ОПТИМИЗАЦИЯ МАРШРУТА С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ ГРАФОВ В ПАКЕТАХ ПРИКЛАДНЫХ ПРОГРАММ

Е.Ю. Вардомацкая, В.Л. Шарстнев,
Я.А. Алексеева

УДК 004.9 : 658

РЕФЕРАТ

ТЕОРИЯ ГРАФОВ, МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ТРАНСПОРТНЫЕ СЕТИ, ОПТИМИЗАЦИЯ МАРШРУТА, КРАТЧАЙШИЙ ПУТЬ, ТРАНСПОРТНЫЕ РАСХОДЫ, СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

В статье приведен анализ методов решения задачи о кратчайшем пути с использованием теории графов в пакетах прикладных программ. Для возможности применения системы компьютерной алгебры данная проблема была представлена в виде математической модели на графе. Подробно рассмотрены несколько вариантов решения искомой задачи с помощью СКА Maple. Так, решение данной задачи реализовано методом Дейкстры, методом имитации отжига и методом муравьиного алгоритма. СКА Mathematica также обладает расширенной поддержкой графов, необходимой для решения задачи о кратчайшем пути.

ABSTRACT

GRAPH THEORY, MATHEMATICAL MODELS, FLOW NETWORKS, ROUTE OPTIMIZATION THE SHORTEST PATH, TRANSPORTATION EXPENSES, COMPUTER ALGEBRA SYSTEM

The analysis of methods for solving the using graph theory in application packages is represented in this article. For possibility of use of computer algebra system this problem was presented in the form of a mathematical model on the graph. Some versions of the solution of the problem by means of various CAS are in detail considered. Indisputable leaders in in this area are CAS Maple and CAS Mathematica. The solution of this problem by means of CAS Maple is realized by Dijkstra's algorithm, Simulated annealing and Ant colony optimization algorithms. CAS Mathematica has expanded support graphs necessary for solution of the shortest path problem.

В современных условиях развития рыночных отношений всё более актуальным становится всестороннее обеспечение конкурентоспособности как экономики Республики Беларусь в целом, так и её отдельных отраслей. В частности, легкой промышленности – одного из важнейших секторов экономики, который производит большое количество товаров народного потребления. В последние годы в сфере товарного обращения ряда стран произошли существенные преобразования. В хозяйственной практике стали использоваться новые методы и технологии доставки товаров, базирующиеся на концепции логистики. Важную роль в создании объективных возможностей для развития логистики сыграл технический прогресс в средствах связи и информатики. Он позволил на более высоком уровне проводить отслеживание всех основных и вспомогательных процессов товародвижения. Так, постепенно на первый план выдвигается поиск возможностей сокращения производственных затрат, в том числе транспортных расходов.

Цель исследования – разработка средств компьютерного моделирования процесса управления цепями поставок.

Объект исследования – логистические системы предприятий легкой промышленности.

Актуальность темы исследования обусловливается необходимостью поиска оптимальных схем грузовых потоков для производимых товаров, позволяющих снизить транспортные издержки предприятий, обеспечивая, тем самым, их конкурентоспособность.

Методы исследования – абстрагирование, математическое моделирование, анализ.

Инструментарий исследования – система компьютерной математики (далее – СКМ) Maple.

Научная новизна исследования заключается в синтезе использования различных моделей оптимизации, построенных на графах, в решении задач транспортной логистики.

Практическая значимость исследования заключается в возможности улучшения технико-экономических показателей работы предприятия, обусловленной экономией трудовых и финансовых ресурсов.

