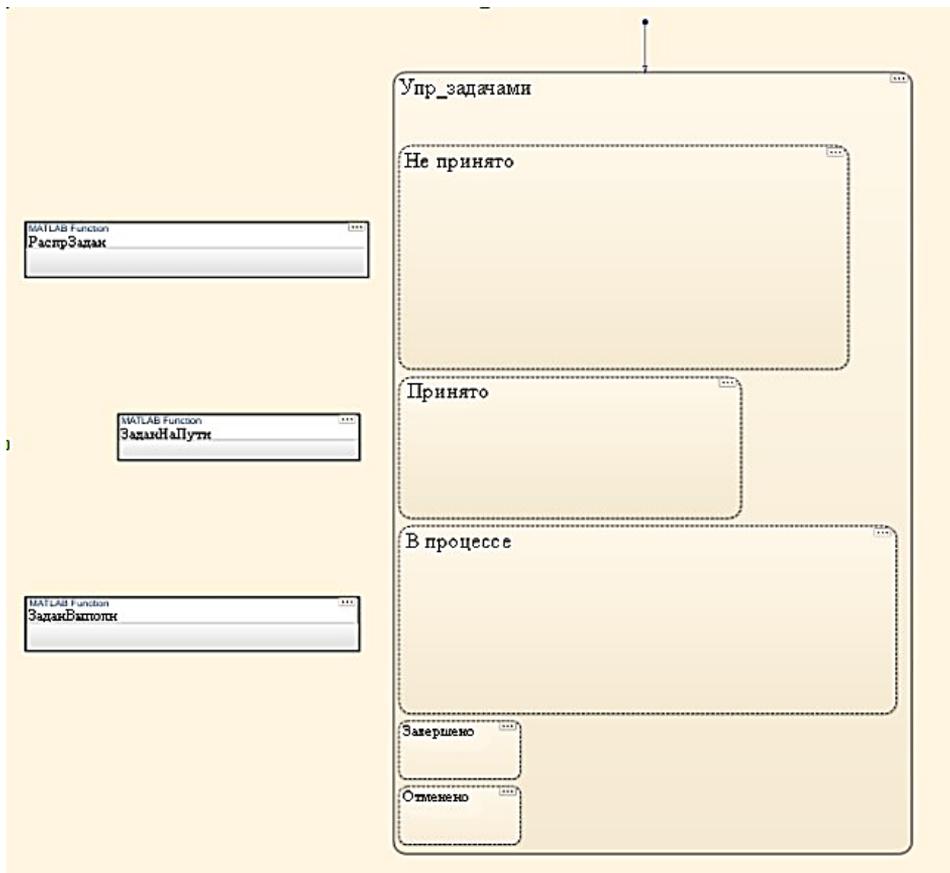


Рис. 2. Распределение задач в Stateflow

Данные для весовой матрицы поступают с обработчика, который определяет количество задач и юнитов, а также присваивает уникальный номер каждому роботу и заданию для определения статуса выполнения задания или занятости робота. Описательный блок представлен на рис. 3.

В свою очередь, вся информация по количеству роботов, заданий, а также окружению поступает с блоков, считывающих данную информацию с динамической карты местности.

