

Алгоритмы формирования устойчивых связей поставщиков и потребителей на рынке БРИКС

Algorithms for the formation of stable links between suppliers and consumers in the BRICS market

УДК 339

Получено: 16.12.2021

Одобрено: 16.01.2022

Опубликовано: 25.02.2022

Чупина Ж.С.,

Канд. экон. наук, доцент, доцент кафедры таможенного дела юридического института, РУДН

e-mail: lemesheva.87@bk.ru

Chupina Zh.S.

Candidate of Economic Sciences, Associate Professor, Associate Professor of the Department of Customs Affairs of the Law Institute, RUDN

e-mail: lemesheva.87@bk.ru

Чупин А.Л.

Начальник отдела организации научно-технических мероприятий, научное управление, РУДН,

e-mail: chupin_al@pfur.ru

Chupin A.L.

Head of the unit for organization of scientific and technical events, Research establishment of RUDN University,

e-mail: chupin_al@pfur.ru

Аннотация

Результатом решения транспортной задачи в сетевой постановке являются величины потоков на дугах сети. Однако такой вид решения не всегда является удовлетворительным. В действительности, часто необходимо знать не только величины нагрузок на дугах сети, но и образующиеся связи между поставщиками и потребителями. Представленные связи рассмотрены на примере рынка стран – участниц БРИКС. Для установления таких связей в рамках построенного оптимального плана служат алгоритмы привязки поставщиков к потребителям.

Ключевые слова: транспортная задача, поставщики, потребители, БРИКС, устойчивость.

Abstract

The result of solving the transport problem in the network formulation is the magnitude of the flows on the arcs of the network. However, this type of solution is not always satisfactory. In fact, it is often necessary to know not only the magnitude of the loads on the network arcs, but also the connections formed between suppliers and consumers. The presented connections are considered on the example of the market of the BRICS member countries. To establish such links within the framework of the optimal plan, algorithms for linking suppliers to consumers are used.

Keywords: transport task, suppliers, consumers, BRICS, sustainability.

1. Введение

Для решения транспортных задач в классической постановке существует довольно много эффективных алгоритмов и программ [1–5]. Однако зачастую при решении практических проблем недостаточно один раз решить транспортную задачу. Часто транспортная задача используется как элемент итеративного процесса, например, при решении задач размещения с дополнительными ограничениями нетранспортного типа или задач с нелинейными ценами [6–11]. В таких случаях полученный в результате решения оптимальный план является лишь этапом работы [12–15, 31]. В зависимости от результатов изменяются исходные задачи, и транспортная задача решается заново. Однако такой способ нельзя признать экономным с точки зрения вычислительной эффективности [16–20]. Для этого необходимо при внесении изменений в исходные данные корректировать полученный ранее оптимальный план при помощи специальных процедур. В настоящей работе приводятся алгоритмы формирования связей поставщиков и потребителей.

2. Типовая исходная задача

Пусть задана ориентированная сеть $G = \{N, U, \beta\}$, где N – множество вершин сети, U – множество ориентированных дуг, β – величина потоков на дугах. Предположим, что $\beta(u) > 0$ для всех $u \in U$.

Для каждой вершины i определим два множества дуг: $A(i)$ – множество входящих дуг, $B(i)$ – множество выходящих.

Величину $Q(i) = \sum_{A(i)} \beta(u) - \sum_{B(i)} \beta(u)$ будем называть мощностью вершины. Если $Q(i) > 0$, то вершина является поставщиком, если $Q(i) < 0$ – потребителем, а если $Q(i) = 0$ – промежуточной вершиной.

Целью мы называем последовательность ориентированных дуг заданным на ней положительным потоком:

$$C = \{(i_0, i_1), (i_1, i_2), \dots, (i_{K-1}, i_K) | \varepsilon\}.$$

Началом цепи всегда будет поставщик, а концом – потребитель. Цепи служат для описания связей между поставщиками и потребителями. Задание цепи C означает, что поставщик i_0 отправляет потребителю $i_K \in$ единиц продукта через пункты i_1, i_2, \dots, i_{K-1} . Через $n(C)$ будем обозначать начальную вершину цепи, через $K(C)$ – конечную, а через $\varepsilon(C)$ – поток цепи.