Задача оптимизации транспортных расходов заключается в отыскании оптимального плана перевозок грузов из пунктов отправления в пункты потребления, что означает необходимость поиска кратчайших расстояний с минимальными затратами. При густоразветвленной сети автомобильных дорог, когда между пунктами отправления и пунктами назначения имеется несколько вариантов сообщений, определить такой путь бывает сложно. Для нахождения оптимального варианта сообщения применяют математические модели, основанные на использовании в качестве исходной информацию транспортной сети, отражающей транспортные связи между пунктами отправления и назначения грузов. Из всех математических объектов графы занимают одно из первых мест в качестве формальных моделей реальных систем. Граф представляет собой совокупность конечного числа точек, называемых вершинами графа, и попарно соединяющих некоторые из этих вершин линий, называемых ребрами или дугами графа. Использование графовых моделей в решении оптимизационных задач сохраняет наглядность и содержательность описываемых объектов и позволяет строить формальные алгоритмы обработки этих моделей, которые легко обрабатывается на ЭВМ.

Таким образом, транспортные сети удобно представить в виде графа.

Транспортная сеть учитывает только ту часть дорожной сети, по которой возможно организовать соответствующие перевозки, то есть учитываются ограничения по состоянию улиц (дорог), одностороннее движение, ограничения на движение грузового транспорта, на полную массу транспортного средства, нагрузка на ось и другие.

Транспортная сеть может быть представлена только связным графом.

Моделирование транспортной сети начинают с размещения вершин. Вершины присваивают грузообразующим и грузопоглощающим пунктам, центрам крупных жилых кварталов, обособленных населенных пунктов. Вершины, имеющие между собой транспортное сообщение, связывают ребрами или (в случае односторонней связи) ориентированными дугами.

Каждому ребру сопоставляют критерий выгодности, определяемый не только затратами времени, а той целью, которую необходимо достигнуть при решении задачи оптимального варианта перевозок. Наиболее часто в качестве критерия принимается минимум суммарного пробега, так как при одинаковых условиях движения на всех участках маршрута план, оптимальный по пробегу, будет оптимальным по затратам времени и стоимости. Кроме того, в качестве критерия выгодности так же могут быть использованы такие показатели, как платность или загруженность дорог, частота пересечения данной дороги населенных пунктов и так далее.

При построении графа следует выбирать рациональное число вершин. С одной стороны, число вершин должно быть как можно больше. С другой стороны, чем больше число вершин, тем транспортная сеть будет сложнее, определение кратчайших расстояний потребует длительного времени.

Для снижения размерности задачи и ускорения расчетов для транспортных сетей больших городов или районов применяют микро- и макрорайонирование. При микрорайонировании транспортной сети в качестве вершин используют не пересечения улиц (дорог) и конкретные пункты отправления и назначения, а центры микрорайонов (районов получения или назначения грузов). Макрорайонирование транспортной сети заключается в разбиении ее на отдельные подсети, расчеты по которым выполняются раздельно, а затем объединяются для получения общего результата. При изменениях дорожной обстановки производится пересчет отдельной подсети, в которой произошли изменения, но не по всей транспортной сети [5, с. 171-172].

Задача о кратчайшем пути является одной из важнейших классических задач теории графов. Значимость данной задачи определяется ее различным практическим применениям. Так, задача поиска кратчайшего пути на графе широко используется для нахождения путей между физическими объектами на картографических сервисах, в GPS-навигаторах, при определении наименьшего расстояния в сети дорог.

Выделяют несколько вариантов решения задачи о кратчайшем пути на графах с помощью различных пакетов прикладных программ. Бесспорным лидером в данной области является СКА Maple. Рассмотрим решение исходной проблемы на примере использования данного программного пакета, имеющего специализированную библиотеку networks для работы с графами.

Выделяют несколько способов решения данной задачи – это, прежде всего, метод Дейкстры, метод отыскивания всех гамильтоновых циклов, алгоритм ближайшего соседа (Nva), муравьиный алгоритм, метод имитации отжига и другие. Остановимся на некоторых из них более подробно.

Наиболее точным и при этом простым методом решения транспортной задачи в СКА Maple на графах является метод Дейкстры – пошаговый алгоритм определения кратчайшего расстояния от вершины A до B.