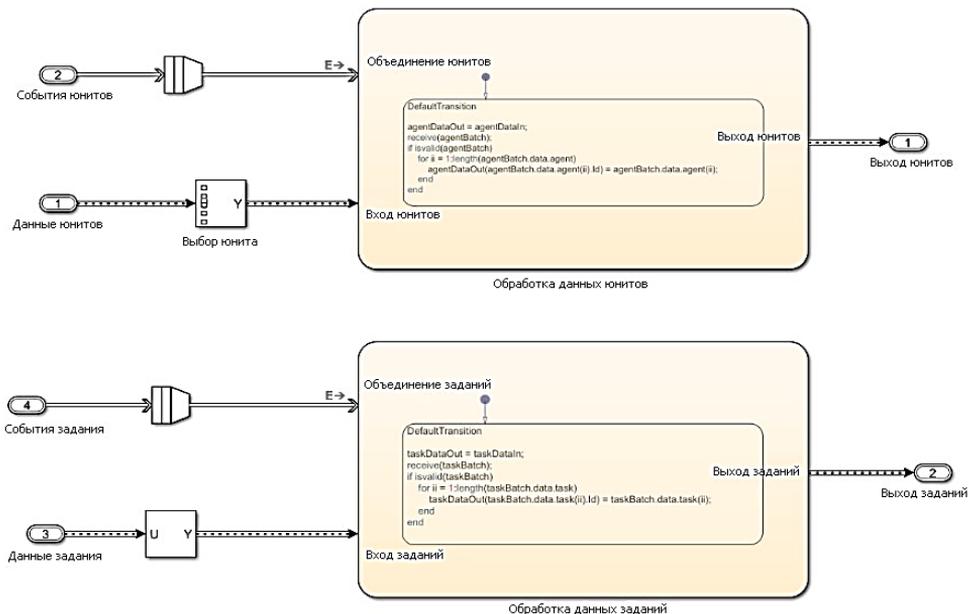


Рис. 3. Задание данных для матрицы

Описание алгоритма маршрутизации. Математическая модель алгоритма. В ходе научно-исследовательской работы был выбран алгоритм маршрутизации, названный алгоритмом Ли, или волновой алгоритм. Именно второе название в лучшей мере описывает метод, по которому работает алгоритм. Алгоритм применяется для нахождения кратчайшего пути в графе. Положение исходной и целевой точки в этом случае определено, от начальной точки исходят «волны», при этом вес каждой следующей «волны» больше предыдущей. Для препятствий вес назначается максимальным, таким образом «волна» определяет их и обходит [5]. Таким образом выстраивается единственный кратчайший путь от старта до цели. Пример работы алгоритма изображен на рис. 4.

Маршрутный алгоритм можно построить на основе следующего рекуррентного соотношения:

$$y(x) = 2y(x + h) + y(x + 2h) + d,$$

где x , $y(x)$ — абсцисса и ордината элемента занимаемого трассой на данном шаге; $(x + h)$ — ордината элемента занимаемого трассой на предыдущем шаге; $(x + 2h)$ — ордината элемента отстоящего от вычисляемого на два шага; h — изменение абсциссы на каждом шаге; d (delta) — функция, определяющая вид

трассы: если $d = 0$, то строится прямолинейная трасса, если $d = \text{const}$ — параболическая.

