

Стратегическое и операционное планирование в задачах логистики

Р. С. Валеев • С. С. Зайнуллин • А. И. Хамитов • М. Ю. Федоров • А. Ф. Валеева

Уфимский университет науки и технологий

Рассматривается ряд логистических задач, а именно: задача стратегического планирования для предприятия, выпускающего некоторую продукцию с целью нахождения эффективного производственного плана, и задача маршрутизации (задача операционного планирования), позволяющая определить наилучшие маршруты для доставки произведенной продукции различным клиентам с учетом их размещения в автомобильные транспортные средства. Перечисленные задачи предлагается интегрировать в систему SAP TM, являющейся одной из систем автоматизированной системы управления ресурсами предприятия SAP ERP. Представлены математические модели, алгоритмы рассматриваемых задач, приведен практический пример.

Задача стратегического планирования; задача операционного планирования; маршрутизация; SAP TM; SAP ERP.

ВВЕДЕНИЕ

На предприятиях, связанных с производством и реализацией некоторой произведенной продукции, возникает целый ряд логистических задач, эффективное решение которых позволило бы удовлетворить спрос на нее точно в срок [Val24, Shay25, Vox23, Bit25]. Каждая такая задача является достаточно трудоемкой и предусматривает решение множества подзадач. В работах [Lan05, Van20] рассматриваются методики принятия решений в таких задачах, в частности, в [Lan05] предложено рассмотреть три класса: задачи стратегического, сетевого и операционного планирования.

Задачи стратегического планирования, как правило, включают в себя разработку плана предприятия на некоторый период, цель которого определение условий, факторов, позволяющих предприятию получить наибольшую прибыль, и удовлетворить спрос клиентов. На основе решений стратегического планирования решаются задачи второго класса, а именно: проектирование, строительство дополнительных объектов и т. п. Третий класс задач – задачи операционного планирования – составляют оптимизационное ядро, представляющее ряд оптимизационных задач: прогнозирование объемов выпускаемой продукции; управление запасами; управление поставками и закупками; поиск оптимальных маршрутов доставки заказов клиентам; оптимальное размещение заказов внутри транспортных средств (ТС), в складских помещениях.

На рис. 1 приведена схема принятия логистических решений, в которой, в отличие от схемы в работе [Lan05], добавлены блоки с задачами 1–7. Рассмотрим некоторые из них. Так, в блоке 1 решается задача управления проектами в рамках бюджета и времени; в блоке 2 осуществляется контроль качества производимой продукции; в блоке стратегического планирования решается задача поиска наилучшего плана предприятия, выпускающего некоторую

продукцию; в блок 3 предлагается включить CRM систему для управления информационной базой заказчиков и вендоров, а в блок 4 – WMS-систему для автоматизации управления складскими процессами; в блоке 5 решаются вопросы сбыта и реализации произведенной продукции для оформления заказов клиентов и формирования доходной документации, а в блоке 6 – вопросы финансов и контроллинга для управления бухгалтерией и затратами; в блок 7 включены системы TSW (Trader's and Scheduler's Workbench) и TM (Transportation Management) для планирования и управления транспортировкой; в блоке транспортировка решается задача доставки заказов клиентам наилучшими маршрутами.