В данной задаче требуется найти такое множество цепей C_1, \dots, C_l , чтобы выполнялись следующие условия.

Для любой цепи

$$Q(n(C_j)) > 0, Q(K(C_j)) < 0. \quad (1)$$

Для каждой вершины

при $Q(i) > 0$

$$\sum_{C_j | n(C_j)=i} \varepsilon(C_j) = Q(i) \quad (2)$$

при $Q(i) < 0$

$$\sum_{C_j | n(C_j)=i} \varepsilon(C_j) = -Q(i).$$

Для каждой дуги

$$\sum_{C_j | u \in C_j} \varepsilon(C_j) \leq \beta(u). \quad (3)$$

Первое условие означает, что цепи начинаются в поставщиках и заканчиваются в потребителях. Второе условие означает, что сумма потоков по цепям, начинающимся в поставщике, равна мощности этого поставщика, а сумма потоков по цепям, заканчивающимся в потребителе, равна абсолютной величине его мощности. Из третьего условия следует, что сумма потоков по цепям, проходящим через данную дугу, не превосходит потока по этой дуге. Легко заметить, что эти три условия формализуют интуитивные представления о правилах привязки поставщиков к потребителям [21–25].

3. Формализованное представление алгоритма

Представим алгоритм построения искомого множества цепей сначала в случае ациклической сети – сети без ориентированных циклов. На самом деле, в результате решения транспортной задачи в абсолютном большинстве случаев будет получена именно ациклическая сеть. Действительно, для транспортной задачи без ограничений пропускной способности решение будет деревом, а для транспортной задачи с ограничениями ориентированный цикл в оптимальном плане может появиться только тогда, когда в исходной сети существует ориентированный цикл, состоящий из дуг нулевой стоимости [26, 27].

Прежде чем перейти к описанию алгоритма, необходимо сформулировать определение и доказать одно свойство ациклических сетей.

Вершина, в которую не входит ни одна дуга, называется исходной, а вершина, из которой не выходит ни одной дуги, называется тупиковой [28–30].

Лемма 1. В ациклической сети найдется одна исходная вершина.

Доказательство. Пусть это не так и в любую вершину входит хотя бы одна дуга. Рассмотрим произвольную вершину сети i_0 . Возьмем (i_0, i_1) , входящую в вершину сети i_0 . Далее возьмем дугу i_1, i_2 и т.д. Прерваться этот процесс не может, по предположению. Рассмотрим полученную последовательность вершин $i_0, i_1, i_2, \dots, i_K, \dots$. Среди них найдутся одинаковые: $i_l = i_m$. Но тогда дуги $(i_m, i_{m-1}), (i_{m-1}, i_{m-2}), \dots, (i_{l+1}, i_l)$ образуют ориентированный цикл, что противоречит ациклическости сети.

Коротко опишем структуру информации, необходимую для работы алгоритма. В каждой вершине нам нужен список выходящих из нее дуг. Обозначим этот список через $A(i)$. Кроме того, для каждой вершины будем помнить $deg(i)$ – количество входящих в нее дуг.

Алгоритм привязки выглядит следующим образом.

1. Выбираем вершину i_0 такую, что $deg(i_0) = 0$. Если такой вершины нет, то заканчиваем работу. Иначе переходим к шагу 2.

2. Если вершина i_0 тупиковая, т.е. $A(i_0)$ пуст, то удаляем i_0 из сети и переходим к шагу 1. Если $A(i_0)$ не пуст, то объявляем i_0 последней вершиной построенной цепи. Полагаем $\varepsilon = \infty$. Переходим к шагу 3.

3. Пусть i_K последняя вершина цепи. Если $A(i_K)$ пуст, заканчиваем построение цепи и переходим к шагу 4. Если $A(i_K)$ не пуст, то обозначим через i_{K+1} конечную вершину 1-й дуги списка $A(i_K): (i_K, i_{K+1})$. Полагаем, $\varepsilon = \min(\varepsilon, \beta(i_K, i_{K+1}))$. Объявляем i_{K+1} последней вершиной цепи и повторяем шаг 3.

