

УДК 004.94

DOI: 10.18384-2310-7251-2019-1-107-118

МОДЕЛИРОВАНИЕ ПЕРЕМЕЩЕНИЯ ОБЪЕКТА СЛОЖНОЙ НЕИЗМЕНЯЕМОЙ КОНФИГУРАЦИИ В ОГРАНИЧЕННОМ ПРОСТРАНСТВЕ

Акбаров Э. А., Калашников Е. В.

Московский государственный областной университет

141014, Московская область, г. Мытищи, ул. Веры Волошиной, д. 24

Российская Федерация

Аннотация. Анализируется задача перемещения и встраивания объекта неизменяемой конфигурации в ограниченном пространстве. Строится алгоритм и программа, имитирующая поведение такого объекта, на примере перемещения «коня» по шахматной доске. Находятся оптимальные пути перемещения объекта неизменяемой формы в нужную клетку доски и минимальное время для преодоления всего пути и встраивания в нужную точку.

Ключевые слова: имитационное моделирование, неизменяемая сложная конфигурация, ограниченное пространство, алгоритм перемещения.

SIMULATION OF MOTION OF AN OBJECT WITH IMMUTABLE COMPLEX CONFIGURATION IN A CONFINED SPACE

E. Akbarov, E. Kalashnikov

Moscow Region State University

ul. Very Voloshinoy 24, 141014 Mytishchi, Moscow Region, Russian Federation

Abstract. The problem of moving and embedding an object of immutable form in a confined space is analyzed. An algorithm and a program simulating the behavior of such an object are constructed on the example of moving a 'knight' on a chessboard. It is shown that there exist optimal ways to move an object of immutable form in the required chessboard cell and a minimum time to overcome all the way and embed it in the desired point.

Keywords: simulation modeling, immutable complex configuration, confined space, movement algorithm.

Введение

Задача транспорта объектов (тел) сложной неизменяемой конфигурации возникает в разных областях современной науки. В частности, в задачах редактирования ДНК молекулы [1; 2] необходимо доставлять в определённое место этой молекулы и в определённое время вполне определённого типа (определённой конфигурации и определённых размеров) молекулу белка. При этом транспортируемая молекула в процессе встраивания может поджиматься, вращаться, но, встроившись, принимает форму, допустимую молекулой ДНК. Другая такая

проблема возникает в задачах выращивания (например, полупроводниковых) гетероструктур наномасштабов. Здесь важно, чтобы в процессе роста двумерной структуры в перемещающийся фронт вовремя подходили атомы определённой (электронной SP или SPD) конфигурации и могли встроиться в нужное место фронта для того, чтобы гарантировать требуемую зонную структуру [3]. Но приведённые задачи, кроме того, включают в себя ещё одну сложность. Она связана с тем, что в этих задачах необходимо перемещать и встраивать молекулы или атомы на ограниченном пространстве. Другими словами, выстраивание новой структуры (редактирование ДНК молекулы или рост гетероструктуры) связано с переносом молекулы или атома определённой конфигурации и размеров, имеющих неизменяемую (или мало изменяемую) форму в ограниченном пространстве, и их встраиванием в объекты с предписанной конфигурацией. Наконец, существует острая социальная проблема – адаптация личности в коллективе [4–7]. В этом случае вместо пространственной неизменяемой конфигурации возникает набор состояний или принципов отношения личности к окружению. Набор таких принципов для личности может служить аналогом неизменяемой конфигурации. Задачи сложные и требуют, на первый взгляд, всякий раз своего индивидуального подхода. Тем не менее, возможен общий подход к выше приведённой проблеме, основанный на имитационном моделировании.

Имитационное моделирование – это метод исследования, в ходе которого изучаемая система заменяется описывающей её моделью. Подобный подход является эффективным средством исследования сложных систем, параметры которых изменчивы во времени [8]. В данной работе рассматриваются системы, основным аспектом поведения в которых является перемещение в пространстве одного объекта, иные аспекты не рассматриваются.

Перемещение объекта сложной неизменяемой конфигурации в ограниченном пространстве – проблема, заключающаяся в том, что для расчёта наискорейшего достижения объектом заданной точки необходимо, помимо ограничений пространства, учитывать строго заданную конфигурацию объекта, накладывающую дополнительные ограничения на возможности его перемещения [9].