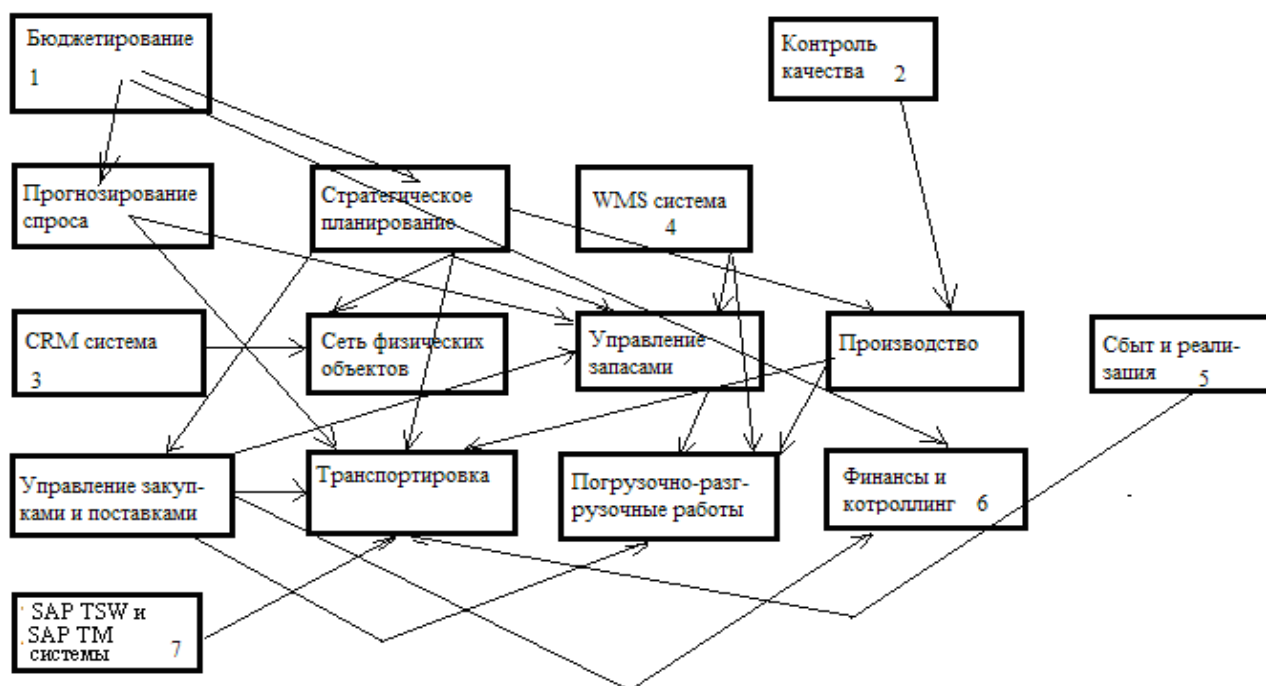


Рис. 1. Схема принятия логистических решений

Статья посвящена решению задачи стратегического планирования – поиску плана выпуска производимой продукции с наибольшей прибылью (блок стратегического планирования, рис. 1), а также задаче операционного планирования – нахождению наилучших маршрутов для доставки произведенной продукции различным клиентам с учетом ее размещения в автомобильные ТС (блок Транспортировка, рис. 1). Рассматриваемые задачи предлагается интегрировать в автоматизированную систему управления ресурсами предприятия SAP ERP, а именно в SAP TSW и SAP TM (блок 7, рис. 1). Система SAP ERP состоит из системы управления складами SAP WMS¹; системы управления взаимоотношениями с клиентами SAP CRM²; систем SAP TSW и SAP TM (управление отгрузками и поставками)³ [Kuz19, Grz19]. Результаты работы системы SAP TSW обеспечивают транспортные потребности отдельных отраслей экономики при доставке продукции клиентам. Основными задачами системы SAP TM являются следующие: планирование и управление транспортом; построение оптимальных маршрутов и мониторинг транспорта; проведение конкурсов на выбор перевозчика по заданным критериям; управление заказами на фрахт (бронирование и расчет фрахта); расчет стоимости перевозок в каждом звене цепочки поставок; моделирование тарифов на перевозки и экспедирование; планирование и контроль загрузки транспортных средств на складе; контроль исполнения доставок с учетом промежуточных точек маршрута.

¹ <https://www.itctg.ru> (дата обращения 15.04.2026).

² <https://crm-systems.info> (дата обращения 15.04.2026).

³ <https://www.unicon-consulting.ru> (дата обращения 15.04.2026).