4. Добавляем цепь $C = ((i_0, i_1), (i_1, i_2), \dots, (i_{K-1}, i_K) | \varepsilon)$ в список построенных цепей.

На всех дугах цепи производим пересчет потоков:

$$\beta(i_l, i_{l+1}) = \beta(i_l, i_{l+1}) - \varepsilon.$$

Если после пересчет $\beta(i_l, i_{l+1}) = 0$, то выбрасываем дугу (i_l, i_{l+1}) из списка $A(i_l)$ и уменьшаем $deg(i_{l+1})$ на единицу. Переходим к шагу 2.

Из леммы 1 следует, что алгоритм закончит работу, когда все вершины будут удалены из сети.

Рассмотрим теперь сеть, содержащую циклы. В этом случае описанный алгоритм не применим. Для того, чтобы получить возможность им пользоваться, необходимо предварительно «разорвать» в сети все ориентированные циклы, что и делает приведенный ниже алгоритм удаления циклов. Информация, используемая этим алгоритмом, такая же, как и у предыдущего.

Алгоритм удаления циклов выглядит следующим образом.

1. Берем произвольную непросмотренную вершину сети i_0 . Если все вершины просмотрены, то заканчиваем работу. Объявляем i_0 последней вершиной построенной последовательности вершин.

2. Пусть i_K последняя вершина построенной последовательности.

2.1. Пусть $A(i_K)$ пуст. Объявляем i_K просмотренной.

Если $K \neq 0$, то объявляем i_{K-1} последней вершиной последовательности; дугу (i_{K-1}, i_K) удаляем из списка $A(i_{K-1})$ и повторяем шаг 2.

Если $K = 0$, то переходим к шагу 1.

2.2. Пусть $A(i_K)$ не пуст. Обозначим через i_{K+1} конечную вершину 1-й дуги списка $A(i_K)$: (i_K, i_{K+1}) .

Если вершина i_{K+1} ранее не принадлежала построенной последовательности, то объявляем i_{K+1} последней вершиной последовательности и повторяем шаг 2.

Если $i_{K+1} = i_l$, $l \leq K$, то построен ориентированный цикл. Переходим к шагу 3.

3. Вычисляем $\varepsilon = \min \beta(i_j, i_{j+1})$ – минимум величин потоков на цикле. На всех дугах цикла производим пересчет потоков:

$$\beta(i_j, i_{j+1}) = \beta(i_j, i_{j+1}) - \varepsilon.$$

Если на дуге (i_j, i_{j+1}) поток стал равен нулю, то удаляем ее из списка $A(i_j)$ и исключаем эту дугу из сети.

Объявляем i_l последней вершиной построенной последовательности и переходим к шагу 2.

В следующем пункте будет доказано, что после окончания работы этого алгоритма получается ациклическая сеть и для привязки поставщиков и потребителей можно пользоваться ранее описанным алгоритмом.

4. Прежде всего заметим, что при построении в первом алгоритме цепи, благодаря ациклическости сети, мы всегда достигнем тупиковой вершины за конечное число шагов. Ясно, что отсюда легко следуют конечность описанного алгоритма. Докажем теперь, что множество построенных цепей удовлетворяет требованиям, сформулированным в п. 2.

Лемма 2. В описанном алгоритме началом построенной цепи может быть только поставщик, а концом – только потребитель.

Доказательство. Пусть $Q^+(i) = \sum_{A(i)} \beta(u)$, а $Q^-(i) = \sum_{B(i)} \beta(u)$. Если i_0 – исходная вершина цепи, то $\deg(i_0) = 0$ и, следовательно, множество входящих в i_0 дуг $B(i_0)$ пуст. Отсюда следует, что $Q^+(i_0) > 0$, $Q^-(i_0) = 0$ и $Q(i_0) > 0$, а также $Q^-(i_K) > 0$, $Q^+(i_K) = 0$ и $Q(i_K) < 0$, т.е. мощность исходной вершины положительна, а конечной – отрицательна. Заметим, что после пересчета потоков на цепи мощности внутренних вершин не меняются, мощность i_0 уменьшается на ε , а мощность i_K увеличивается на ε . Но $Q(i_0) = Q^+(i_0)$ остается неотрицательной, а $Q(i_K) = -Q^-(i_K)$ уже положительной, так как $\varepsilon \leq \min(\beta(i_0, i_1), \beta(i_K, i_{K-1}))$, а $\beta(i_0, i_1) \leq Q^+(i_0)$. Следовательно, знаки Q после пересчета сохраняются. Так как началом цепи всегда является вершина с положительной, а концом – вершина с отрицательной мощностью, то из того, что мощность сохраняет знак при преобразовании потоков, непосредственно следует утверждение леммы.