Моделирование объекта сложной неизменяемой конфигурации в ограниченном пространстве (ОСНКОП) будет рассмотрено на примере движения шахматного коня по шахматной доске из одной заданной точки (клетки) в другую. Доска выступит ограниченным пространством, а необходимость делать ход конём буквой «Г» – его неизменяемой конфигурацией. Задача перемещение коня по доске в заданную точку (клетку) оказывается более сложной задачей по сравнению с вышеперечисленными. Поскольку здесь рассматривается «жёсткая» двумерная задача: конь (его буква «Г») не может быть поджат, перевёрнут вокруг своей оси – то это сложная двумерная, неизменяемая фигура перемещается в ограниченном двумерном пространстве в заданную его клетку [10; 11]. Таким образом, задачей данной работы является исследование перемещения двумерного объекта сложной неизменяемой конфигурации из одной точки (клетки) в другую, предписанную точку (клетку) ограниченного двумерного пространства.

1.1. Постановка задачи

Для поиска решения поставленной задачи создаём программу, имитирующую движение коня по доске на языке программирования PascalABC.

Алгоритм действий:

Пользователем вводятся координаты начальной и конечной точек маршрута, а также координаты клетки, занятой другой фигурой:

- конь движется по доске, совершая ходы в случайном порядке;
- дойдя до указанной клетки, конь останавливается;
- конь не ходит на занятую другой фигурой клетку.

Функционал программы: вывод на экран условного шахматного поля с отметками точек, координаты которых вводит пользователь. Построение маршрута движения коня по доске.

Для написания кода используется PascalABC.Net.

1.2. Маршрут движения шахматного коня

Конь может пойти на одно из полей, ближайших к тому, на котором он стоит, но не на той же самой горизонтали, вертикали или диагонали, то есть он ходит кириллической буквой «Г» (или латинской «L»).

Всего конь имеет 8 различных вариантов хода [12] (рис. 1.2.1).

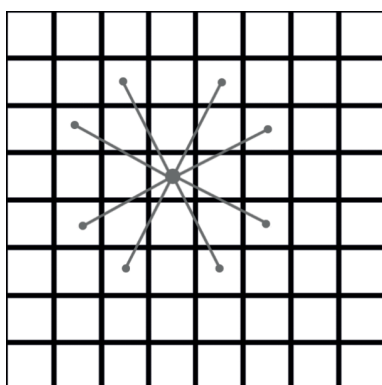


Рис. 1.2.1. Доступные шахматному коню варианты движения.

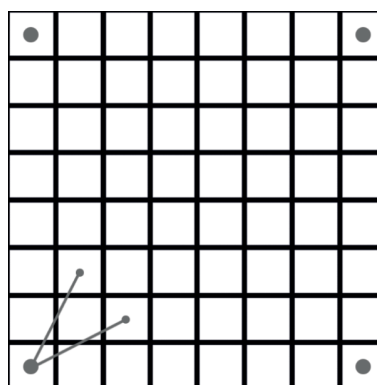


Рис. 1.2.2. Клетки с двумя вариантами движения.

Из угловых клеток ему доступно всего два варианта движения (рис. 1.2.2.).

Из клеток, прилегающих к одной из сторон и отстоящих от другой на одну клетку, – три варианта движения (рис. 1.2.3).

Из клеток, отступающих от обеих сторон на одну клетку, – четыре варианта движения (рис. 1.2.4).

Из клеток, отступающих от одной стороны на одну клетку и от другой на две, – шесть вариантов движения (рис. 1.2.5).

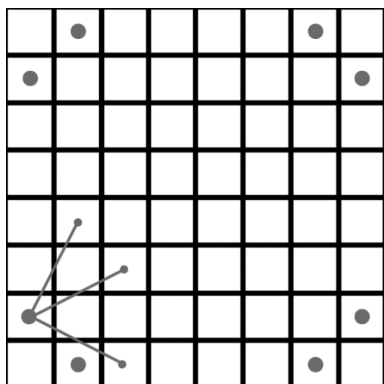


Рис. 1.2.3. Клетки с тремя вариантами движения.

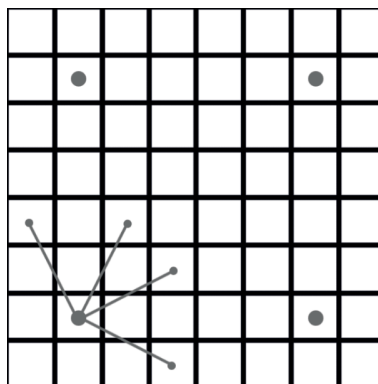


Рис. 1.2.4. Клетки с четырьмя вариантами движения.