ЗАДАЧА СТРАТЕГИЧЕСКОГО ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА НЕКОТОРОЙ ПРОДУКЦИИ

Целью любого предприятия, производящего какую-либо продукцию, является получение наибольшей прибыли, поэтому задача планирования производства продукции нацелена на получение эффективного производственного плана на некоторый плановый период, позволяющий предприятию реализовать эту цель. Рассмотрим предприятие, выпускающее некоторую продукцию, которую оно доставляет различным клиентам – оптовым торговым организациям.

Задача планирования производства некоторой продукции (OPER_PLAN)

Предприятие производит некоторую продукцию, которая в ходе производственного цикла должна пройти k процессов обработки в определенном порядке, причем каждый процесс обработки проходит независимо от предыдущих $k - 1$ процессов. Оно получает в единицу времени некоторое число заказов на выпускаемую им продукцию. При этом заранее нельзя предсказать точное число заказов на свою продукцию, спрос на нее, выпуск продукции в единицу времени каждым из k процессов. Предприятие принимает все заказы в течение планового периода.

Требуется:

- получить такой план выпуска производимой продукции для руководства предприятия на некоторый плановый период, который бы позволил получить наибольшую прибыль;
- найти маршрут доставки произведенной продукции различным клиентам с минимальными транспортными расходами автомобильными ТС с учетом ее размещения в минимальное количество ТС.

Реализация первого требования – это одна из задач стратегического планирования. Рассматриваемый производственный цикл для выпуска некоторой продукции представляет собой систему массового обслуживания (СМО) с устройствами, соединенными последовательно (устройства – это оборудования для реализации каждого процесса, требования – поступающие заказы клиентов). При этом заказы прибывают в случайные моменты времени, и представляет собой неоднородный пуассоновский процесс с функцией интенсивности прибытия заказов $\lambda(t)$, зависящей от времени t поступления заказа. Для реализации данного производственного цикла со случайными параметрами предлагается имитационный алгоритм *OPER_PLAN_IM*, базирующийся на алгоритме, предложенном в [Ней75, She23].

Имитационный алгоритм OPER_PLAN_IM

Входные данные:

T – продолжительность планового периода;

NS – количество производственных процессов;

NP – количество планов;

$price$ – цена единицы произведенной продукции;

C – полные затраты предприятия;

NO – количество заказов;

Q – количество выполненных заказов в течение периода T ;

$E(D_t)$ – ожидаемый спрос на число заказов D_t , получаемых предприятием в момент времени t , $t = 0, \dots, T$;

$E(Q_j)$ – ожидаемые темпы производства выпускаемой продукции Q_j j -м процессом, $j = 1, \dots, NS$; N – количество реплик.

Состояние предприятия: $(AT_i, PT_{ij}, WT_{ij}, FT_{ij}, DT_{ij})$;

Случайные переменные:

AT_i , $i = 1, \dots, NO$ – время поступления i -го заказа;

PT_{ij} – время обработки i -го заказа в j -м процессе, $i = 1, \dots, NO$, $j = 1, \dots, NS$;

WT_{ij} – время ожидания очереди i -м заказа в j -м процессе, $i = 1, \dots, NO$, $j = 1, \dots, NS$;

DT_{ij} – время, в течение которого простаивает j -й процесс в ожидании i -го заказа, $i = 1, \dots, NO$, $j = 1, \dots, NS$;

FT_{ij} – время, складывающееся из времени ожидания очереди i -м заказом в j -м процессе и времени простоя j -го процесса в ожидании i -го заказа, $i = 1, \dots, NO$, $j = 1, \dots, NS$: $FT_{ij} = WT_{ij} + DT_{ij}$.

Реальная прибыль $profit$ предприятия: $profit = price \cdot Q - C$, ожидаемая прибыль $profit_w$:

$$profit_w = price \cdot \left((E(D_t) \cdot T - \sum_{j=1}^{NS} \frac{\frac{E(D_t)}{E(Q_j)}}{1 - \frac{E(D_t)}{E(Q_j)}}) \right) - C, t = 0, \dots, T.$$

События: поступление заказа для обработки j -м процессом, выпуск заказа после завершения j -го процесса.