Лемма 3. Сумма потоков по всем цепям, выходящим из поставщика, равна мощности этого поставщика.

Доказательство. Пусть вершина i_0 – поставщик. Из леммы 2 следует, что i_0 не может быть конечной вершиной цепи. Значит прежде чем i_0 станет начальной вершиной, она может быть только промежуточной. Следовательно, в момент, когда i_0 стала начальной вершиной, мощность i_0 по сравнению с началом работы алгоритма не изменилась и равна сейчас $Q^+(i_0)$. Пусть C_1, C_{l+1}, \dots, C_m – цепи, началом которых является i_0 . Если i_0 стала начальной вершиной цепи, то другая вершина может стать началом цепи только когда $A(i_0)$ будет исчерпан. Однако, если $A(i_0)$ исчерпан, то $Q^+(i_0) = 0$. Следовательно, $\sum_{d=1}^m \varepsilon(C_d) = Q(i_0)$, что и требовалось доказать.

Лемма 4. Сумма потоков по всем цепям, заканчивающимся в потребителе, равна по абсолютной величине мощности.

Доказательство. Во-первых, сумма потоков на заканчивающихся в потребителе цепях не может по абсолютной величине превышать мощность потребителя, так как, по

лемме 2, потребитель может быть только концом цепи, а знаки мощности не меняются. Из определения следует, что сумма мощностей всех вершин сети равна нулю, значит $\sum_{Q(i)>0} Q(i) = \sum_{Q(i)<0} |Q(i)|$. Но, по лемме 3 $\sum_{Q(i)>0} Q(i) = \sum \varepsilon(C_j)$.

Так как сумма потоков на цепях, заканчивающихся в потребителе, не превосходит по абсолютной величине его мощности, то из равенства $\sum_{Q(i)<0} |Q(i)| = \sum \varepsilon(C_j)$ следует, что для каждого потребителя $|Q(i)| = \sum_{K(C_j)=i} \varepsilon(C_j)$.

Лемма доказана.

Приведем теперь полное обоснование первого алгоритма. Из лемм 2-4 непосредственно следует справедливость условий (1) и (2) для построенного множества цепей. Условие (3) выполняется потому, что каждый раз, назначая поток по цепи, мы соответственно уменьшаем потоки на дугах. Таким образом, для построенного множества цепей выполняются все три условия привязки.

Далее необходимо доказать, что после выполнения второго алгоритма будет получена ациклическая сеть. Докажем, что через пространственную вершину не может проходить ориентированный цикл. У просмотренной вершины список A пуст, поэтому достаточно доказать, что через дугу, выбрасываемую из списка A , цикл не проходит. Докажем это утверждение индукцией по порядку выбрасывания дуг. Дуга выбрасывается из списка, если поток на ней стал равен нулю или если она входит в вершину с пустым списком A . В первом случае очевидно, что никакой цикл по этой дуге пройти не может, так как дуга с нулевым потоком исключается из сети. Во втором случае, если это первая выбрасываемая дуга, то она входит в вершину, из которой вообще нет выходящих дуг, т.е. никакой цикл через эту дугу не пройдет. Если же это не первая выбрасываемая дуга, то она входит в вершину с пустым списком A , но через дуги, ранее выброшенные из списков, циклы не проходят по предположению индукции. Следовательно, циклы не проходят и через нашу дугу. Таким образом, доказано, что через просмотренную вершину циклы не проходят. Так как в конце работы алгоритма все вершины просмотрены, то результатом работы является ациклическая сеть.