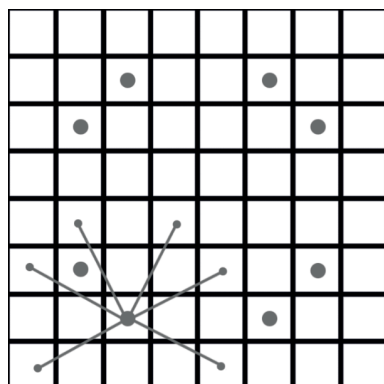


Рис. 1.2.5. Клетки с шестью вариантами движения.

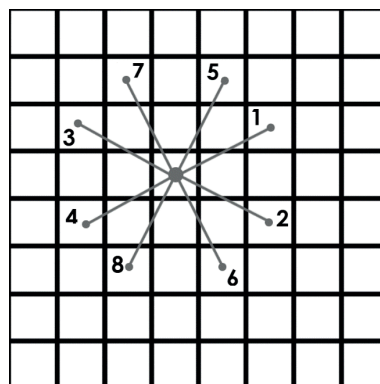


Рис. 2.1.1. Обозначение соотношений ходов коня.

Из всех остальных клеток доски доступно восемь вариантов движения.

2.1. Постановка условий движения

1. Движение коня происходит в случайном порядке, выбирается один из восьми вариантов движения.

2. Конь не должен выходить за границы «доски» (8x8 клеток).

3. Конь не может двигаться назад на ту клетку, с которой только что пришёл, чтобы избежать топтания на месте [13]. Соотношения ходов на одной линии (рис. 2.1.1):

1. 1-4, 4-1

2. 2-3, 3-2

3. 5-8, 8-5

4. 6-7, 7-6

4. Конь не должен останавливаться на клетке, которая занята другой фигурой. Если выбранная случайно клетка оказалась занята, конь выбирает новую.

5. Если необходимая клетка находится в пределах досягаемости одного хода, фигура сразу переходит на неё.

2.2. Написание кода программы

1. Подключается модуль *graphABC*, задаются все необходимые переменные.

```
usesgraphABC;
vari, n, m, wst, hst, lm:integer;
xS, yS, xs1, ys1, xs2, ys2, xs3, ys3, a, ci: real;
check: boolean;
```

2. Задаются размеры окна и одной клетки доски, создаётся само окно и его разметка.

```
begin
randomize;
wst:=600;
hst:=wst;
SetWindowSize(wst, hst);
setpencolor(clblack);
setpenwidth (1);
xS:=0;
yS:=0;
a:=wst/8;
ci:=0;
fori := 1 to 9 do
begin
line (round(xS), round(yS+(a*(i-1))), round(xS+a*8), round(yS+(a*(i-1))));
line (round(xS+a*(i-1)), round(yS), round(xS+a*(i-1)), round(yS+a*8));
end;
```

3. Пользователем вводятся координаты начальной и конечной точек, а также координаты занятой клетки. Программа не продолжит выполнение, пока введённые координаты не будут больше или равны 1 и меньше или равны 8. Результаты отображаются в созданном ранее окне.

```
setpencolor(clred);
setpenwidth (12);
repeat
writeln ('Введите координаты начальной точки');
readln (n, m);
until (n>=1) and (n<=8) and (m>=1) and (m<=8);
xs1:=a*n-a/2;
ys1:=yS+a*m-a/2;
```

```

Circle(round(xs1), round(ys1), 5);
Moveto(round(xs1), round(ys1));
setpencolor (clblue);
repeat
writeln ('Введите координаты конечной точки');
readln (n, m);
until (n>=1) and (n<=8) and (m>=1) and (m<=8);
xs2:=a*n-a/2;
ys2:=a*m-a/2;
Circle(round(xs2), round(ys2), 5);
repeat
writeln ('Введите координаты занятой клетки');
readln (n, m);
until (n>=1) and (n<=8) and (m>=1) and (m<=8);
xs3:=xs+a*n-a/2;
ys3:=ys+a*m-a/2;
FloodFill(round(xs3), round(ys3), clblue);
setpencolor(clblack);
setpenwidth (1);

```

4. Для построения пути фигуры используется цикл *repeat*, который будет выполняться до тех пор, пока координаты положения фигуры $(xs1, ys1)$ не будут равны координатам конечной точки $(xs2, ys2)$. Задаётся значение переменной типа *boolean* для преждевременного выхода из цикла *while* при значении *false*. Для выбора одного из восьми возможных вариантов движения задаётся случайное значение переменной *lm*. В зависимости от её значения рассматривается один из условных операторов *if*.