Реализацию метода Дейкстры в СКА Maple проиллюстрируем решением следующей практической задачи. Предприятие легкой промышленности города Витебска наладило поставки своей продукции в магазины регионов Республики Беларусь. Необходимо найти кратчайший путь перевозок грузов от места производства к местам потребления, оптимизирующий транспортные издержки предприятия.

В программе, представленной на рисунке 2, реализован данный метод на примере орграфа с неотрицательными весами, смоделированного на основе путей сообщения между пятью городами Республики Беларусь (рисунок 1).

Ответ заносится в переменную MinPath. Алгоритм заканчивает свою работу, когда flag принимает значение true, то есть конечная вершина (target) приобретает постоянную метку. После завершения работы программы список постоянных меток можно посмотреть, раскрыв переменную V: evalm(V) [3, с. 111].

Также кратчайший путь в орграфе можно найти, используя стандартные процедуры СКА Maple. Рассмотрим заданный в алгоритме Дейкстеры граф и вычислим кратчайший путь. В программе на рисунке 3 независимо используются два оператора, shortpathtree и allpairs.

В алгоритме ближайшего соседа (рисунок 4) выбор дальнейшего пути производится, исходя из минимального расстояния до очередной вершины. Обход графа начинается с произвольной вершины, от выбора которой часто зависит результат. Движение по графу происходит по исходящим (Out) дугам минимального веса с по-

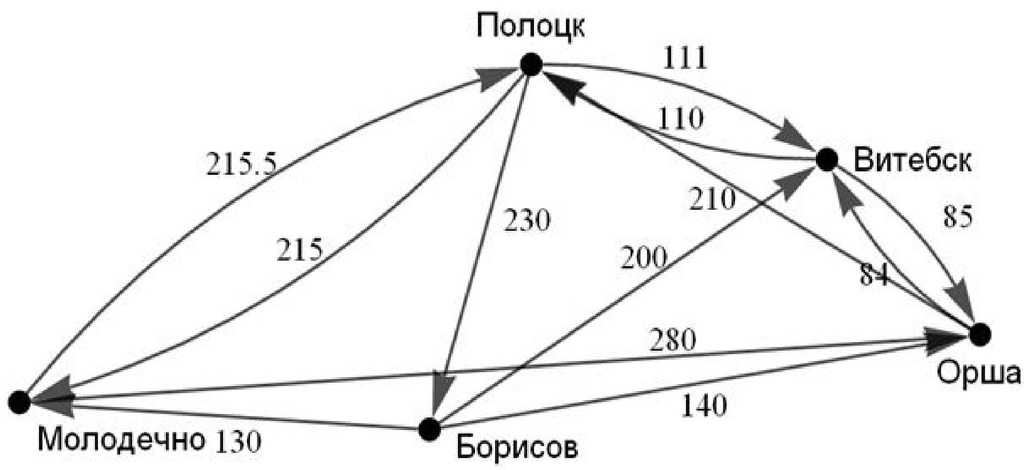

Рисунок 1 – Ориентированный взвешенный граф

```
> restart : with(networks) :
> new(G) : n := 5 :
> addvertex(i$i = 1 ..n, G) :
> addedge([[1, 2], [2, 1], [1, 3], [3, 1], [3, 2], [5, 3], [4, 3], [5, 4], [4, 2], [2, 4], [2, 5], [5, 1]],
  weights = [110, 111, 85, 84, 210, 140, 280, 130, 215.5, 215, 230, 200], G) :
> V := Vector(1 ..n) :
> for i to n do V[i] := infinity; od:
> s := 1 : target := 5 : k := s : V[k] := 0 : U := [0$3·n] :
> flag := false :
> for i while not flag do
     U[i] := k :
     d := outdegree(k, G) : z := departures(k, G) :
     for j to d do CW1 := eweight(op(edges([k, z[j]], G)), G) :
        if ((V[z[j]] = 0) or (V[z[j]] > CW1 + V[k]))
        then V[z[j]] := eweight(op(edges([k, z[j]], G)), G) + V[k] : fi; od;
     Next := n;
     for j from 2 to n do
        if not member(j, U) and V[j] < V[Next] then Next := j; fi; od:
     k := Next;
     flag := is(k = target);
   od:
> evalm(V);
                              [ 169  110  85  325  340 ]
> MinPath := V[4];
                                   MinPath := 325
```