5. Также необходимо остановиться на оценке трудоемкости алгоритмов. Из описания первого алгоритма следует, что при построении одной цепи мы работаем только с принадлежащими этой цепи дугами. На обработку дуги требуется конечное число операций. Следовательно, общую трудоемкость алгоритма можно оценить величиной порядка $O(n) + O(C_1)$, где n – число вершин сети, а C_1 – сумма длин всех построенных цепей. Слагаемое $O(n)$ появляется из-за того, что мы тратим некоторое количество действий на обработку первой вершины цепи; при этом цепь может и не быть построена. Трудоемкость второго алгоритма оценивается величиной порядка $O(p) + O(C_2)$, где p – число дуг сети, а C_2 – сумма длин всех уничтожаемых циклов. Слагаемое $O(p)$ появляется потому, что мы расходует некоторое количество действий на удаление дуг из списков A , а слагаемое $O(C_2)$ – потому что на каждой дуге построенного цикла (и только на них) мы выполняем некоторые действия. Из самого вида этих оценок ясно, что они близки к оптимальным. Отметим, что после пересчета потоков на цепи в первом алгоритме и на цикле во втором, хотя бы на одной дуге поток становится равным нулю. Таким образом, число цепей, получаемых в первом алгоритме, и циклов, уничтожаемых во втором, можно оценить сверху величиной p . Отсюда следует, что $C_1 \leq n, p$ и $C_2 \leq n, p$ и оба алгоритма имеют полиномиальную оценку трудоемкости относительно числа вершин и дуг сети.

Приведем теперь более точную оценку количества цепей, получившихся при работе первого алгоритма.

Пусть в сети n вершин, p дуг, s поставщиков и t потребителей.

Теорема. Число цепей не превосходит $p - n + s + t$.

Доказательство. Заметим, что после окончания работы алгоритма потоки на всех дугах сети равны нулю. Каждой внутренней вершине i сопоставим цепь $C_{j(i)}$ после

пересчета потоков, на которой $Q^+(i)$ становится равной нулю. Но, так как вершина $Q^-(i)$ становится равной нулю. Но, так как вершина i внутренняя, то $Q(i) = 0$, а значит и $Q^-(i) = 0$.

Следовательно, после пересчета потоков цепи $C_{j(i)}$ для любой внутренней вершины i поток станет равен нулю по крайней мере на двух дугах. После пересчета потоков на других цепях поток становится равным нулю, хотя бы одной дуге. Следовательно, общее количество цепей можно оценить величиной $p - n_{BH}$, где n_{BH} – число внутренних вершин. Теперь из равенства $n_{BH} = n - s - t$ следует справедливость утверждения теоремы.

Следствие. Если исходная сеть является деревом, то число цепей не превосходит $s + t - 1$.

4. Заключение

При использовании приведенных алгоритмов для привязки поставщиков к потребителям на рынке БРИКС алгоритмов после решения сетевых транспортных задач следует преобразовать полученный оптимальный план к виду, необходимому для работы алгоритмов привязки. Здесь мы отходим от принятых способов хранения оптимального плана в целях единообразия обработки решений транспортных задач с ограничениями пропускной способности и без ограничений. Отметим, что алгоритм удаления циклов следует применять только к решению задачи с ограничениями и только в том случае, когда в исходной сети есть ориентированные циклы из дуг нулевой стоимости.

Отметим, что для задачи без ограничений достаточно запомнить лишь начало цепи, конец цепи и поток по ней, так как маршрут перевозки однозначно восстанавливается по этим данным и по дереву оптимального плана, тогда как для задачи с ограничениями необходимо запоминать весь маршрут перевозки; здесь может быть, например, несколько различных цепей с одинаковыми началами и концами.

Описанный алгоритм привязки осуществляет закрепление поставщиков за потребителями на рынке БРИКС без учета каких-либо требований к этому закреплению. В некоторых случаях такое закрепление не является удовлетворительным.