Внутри каждого из операторов *if* задаётся дополнительный условный оператор для проверки выхода за границы поля, возвращения назад (*ci* = номер из соотношения в п. 2.1.3) и повторения хода (*ci=lm*). Если одно из этих условий выполняется – происходит выход из цикла. В противном случае задаются новые координаты $(xs1, ys1)$ и *ci* присваивается значение *lm*.

```

repeat
check:=false;
while check = false do
begin
lm:=random(1,8);

if lm=1 then
begin
if ((xs1+a*2)>600) or ((ys1-a*1)<0) or (((xs1+a*2)=xs3) and ((ys1-a*1)=ys3)) or
(ci=4) or (ci=1) then
check:=false

```

```
else
begin
xs1:=xs1+a*2;
ys1:=ys1-a*1;
check:=true;
end;
end
else

if lm=2 then
begin
if ((xs1+a*2)>wst) or ((ys1+a*1)>hst) or (((xs1+a*2)=xs3) and ((ys1+a*1)=ys3))
or (ci=3) or (ci=2) then
check:=false
else
begin
xs1:=xs1+a*2;
ys1:=ys1+a*1;
check:=true;
end;
end
else

if lm=3 then
begin
if ((xs1-a*2)<0) or ((ys1-a*1)<0) or (((xs1-a*2)=xs3) and ((ys1-a*1)=ys3)) or
(ci=2) or (ci=3) then
check:=false
else
begin
xs1:=xs1-a*2;
ys1:=ys1-a*1;
check:=true;
end;
end
else

if lm=4 then
begin
if ((xs1-a*2)<0) or ((ys1+a*1)>hst) or (((xs1-a*2)=xs3) and ((ys1+a*1)=ys3)) or
(ci=1) or (ci=4) then
check:=false
else
begin
```

```
xs1:=xs1-a*2;
ys1:=ys1+a*1;
check:=true;
end;
end
else

if lm=5 then
begin
if ((xs1+a*1)>wst) or ((ys1-a*2)<0) or (((xs1+a*1)=xs3) and ((ys1-a*2)=ys3)) or
(ci=8) or (ci=5) then
check:=false
else
begin
xs1:=xs1+a*1;
ys1:=ys1-a*2;
check:=true;
end;
end
else

if lm=6 then
begin
if ((xs1+a*1)>wst) or ((ys1+a*2)>hst) or (((xs1+a*1)=xs3) and ((ys1+a*2)=ys3))
or (ci=7) or (ci=6) then
check:=false
else
begin
xs1:=xs1+a*1;
ys1:=ys1+a*2;
check:=true;
end;
end
else

if lm=7 then
begin
if ((xs1-a*1)<0) or ((ys1-a*2)<0) or (((xs1-a*1)=xs3) and ((ys1-a*2)=ys3)) or
(ci=6) or (ci=7) then
check:=false
else
begin
xs1:=xs1-a*1;
ys1:=ys1-a*2;
```



```

check:=true;
end;
end
else

if lm=8 then
begin
if ((xs1-a*1)<0) or ((ys1+a*2)>hst) or (((xs1-a*1)=xs3) and ((ys1+a*2)=ys3)) or
(ci=5) or (ci=8) then
check:=false
else
begin
xs1:=xs1-a*1;
ys1:=ys1+a*2;
check:=true;
end;
end;
end;
ci:=lm;

```

5. Происходит движение в точку с указанными координатами. Для наглядности в конце пути рисуется небольшая точка.

```

lineto(round(xS1), round(yS1));
moveto(round(xS1), round(yS1));
Circle(round(xS1), round(yS1), 3);

```

6. Проводится проверка нахождения конечной точки в пределах одного хода от последней точки пути. Если условие выполняется – фигура сразу перемещается на конечную точку.

```

if ((xs1+a*2=xs2) and (ys1-a=ys2)) or ((xs1+a*2=xs2) and (ys1+a=ys2)) or
((xs1-a*2=xs2) and (ys1-a=ys2)) or ((xs1-a*2=xs2) and (ys1+a=ys2)) or
((xs1+a=xs2) and (ys1-a*2=ys2)) or ((xs1+a=xs2) and (ys1+a*2=ys2)) or
((xs1-a=xs2) and (ys1-a*2=ys2)) or ((xs1-a=xs2) and (ys1+a*2=ys2)) then
begin
lineto(round(xS2), round(yS2));
xs1:=xs2;
ys1:=ys2;
end;

until (xs1=xs2) and (ys1=ys2);
end.