Рисунок 2 – Программная реализация метода Дейкстры

```
> restart : with(networks) :
> new(G) : n := 5 :
> V := [1, 2, 3, 4, 5] : n := nops(V) :
> addvertex(V, G) :
  addedge([[1, 2], [2, 1], [1, 3], [3, 1], [3, 2], [5, 3], [4, 3], [5, 4], [4, 2], [2, 4], [2, 5], [5, 1]],
  weights = [110, 111, 85, 84, 210, 140, 280, 130, 215.5, 215, 230, 200], G) :
> T := shortpathtree(G, 1) :
> W := vweight(T);
              W := table(sparse, [1 = 0, 2 = 110, 3 = 85, 5 = 340, 4 = 325])
> MinPath := W[4];
                                   MinPath := 325
> allpairs(G)[1, 4];
                                        325
```

Рисунок 3 – Программа нахождения кратчайшего пути с использованием операторов shortpathtree и allpairs

```
> restart : with(networks) :
> new(G) : n := 5 :
> addvertex(i$i = 1..n, G);
                                        1, 2, 3, 4, 5
> addedge([[1, 2], [2, 1], [1, 3], [3, 1], [3, 2], [5, 3], [4, 3], [5, 4], [4, 2], [2, 4], [2, 5], [5, 1]],
      weights = [110, 111, 85, 84, 210, 140, 280, 130, 215.5, 215, 230, 200], G) :
> G1 := duplicate(G) :
> a0 := 5 : # Начальная вершина
> a1 := a0 : sw := 0 :
> s := [a0] :
> for k to n-1 do
    for v in incident(a1, G1, Out) do
      if eweight(v, G1) = min(op(eweight([op(incident(a1, G1, Out))], G1)))
      then u := v; sw := sw + eweight(v, G1); break; fi; od;
    a2 := ends(u, G1)[2];
    delete(a1, G1) :
    a1 := a2 : s := [op(s), a2] :
  od:
> sw := sw + eweight(op(edges([a2, a0], G)), G); # Сумма
sw := 541.5 + table([e12 = 200, e9 = 215.5, e8 = 130, e4 = 84, e1 = 110, e7 = 280, e5 = 210, e3
    = 85, e10 = 215, e11 = 230, e2 = 111, e6 = 140])
> s; # Контур
                                        [5, 4, 2, 1, 3]
```

Рисунок 4 – Программная реализация метода ближайшего соседа (Nva)

следующим удалением пройденной вершины. Поэтому перед началом процесса необходимо запастись дубликатом графа, из которого затем будем извлекать информацию о длине пройденного пути. Оператор break досрочного выхода из цикла введен для случая двух или более одинаковых дуг минимального веса, выходящих из текущей вершины [3, с. 145]. Если дать возможность алгоритму случайно выбирать решение, оптимальное на каждом шаге, то можно пропустить ход, не лучший локально, но дающий в результате более эффективное решение, что, несомненно, является существенным недостатком данного алгоритма. Однако несмотря на предельную простоту алгоритма, для небольших графов он дает достаточно близкие к точному решению ответы.

Еще один способ решения задачи о кратчайшем пути основан на использовании программы отыскивания всех гамильтоновых циклов, представлен в общем виде на рисунке 5. Для перемножения матриц используется операция некоммутативного умножения. Поэлементно присваиваем ***C[i,j]:=P[i,j]***, так как простое присваивание ***C:=P***, предусмотренное для работы с матрицами, работает ненадежно и часто является причиной ошибок [3, с. 143].

Для решения задачи о кратчайшем пути необходимо вычислить количество ***n1:=nops(H)*** гамильтоновых циклов и найти минимальный из них. Для этого разработана процедура вычисления веса дуги ***u-v*** в цикле ***k*** и использована стандартная функция отыскания минимума последовательности (рисунок 6) [3, с. 146].