Литература

1. *Липатова Н.Г., Черныш А.Я.* Применение математических методов при проведении диссертационных исследований. – Москва: Российская таможенная академия, 2011. – 514 с.
2. *Анисимов В.Г., Анисимов Е.Г.* Метод решения одного класса задач целочисленного программирования // Журнал вычислительной математики и математической физики. – 1989. – Т. 29. – № 10. – С. 1586-1590.
3. *Алексеев О.Г.* Модели распределения средств поражения в динамике боя / О.Г. Алексеев [и др.]. - Ленинград: Министерство обороны СССР. – 1989. – 109 с.
4. *Анисимов В.Г., Анисимов Е.Г.* Алгоритм оптимального распределения дискретных неоднородных ресурсов на сети // Журнал вычислительной математики и математической физики. – 1997. – Т. 37. – № 2. – С. 54-60.
5. *Анисимов В.Г., Анисимов Е.Г.* Оптимизационная модель распределения возобновляемых ресурсов при управлении экономическими системами // Вестник Российской таможенной академии. – 2007. – № 1. – С. 49-54.
6. *Губенко А.В., Сазыкин А.М.* Теоретические основы создания систем поддержки принятия решений в интересах комплексной транспортной безопасности // Известия Российской академии ракетных и артиллерийских наук. – 2015. – № 3 (88). – С. 10-15.
7. *Алексеев, А.О.* Применение двойственности для повышения эффективности метода ветвей и границ при решении задачи о ранце / А.О. Алексеев, О.Г. Алексеев [и др.] // Журнал вычислительной математики и математической физики. – 1985. – Т. 25. – № 11. – С. 1666-1673.

8. Anisimov V.G. The model and the planning method of volume and variety assessment of innovative products in an industrial enterprise / V.G. Anisimov [и др.] // Journal of Physics: Conference Series, 2017, 803(1), 012006. DOI: 10.1088/1742-6596/803/1/012006.

9. *Анисимов Е.Г.* Методические положения математического моделирования задач адаптивного распределения дискретных ресурсов при управлении войсками и оружием в режиме реального времени / Е.Г. Анисимов [и др.] // Известия Российской академии ракетных и артиллерийских наук. – 2016. – № 1 (91). – С. 32-37.

10. Zotova E. A model for setting up development programs for logistics systems in the electric power industry to achieve electric power security / E. Zotova [и др.] // В сборнике: E3S Web of Conferences. Сер. "Ural Environmental Science Forum "Sustainable Development of Industrial Region", UESF 2021" 2021, с. 2027. DOI: 10.1051/e3sconf/202125802027.

11. *Черныш А.Я., Мельник Д.А.* Модель поддержки принятия решений при формировании программ инновационного развития предприятий электротехнической отрасли машиностроения // Вестник Российского экономического университета имени Г.В. Плеханова. – 2021. – Т. 18. – № 4 (118). – С. 140-151.

12. *Ильин И.В.* Математические методы и инструментальные средства оценивания эффективности инвестиций в инновационные проекты / И.В. Ильин [и др.]. – Санкт-Петербург, 2018. – 289 с.

13. Anisimov V., Shaban A., Anisimov E., Saurenko T., Yavorsky V. Optimization of perishable goods delivery in supply chains with random demand // В сборнике: E3S Web of Conferences. Сер. "Ural Environmental Science Forum "Sustainable Development of Industrial Region", UESF 2021" 2021. С. 2017. DOI: 10.1051/e3sconf/202125802017.

14. *Сауренко Т.Н.* Модель сравнительной оценки морских торговых портов в международной транспортной инфраструктуре / Т. Н. Сауренко [и др.] // Ученые записки Крымского федерального университета имени В.И. Вернадского. Экономика и управление. – 2020. – Т. 6. – № 1. – С. 153-160.

15. *Родионова Е.С.* Модель и метод календарного планирования логистических процессов перерабатывающих предприятий агропромышленного комплекса / Е.С. Родионова [и др.] // Управленческое консультирование. – 2018. – № 11 (119). – С. 109-118.