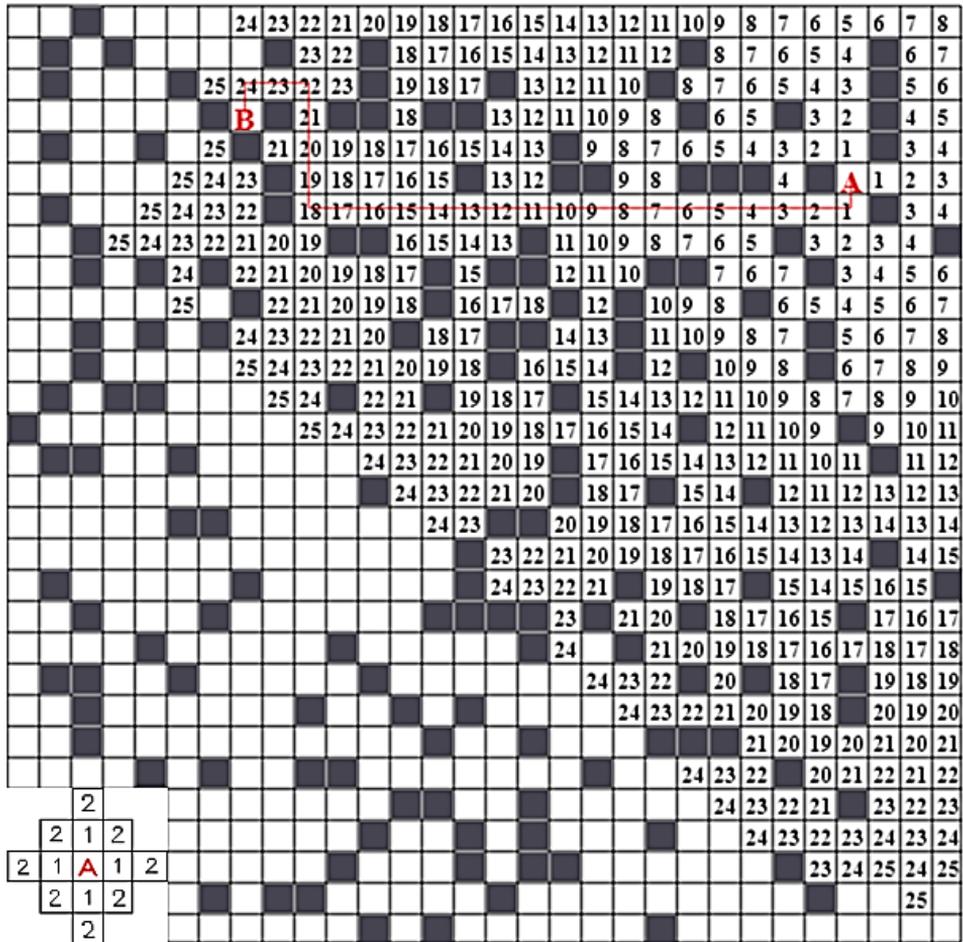


Рис. 4. Представление работы алгоритма Ли

Ордината очередного элемента трассы вычисляется по рекуррентной формуле, а абсцисса трассы — по формуле [6]

$$D = X_n = X_{n-1} + h.$$

Знак плюс или минус в рекуррентной формуле выбирают исходя из того, откуда начинается построение трассы: из начального элемента — плюс, из конечного — минус.

Чтобы вычислить третий элемент трассы по этой формуле, необходимо знать два предыдущих. Первым элементом является исходный элемент $A(XA, YA)$, тогда ордината второго элемента вычисляется по формуле [5]

$$Y(X) = Y(XA) + ((Y(XA) - Y(XB))/(XA - XB)) \times h.$$

Программная реализация. Программная реализация алгоритма разбивается на множество подзадач, таких как дискретизация пространства, определение местоположения роботов в момент времени, разрешения конфликтных ситуаций и прописывания нескольких сценариев перемещения [7]. Для работы алгоритма использовалось множество блоков. Начнем с описания программного блока алгоритма.

В блоке пакета Stateflow была написана основная программа, представляющая собой простой волновой алгоритм. Программа разделяется на три основных части. Преобразование окружения в граф, поиск начальной и конечной позиции, прокладка пути. В случае если путь заблокирован другим роботом, вводится прерывание для поиска другого доступного пути. Управляющий блок, составленный в пакете Stateflow, можно увидеть на рис. 5.

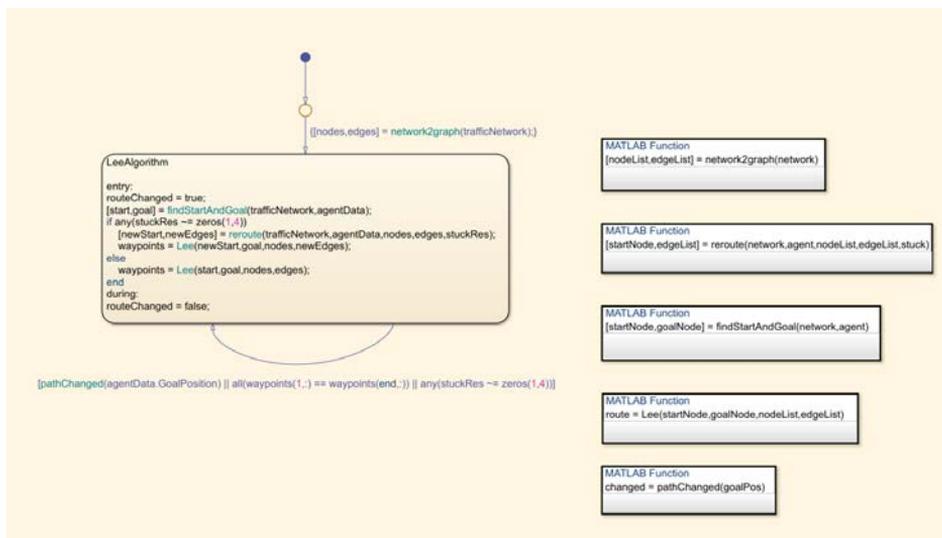


Рис. 5. Алгоритм Ли, реализованный на базе Stateflow

Основной конфликтной ситуацией в рассматриваемой задаче является столкновение двух участников коллектива [8]. Чтобы избежать ущерба, а также устранить ошибки в исполнении алгоритма, вводят программу разрешения конфликта (рис. 6).