Время вычисления при этом будет в несколько раз больше, и с увеличением порядка графа разность в быстродействии программ будет расти.

Несомненное достоинство алгоритма – его точность и простота.

Муравьиный алгоритм Марко Дориго является методом искусственного интеллекта. По форме этот алгоритм похож на ***Nva*** и в некоторой степени является его обобщением, однако здесь выбором управляет случайная функция, направляющая движение от текущего положения с

```
> restart : with(networks) : new(G) :
> V := [a, b, c, d, g] : n := nops(V) :
> addvertex(V, G) :
> addedge({{a, c}, {a, d}, {b, g}, {b, c}, {c, d}, {d, g}, [a, g], [a, b]}, G) :
> B := Matrix(n) :
> A := adjacency(G) :
> for i to n do
    for j to n do if A[i, j] = 1 then B[i, j] := V[j]; fi;  od:
  od:
> C := A :
> for k to n-1 do
    for i to n do
      for j to n do P[i, j] := expand(subs(V[i] = 0, V[j] = 0, add(B[i, m]&*C[m, j], m = 1 ..n)));
      od;
    od;
    for i to n do
      for j to n do C[i, j] := P[i, j]; od;
    od;
  od:
> S := { } :
> for i to n do  if C[i, i] ≠ 0 then  S := S union {expand(C[i, i]&*V[i])}; fi; od:
> F := { } :
> for q in S do if whattype(q) = `+` then F := F union {op(q)} else F := F union {q} : fi; od;
> S := map(x → convert(x, list), F) :
> for q in S do if nops(q) ≠ n then S := S minus {q}; fi; od;
> k := nops(S) :
> H := { } :
> for j to k do
    for i to n do if S[j][i] = V[1] then m := i; fi; od;
     for i to n do Z[i] := S[j][(i + m-2 mod n) + 1]; od;
     H := H union {convert(Z, list)} :
  od:
> H;
              {[a, b, c, g, d, f], [a, b, g, c, d, f], [a, f, d, c, b, g], [a, f, d, c, g, b]}
```

Рисунок 5 – Программа отыскивания всех гамильтоновых циклов

```
> L := (u, v, k) → eweight(op(edges([H[k][u], H[k][v]], G)), G) :
> min(seq(add(L(i, i + 1, k), i = 1 ..n-1) + L(n, 1, k), k = 1 ..n1));
```

Рисунок 6 – Продолжение программы отыскивания гамильтоновых циклов

большей вероятностью в вершину, в которой наибольшее значение некоторой функции Pij,к (где i — номер вершины, в которой производится выбор, κ — номер муравья, движущегося по дугам графа). Как и в Nva, во время движения создается список пройденных вершин, что позволяет избежать преждевременного зацикливания.

Рассмотрим программу минимизации транспортных расходов для n городов, координаты $x[i]$ и $y[i]$ которых заданы (рисунки 7, 8). В начале программы вводятся константы задачи. Улучшение сходимости во многом зависит от их значений. Число $Lmin$ требуется для сравнения при выборе минимального маршрута, масштабная константа Q порядка длины маршрута выбирается пропорциональной порядку графа.

```
> P := proc (x)
  local n, sm, i, Beg, End, c, s, m, j;
  n := nops(x); sm := 0;
  for i to n do sm := sm + x[i] od;
  c := 0;
  for i to n do Beg[i] := c; End[i] := c + x[i]/sm; c := End[i] od;
  m := rand(0 .. 100); s := (1/101)*m( );
  for i to n do if Beg[i] ≤ s and s ≤ End[i] then j := i end if end do;
  return j;
  end proc:
```