```

2.3. Анализ результатов

Приведённая задача позволяет понять принцип перемещения тела неизменяемой формы из одной заданной точки ограниченного пространства в другую, предписанную точку.

Для получения кратчайшего из всех возможных путей следует выполнить основную цикл программы число раз, равное количеству возможных комбинаций путей на шахматной доске, добавив счётчик совершенных ходов. После выполнения вывести на экран именно этот вариант.

Выводы

В данной работе был рассмотрен способ моделирования перемещения объекта сложной неизменяемой конфигурации в ограниченном пространстве на примере построения маршрута движения шахматного коня по доске. Такой маршрут предполагает переход из одной клетки доски в предписанную клетку при учёте, что на пути перемещения могут существовать препятствия (желаемые для «хода коня» клетки заняты). Это позволяет оценить минимальное время для прохождения маршрута и встраивания «коня» в предписанную клетку (конфигурацию). Это минимальное время является временем, контролирующим весь сложный процесс перемещения и встраивания объекта неизменяемой конфигурации в предписанную точку ограниченного пространства.

Полученная программа учитывает условия перемещения фигуры, а также не даёт ей выйти за пределы обозначенного поля 8x8 клеток.

Рассмотренный подход ориентирован не только на анализ поведения объекта сложной конфигурации в пространстве. Он позволяет анализировать поведение системы со сложным набором состояний в пространстве случайных событий и может быть использован при построении моделей других систем, основным аспектом поведения которых является встраивание объекта (или набора связанных состояний) в заранее предписанные структуры.

Статья поступила в редакцию 17.01.2019 г.

ЛИТЕРАТУРА

1. An Introduction to Molecular Biology/DNA the unit of life [Электронный ресурс] // Wikibooks : [сайт]. URL: https://en.wikibooks.org/wiki/An_Introduction_to_Molecular_Biology/DNA_the_unit_of_life (дата обращения: 20.12.2018).
2. Xiaoyu Chen, Manuel A.F.V. Gonzalves. DNA, RNA, and Protein Tools for Editing the Genetic Information in Human Cells // iScience. 2018. Vol. 6. P. 247–263.
3. Калашников Е. В. Рост и физические свойства кристаллов. СПб.: Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2013. 117 с.
4. Кон И. С. Постоянство и изменчивость личности // Психологический журнал. 1987. Т. 8. № 4. С. 126–136.
5. Крайг Г. Психология развития. СПб.: Питер, 2000. 992 с.
6. Boon C., Den Hartog D. N. Human resource management, person-environment fit, and trust // Trust and human resource management. 2011. P. 109–121

7. Ben C. H. Kuo. Coping, acculturation, and psychological adaptation among migrants: a theoretical and empirical review and synthesis of the literature // *Health Psychology and Behavioral Medicine*. 2014. Vol. 2. No. 1. P. 16–33.
8. Шишаев М. Г., Елисеенко С. Ю. Имитационная модель пространственных перемещений объектов с квазислучайными параметрами маршрутов // *Труды Кольского научного центра РАН*. 2012. № 6(13). С. 106–114.
9. Космачев С. Н. Автоматизированные информационные системы [Электронный ресурс] // *Пятифан* : [сайт]. URL: <http://5fan.ru/wievjob.php?id=39990> (дата обращения: 20.12.2018).
10. Lee C. Y. An Algorithm for Path Connections and Its Applications // *IRE Transactions on Electronic Computers*. 1961. Vol. EC-10. Iss. 3. P. 346–365.
11. Maze Router: Lee Algorithm [Электронный ресурс]. URL: <http://users.eecs.northwestern.edu/~haizhou/357/lec6.pdf> (дата обращения: 20.12.2018).
12. Loebbing M., Wegener I. The Number of Knight's Tours Equals 33,439,123,484,294 – Counting with Binary Decision Diagrams // *The Electronic Journal of Combinatorics*. 1996. Vol. 3. Iss. 1. P. 36–40.
13. Parberry I. An Efficient Algorithm for the Knight's Tour Problem // *Discrete Applied Mathematics*. 1997. Vol. 73. Iss. 3. P. 251–260.