Рис. 6. Программа разрешения конфликта столкновения

Работа программы представляется весьма простой, когда траектории перемещения двух роботов накладываются друг на друга, вводится прерывание, останавливающее одного из роботов, таким образом, превращая его в статическое препятствие [9]. Второй робот в это время начинает просчитывать новую траекторию перемещения для обхода этого препятствия [10].

Заключение. В ходе работы был реализован венгерский алгоритм целеуказания, волновой алгоритм маршрутизации в среду блочного программирования StateFlow, являющейся инструментальным пакетом для среды разработки MATLAB. Данные алгоритмы отлично подходят для применения при управлении группой роботов на ровных поверхностях в заранее известном окружении, например, в складских помещениях.

Литература

- [1] Пшихопов В.Х. *Групповое управление подвижными объектами в неопределенных средах*. Москва, Физматлит, 2015, 305 с.
- [2] Каляев И.А., Гайдук А.Р., Капустян С.Г. *Модели и алгоритмы коллективного управления в группах роботов*. Москва, Физматлит, 2009, 280 с.

- [3] Михайлов Б.Б., Назарова А.В., Ющенко А.С. Автономные мобильные роботы – навигация и управление. *Известия ЮФУ. Технические науки*, 2016, № 2 (175), с. 48–67.
- [4] Успенский В.А. *Что такое нестандартный анализ?* Москва, Наука. Главная редакция физико-математической литературы, 1987, 128 с.
- [5] Bellman R. On a Routing Problem. *Quarterly of Applied Mathematics*, 1958, vol. 16, no. 1, pp. 87–90.
- [6] Dijkstra E.W. A note on two problems in connection with graphs. *Numer. Math.*, 1959, vol. 1, iss. 1, pp. 269–271.
- [7] Ford L.R.Jr., Fulkerson D.R. *Flows in Networks*. Princeton, Princeton University Press, 1962.
- [8] Hart P.E., Nilsson N.J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968, no. 2, pp. 100–107. <https://doi.org/10.1109/TSSC.1968.300136>
- [9] Hart P.E., Nilsson N.J., Raphael B. Correction to “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. *SIGART Newsletter*, 1972, vol. 37, pp. 28–29. <https://doi.org/10.1145/1056777.1056779>
- [10] Lee C.Y. An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers*, 1961, vol. EC-10, no. 2, pp. 346–365.

Поступила в редакцию 25.04.2024

Ростов Евгений Евгеньевич — студент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Рубцов Василий Иванович, кандидат технических наук, доцент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Ростов Е.Е. Реализация венгерского и волнового алгоритма в программе StateFlow. *Политехнический молодежный журнал*, 2024, № 03 (92).

URL: <https://ptsj.bmstu.ru/catalog/menms/robots/981.html>

HUNGARIAN AND WAVE ALGORITHMS REALIZATION IN THE STATEFLOW PROGRAM

E.E. Rostov

e.rostov@mail.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

The paper describes the process of transferring the Hungarian target allocation algorithm, as well as the discrete space wave routing algorithm, to the StateFlow block-programming environment, which is a toolkit in the MATLAB development environment. It lists main steps in constructing a Hungarian algorithm and provides its mathematical model, objective function, and operation flowchart. The paper presents the wave algorithm and its operation description, as well as the algorithm objective function. StateFlow program blocks identifying the algorithms operation are shown. A conclusion is made on application of the selected routing and target allocation algorithms in the group robotics.

Keywords: StateFlow, MATLAB, target allocation algorithms, Hungarian algorithm, routing algorithms, wave algorithm, group robotics

Received 25.04.2024

Rostov E.E. — Student, Department of Robotic Systems and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Rubtsov V.I., Ph. D. (Eng.), Associate Professor, Department of Robotic Systems and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Rostov E.E. Hungarian and wave algorithms realization in the StateFlow program. *Politekhnikheskiy molodezhnyy zhurnal*, 2024, no. 03 (92). (In Russ.).

URL: <https://ptsj.bmstu.ru/catalog/menms/robots/981.html>