Рисунок 7 – Фрагмент программы муравьиного алгоритма Марко Дориго

```
> for k0 to 100 do # Основной цикл
    for ant to n do # Цикл по муравьям
      s := {$ 1 ..n } : # Список непосвященных вершин
      j := ant : # Начальная вершина для маравья ant
      for j1 to n − 1 do s := s minus { j}; # Tabu list увеличился
        sp := [ ] : k1 := 0 :
        for i in s do # Каждой вершине - свой вес
          sp := [op(sp), 1 / W[ j, i]^b * Wt[ j, i]^a] :
          k1 := k1 + 1 : nn[k1] := i;
        od:
        j0 := nn[P(sp) ]; # Выбор направления
        v[ j1] := j0; r[ j1] := W[ j, j0];
        j := j0 : # Начало дуги - это конец предыдущей
      od: # Цикл j1 по всем вершинам для муравья ant
      L := add(r[i], i = 1 ..n−1) + W[op(s), ant];
      v[0] := ant; v[n] := ant; # Начало и конец дуги
      v1 := seq([v[m], v[m + 1]], m = 0 ..n−1) : # Дуги
      for i to n do # Пометка дуг феромоном
        DWt[op(v1[i])] := DWt[op(v1[i])] + Q / L;
      od:
      if L < Lmin then Lmin := L :fi: # Выбор min
    od: # ant
    Wt := Wt + DWt*p : # Добавление новых следов
    Wt := Wt* (1−p) : # Испарение феромона
  od: # k0
```

Рисунок 8 – Основной фрагмент программы муравьиного алгоритма

Функция ***P*** вероятности перехода имеет в качестве аргумента список из чисел *x[i]*. Вычисляется сумма ***sm***= ∑*x[i]*. Отрезок прямой от 0 до 1 разбивается на ***n*** участков ***[Beg[i], End[i]]*** с длинами $x[i]/sm$, где $Beg[1] = 0$, $End\,[n] = 1$. Затем случайное число $0 \le s \le l$ указывает выигрышный номер — номер вершины для дальнейшего движения [3, с. 147].

Алгоритм отжига, как и муравьиный алгоритм, относится к вероятностным методам решения. Ключевым моментом в таких подходах является случайный выбор одного из нескольких возможных решений вместо анализа каждого. Это позволяет сократить время счета.

В методе отжига (рисунок 9) очередной порядок следования по маршруту между городами выбирается случайно, небольшим изменением предыдущего решения, предположительно оптимального. Самый простой вариант изменения – перестановка двух случайно выбранных городов в маршруте следования. Если полученный маршрут лучше всех существовавших ранее, то этот маршрут берется за очередной, если маршрут хуже, то самый простой вариант – не брать его. Следует отметить, что решение зависит от нескольких параметров. Меняя, например, число циклов, можно получать различные ответы [3, с. 89].

Таким образом, рассмотрев возможные решения задачи о кратчайшем пути на графах, нельзя однозначно выделить один из методов решения, каждый из них имеет как очевидные достоинства, так и недостатки, такие как, необходимость знания Maple-языка, являющегося функционально полным процедурным языком программирования четвертого поколения (4GL), ручной ввод исходных данных и другие. Однако согласно проведенному анализу, более предпочтительным методом решения исходной проблемы в СКМ Maple является метод Дейкстры.

Экономический эффект от внедрения средств компьютерного моделирования процесса управления цепями поставок может быть лишь косвенным, так как внедренные средства автоматизации не являются прямым источником дохода, а представляют собой вспомогательное средство получения прибыли, делая процессы управления предприятием полностью прозрачными и контролируемыми.

Главный экономический эффект от внедрения разрабатываемой методики заключается в улучшении экономических и хозяйственных показателей работы предприятия, в первую очередь за счет снижения транспортных издержек предприятия. Для большинства предприятий экономический эффект может также выступать в виде экономии трудовых и финансовых ресурсов, получаемой от снижения трудоемкости расчетов, снижения трудозатрат на поиск и подготовку документов, сокращения служащих предприятия.