16. *Пак А.Ю., Чупина Ж.С., Чупин А.Л., Фаттяхетдинова В.Р., Шобекова Ш.У.* Методический подход к выбору комплекса средств для автоматизации операций транспортного контроля в автомобильных пунктах пропуска через таможенную границу ЕАЭС // Экономические стратегии ЕАЭС: проблемы и инновации. Сборник материалов III Всероссийской научно-практической конференции. – 2020. – С. 151-163.

17. *Ведерников Ю.В.* Модели и алгоритмы интеллектуализации автоматизированного управления диверсификацией деятельности промышленного предприятия // Вопросы оборонной техники. Серия 16: Технические средства противодействия терроризму. – 2014. – № 5-6 (71-72). – С. 61-72.

18. *Тебекин А.В.* Методический подход к моделированию процессов формирования планов инновационного развития предприятий // Журнал исследований по управлению. – 2019. – Т. 5. – № 1. – С. 65-72.

19. Anisimov V.G. Methodological approaches to accounting uncertainty when planning logistic business processes / V.G. Anisimov, E.G Anisimov [и др.] // В сборнике: Atlantis Highlights in Computer Sciences. Proceedings of the International Conference on Digital Technologies in Logistics and Infrastructure (ICDTLI 2019). 2019. С. 246-249.

20. *Барабанов В.В.* Метод оптимизации решений по организации логистических процессов // В сборнике: Логистика: современные тенденции развития: Материалы XVII Международной научно-практической конференции. – 2018. – С. 52-56.

21. Saurenko T.N.. Formalization of planning procedureproduction process of the complex industrial patterns of vertical integration / T.N. Saurenko, M.R Gapov [и др.] // Экономические стратегии ЕАЭС: проблемы и инновации: Сборник материалов

Всероссийской научно-практической конференции.- Москва: Российский университет дружбы народов, 2018. С. 154-161.

22. *Босов Д.Б.* Сетевые модели и методы ресурсно-временной оптимизации в управлении инновационными проектами. – Москва, 2006. – 117 с.

23. *Гапов М.Р., Сауренко Т.Н.* Модель поддержки принятия решений при формировании товарной стратегии и производственной программы предприятия // Вестник Российского университета дружбы народов. Серия: Экономика. – 2016. – № 2. – С. 62-73.

24. *Черныш А.Я., Мельник Д.А.* Модель поддержки принятия решений при формировании программ инновационного развития предприятий электротехнической отрасли машиностроения // Вестник Российского экономического университета имени Г.В. Плеханова. – 2021. – Т. 18. – № 4 (118). – С. 140-151.

25. Anisimov V., Anisimov E., Sonkin M. A resource-and-time method to optimize the performance of several interrelated operations // International Journal of Applied Engineering Research. 2015. Т. 10. № 17. С. 38127-38132.

26. *Николаев Г.А., Ничипор В.И., Осипенков М.Н.* Типовые модели и алгоритмы задач поддержки принятия решений при управлении обеспечивающим компонентом военной организации государства. – Москва: Военная академия Генерального штаба Вооруженных сил Российской Федерации, 2019. – 141 с.

27. *Тебекин А.В., Анисимов В.Г., Анисимов Е.Г.* Нелинейная модель оптимизации параметрических рядов в системах управления // Вестник Российской таможенной академии. – 2015. – № 3. – С. 115–122.

28. *Кежаев В.А.* Обоснование решений в задачах целераспределения с использованием инновационных технологий дискретного программирования / В.А. Кежаев [и др.] // Известия Российской академии ракетных и артиллерийских наук. 2012. № 2 (72). – С. 97-103.

29. *Черныш А.Я., Чечеватов А.В.* Оптимизационные модели и методы в управлении инновационными процессами. – Москва, 2006. – 96 с.

30. Chupin A.L., Chupina Z.S., Morozova N.N., Vorotyntseva T.M., Levinsky E.V. Prediction model of the efficacy and the implementation time of transportation intelligent systems IOP Conference Series: Materials Science and Engineering. II International Scientific Practical Conference "Breakthrough Technologies and Communications in Industry and City", ВТСИ 2019. 2020. С. 012006.

31. *Тебекин А.В.* Логистика. / А. В. Тебекин. – Москва: Дашков и К°, 2014. – 354 с.