REFERENCES

1. An Introduction to Molecular Biology/DNA the unit of life. In: *Wikibooks*. Available at: https://en.wikibooks.org/wiki/An_Introduction_to_Molecular_Biology/DNA_the_unit_of_life (accessed: 20.12.2018).
2. Chen X., Gonzalves M.A.F.V. DNA, RNA, and protein tools for editing the genetic information in human cells. In: *iScience*, 2018, vol. 6, pp. 247–263.
3. Kalashnikov E. V. *Rost i fizicheskie svoystva kristallov* [Growth and physical properties of crystals]. St. Petersburg, Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics Publ., 2013. 117 p.
4. Kon I. S. [Constancy and changeability of personality]. In: *Psikhologicheskii zhurnal* [Psychological Journal], 1987, vol. 8, no. 4, pp. 126–136.
5. Craig G. J. *Human Development*. New York, Prentice Hall, 1992, 674 p.
6. Boon C., Den Hartog D. N. Human resource management, person-environment fit, and trust. In: *Trust and human resource management*, 2011, pp. 109–121
7. Ben C. H. Kuo. Coping, acculturation, and psychological adaptation among migrants: a theoretical and empirical review and synthesis of the literature. In: *Health Psychology and Behavioral Medicine*, 2014, vol. 2, no. 1, pp. 16–33.
8. Shishaev M. G., Eliseenko S. Yu. [Simulation model of spatial object movements with quasi-random routes]. In: *Trudy Kol'skogo nauchnogo tsentra RAN* [Works of the Kola Scientific Center of the Russian Academy of Sciences], 2012, no. 6 (13), pp. 106–114.
9. Kosmachev S. N. [Automated information systems]. In: *Pyatifan* [Pyatifan]. Available at: <http://5fan.ru/wievjob.php?id=39990> (accessed: 20.12.2018).
10. Lee C. Y. An Algorithm for Path Connections and Its Applications. In: *IRE Transactions on Electronic Computers*, 1961, vol. EC-10, iss. 3, pp. 346–365.
11. Maze Router: Lee Algorithm. Available at: <http://users.eecs.northwestern.edu/~haizhou/357/lec6.pdf> (accessed: 20.12.2018).
12. Loebbing M., Wegener I. The Number of Knight's Tours Equals 33,439,123,484,294 – Counting with Binary Decision Diagrams. In: *The Electronic Journal of Combinatorics*, 1996, vol. 3, iss. 1, pp. 36–40.

13. Parberry I. An Efficient Algorithm for the Knight's Tour Problem. In: *Discrete Applied Mathematics*, 1997, vol. 73, iss. 3, pp. 251–260.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Акбаров Эльдар Алижонович – студент физико-математического факультета Московского государственного областного университета;
e-mail: melancholywaterhole@gmail.com

Калашников Евгений Владимирович – доктор физико-математических наук, профессор кафедры вычислительной математики и методики преподавания информатики Московского государственного областного университета;
e-mail: ekevkalashnikov1@gmail.com

INFORMATION ABOUT THE AUTHORS

Eldar A. Akbarov – student of the Faculty of Physics and Mathematics, Moscow Region State University;
e-mail: melancholywaterhole@gmail.com

Evgenii V. Kalashnikov – Doctor in Physical and Mathematical Sciences, Professor at the Department of Computational Mathematics and Informatics Teaching Methods, Moscow Region State University;
e-mail: ekevkalashnikov1@gmail.com

ПРАВИЛЬНАЯ ССЫЛКА НА СТАТЬЮ

Акбаров Э. А., Калашников Е. В. Моделирование перемещения объекта сложной неизменяемой конфигурации в ограниченном пространстве // Вестник Московского государственного областного университета. Серия: Физика-Математика. 2019. № 1. С. 107–118.
DOI: 10.18384-2310-7251-2019-1-107-118

FOR CITATION

Akbarov E. A, Kalashnikov E. V. Simulation of motion of an object with immutable complex configuration in a confined space. In: *Bulletin of Moscow Region State University. Series: Physics and Mathematics*, 2019, no. 1, pp. 107–118.
DOI: 10.18384-2310-7251-2019-1-107-118