Рассмотрим на конкретном примере возможный экономический эффект от внедрения разработанной методики оптимизации транспортных расходов по оценочным данным на примере предприятия легкой промышленности города Витебска, занимающегося производством трикотажных изделий, в котором автоматизируется поиск оптимального маршрута транспортировки произведенной продукции в фирменные магазины. Средняя стоимость пакета СКМ Maple составляет 10,25 млн. руб. Предполагаем, что внедрением, технической поддержкой и возможными доработками разработанной методики в данных СКМ будет заниматься дополнительно обученный штатный работник. Стоимость дополнительного обучения примем равной 1 млн. руб.

В итоге затраты на внедрение нового программного обеспечения составят:

```
> for i to 500 do i1 := m() : i2 := m() : Z := Z0 : z := Z[i1] : Z[i1] := Z[i2] : Z[i2] := z :
  L1 := add(W[Z[i], Z[i + 1]], i = 1 ..n−1) + W[Z[n], Z[1]] :
  if L1 < L0 then L0 := L1 : Z0 := Z : k := k + 1 : TT[k] := T : L[k] := L1 : V0 := map('*', Z, 1);
  else P0 := p()/100. :  P := exp(−(L1−L0)/T) :
    if P0 < P then Z0 := Z : fi: fi:
  if (i mod 10) = 0 then T := alpha*T : fi; od:
```

Рисунок 9 – Фрагмент программы "Метод имитации отжига"

$$З_{Maple} = 10,25 + 1 = 11,25 \text{ млн. руб.} \quad (1)$$

Для оценки экономии от разработанной методики в качестве критерия можно выбрать:

- значение коэффициента использования пробега;
- оптимизация маршрута (его протяженность, пропускная способность, скорость, загруженность);
- затраты топлива на 1 т перевезенного груза.

В нашем примере для простоты расчета будем считать, что основная экономия достигается за счет сокращения протяженности пройденного маршрута при транспортировке груза от производителя к потребителю. Таким образом, годовая экономия будет равна экономии, связанной с сокращением транспортных расходов на топливо, а также с сокращением величины износа грузового автомобиля и отчислений в амортизационный фонд.

Рассчитаем экономию за счет сокращения транспортных расходов на топливо. По оценочным данным, средняя величина сокращения протяженности пройденного за счет оптимизации маршрута на одну поездку составляет около 60 км, норма потребления грузовым автомобилем топлива на 100 км в загруженном состоянии может достигать 15 л. Проанализировав текущую ситуацию на рынке автомобильного топлива, можно сделать заключение, что средняя цена топлива составляет 11 000 руб. Экономия, связанная с сокращением транспортных расходов на топливо на одну грузоперевозку, составляет:

$$P = 15 \times \frac{60}{100} \times 11000 = 99000 \, руб. \quad (2)$$

Предположим, что предприятие обновляет ассортимент два раза в месяц в десяти фирменных магазинах. Таким образом, величина годовой экономии составит:

$$P = 99000 \times (10 \times 4 \times 12) = 47{,}52 \, млн.руб. \quad (3)$$

Тогда экономический эффект от внедрения нового программного обеспечения может быть рассчитан по формуле

$$Э_{Maple} = \frac{\sum_{t=1}^{k} (P - З_{Maple})}{k} \quad . \quad (4)$$

где *Э* – годовая суммарная экономия, млн.руб.; *З* – годовые суммарные затраты, связанные с внедрением нового программного обеспечения, млн. руб.; *k* – количество лет реализации проекта плюс один год, лет.

$$Э_{Maple} = \frac{(47{,}52 - 11{,}25) + 47{,}52}{2} = 41{,}9 \, млн.руб. \quad (5)$$

Данные величины экономического эффекта свидетельствует о том, что даже при приблизительном расчете с учетом только одного фактора экономии минимальный эффект от внедрения разработанный методики в расчете на один грузовой автомобиль получился значительным.

Также к числу экономических результатов относятся увеличение объема перевозок и улучшение качества работы транспортной системы, способствующие повышению конкурентоспособности предприятия, эффективности экономики и увеличению национального богатства. Кроме того, экономические результаты заключаются в более эффективном использовании хозяйственных ресурсов, их экономии или предотвращении потерь в производственной и непроизводственной сферах народного хозяйства, достигаемых благодаря совершенствованию транспортного процесса.

Кроме экономического эффекта очевидным является экологический эффект, заключающийся в уменьшении отрицательного воздействия грузового транспорта на окружающую среду в связи со снижением загрязнения её атмосферы вредными выбросами и шумом и достигаемый за счет уменьшения длины транспортных маршрутов, объемов используемого топлива и так далее.

Таким образом, по результатам расчета экономического эффекта от внедрения разработанной методики оптимизации транспортных расходов с использование теории графов в пакетах прикладных программ можно заключить, что это выгодно предприятиям. Хоть выгода и косвенная, но, как правило, заметна в средне- и долгосрочной перспективе за счет снижения себестоимости произведенной продукции. Все это обусловливает перспективность внедрения предложенной методики на предприятиях промышленности и бытового обслуживания населения всех форм собственности Республики Беларусь.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аладьев, В.З., Бойко, В.К., Ровба, Е.А. (2011), *Программирование в пакетах Maple и Mathematica: Сравнительный аспект*, Гродно, 2011, 516 с.
2. Березина, Л.Ю. (1979), *Графы и их применение*, Москва, Просвещение, 1979, 143 с.
3. Кирсанов, М.Н. (2007), *Графы в Maple. Задачи, алгоритмы, программы*,Москва, 2007, 168 с.
4. Кристофидес, Н. (1978), *Теория графов. Алгоритмический подход*, Москва, Мир, 1978, 432 с.
5. Хлевной, И.И. (2006), *Грузовые перевозки*, СПб., 2006, 290 с.
6. *Расчет экономического эффекта от внедрения системы автоматизации* [Электронный ресурс] / Компания «Antegraconsulting». – Режим доступа: http://www.antegra.ru. – Дата доступа: 14.05.2014.
7. Sharstniou, U. L., Vardamatskaja, A.U. (2007) Computer information technology: software packages for modeling and analysis of problems in economics: a tutorial [Компьютерные информационные технологии: пакеты прикладных программ для моделирования и анализа задач экономики, Vitebsk EE «VSTU», 2007. 138str

REFERENCES

1. Alad'ev, V.Z., Bojko, V.K., Rovba, E.A. (2011), *Programmirovanie v paketah Maple i Mathematica: Sravnitel'nyj aspekt* [Programming in the Maple and Mathematica packages: A comparative aspect], GGU, Grodno, 2011, 516 p.
2. Berezina, L. Ju. (1979), *Grafy i ih primenenie* [Graphs and their application], Prosveshhenie, 1979, 143 p.
3. Kirsanov, M.N. (2007), *Grafy v Maple. Zadachi, algoritmy, programmy* [Graphs in Maple. Tasks, programs, algorithms], Moscow, 2007, 168 p.
4. Kristofides, N. (1978) *Teorija grafov. Algoritmicheskij podhod* [Graph theory. An algorithmic approach], Moscow, Mir, 1978, 432 p.
5. Hlevnoj, I.I. (2006), *Gruzovye perevozki* [Freight transport], SPb., 2006, 290 p.
6. *The calculation of the economic effects of the introduction of automation systems* [electronic resource] [Raschet jekonomicheskogo jeffekta ot vnedrenija sistemy avtomatizacii] [Elektronnyj resurs] / Kompanija «Antegraconsulting». – Rezhim dostupa: http://www.antegra.ru. – Data dostupa: 14.05.2014.
7. Sharstniou, U. L., Vardomatskaja, A.U. (2007) Computer information technology: software packages for modeling and analysis of problems in economics: a tutorial [Kompjuternye informacionnye tehnologii: pakety prikladnyh programm dla modelirovanija i analiza zadach economiki], Vitebsk EE «VSTU», 2007. 138 str.

Статья поступила в редакцию 18. 01. 2016 г.