Параметры, используемые в процессе работы алгоритма:

$time$ – время выполнения заказа;

DI_j – параметр, определяющий простой j -го процесса, либо ожидание обработки заказа в j -м процессе: если $DI_j > 0$, то j -й процесс ожидает обработку, иначе j -й процесс простаивает.

Выходные данные:

$profit$ – средняя прибыль предприятия за плановый период T .

Шаг 1. Ввод данных $T, NS, NP, price, C, E(D_t), E(Q_j)$.

Шаг 2. Инициализация параметров: $NO, N, time, DI_j$.

Шаг 3. Цикл по $p = 1, \dots, NP$.

Шаг 3.1. Цикл по $L = 1, \dots, N$.

Шаг 3.1.1. $AT = \text{Генерация_прибытия_заказов}()$.

Шаг 3.1.2. $NO = \text{Length}(AT)$.

Шаг 3.1.3. $\text{Состояние_предприятия_для } i = 1 \text{ заказа}()$.

Шаг 3.1.4. $\text{Состояние_предприятия_для } i = 2, \dots, NO \text{ заказов}()$.

Шаг 3.1.5. Вычисление реальной прибыли $profit_L = price \cdot Q - C$.

Шаг 3.2. Вычисление ожидаемой прибыли $profit_w$.

Шаг 3.3. Вычисление средней прибыли для каждого плана

$$profit = \frac{\sum_{L=1}^N profit_L}{N}.$$

Рассмотрим процедуры, применяемые в алгоритме $OPER_PLAN_IM$.

Процедура Генерация_прибытия_заказов():

Вход: T, λ – интенсивность прибытия заказов.

Выход: $AT_i, i = 1, \dots, NO$.

Шаг 1. $t = 0$.

Шаг 2. Пока $t \leq T$ выполнить:

Шаг 2.1. Сгенерировать случайное число $U_1 = \text{Uniform}(0,1)$.

Шаг 2.2. $t = t - 1/\lambda \cdot \ln U_1$.

Шаг 2.3. Сгенерировать случайное число $U_2 = \text{Uniform}(0,1)$

Шаг 2.4. Если $U_2 \leq \frac{\lambda(t)}{\lambda}$, то $AT = t$.

Шаг 2.5. Иначе перейти к шагу 2.

Шаг 3. Сформировать список времен поступления заказов AT

Данная процедура представляет собой генерацию времен поступления заказов (неоднородный пуассоновский процесс) с интенсивностью $\lambda > 0$, функцией интенсивности $\lambda(t)$, причем $\lambda(t) \leq \lambda, t \in [0, T]$, которая выбирается эмпирически.

Процедура Состояние_предприятия_для i = 1 заказа():

Цикл по $j = 1, \dots, NS$ выполнить:

Шаг 1. Если $time \leq T$ то выполнить:

Шаг 1.1. $PT_{1j} = \text{Генерация_времени_обслуживания}()$.

Шаг 1.2. $WT_{1j} = 0$.

Шаг 1.3. $FT_{1j} = PT_{1j}$.

Шаг 1.4. Если $j = 1$ то $DT_{11} = 0$ иначе $DT_{1j} = DT_{1(j-1)} + PT_{1(j-1)}$.

Шаг 2. $Q_j = Q_j + 1$.

Процедура Генерация_времени_обслуживания():

Вход: $E(Q_j), j = 1, \dots, NS$.

Выход: $time$.

Шаг 1. Сгенерировать случайное число $U = Uniform(0,1)$.

Шаг 2. Возвратить величину $time = -E(Q_j) \cdot \ln U$.

Процедура Состояние_предприятия_для $i = 2, \dots, NO$ заказов():

Цикл по $i = 2, \dots, NO$ выполнить:

Шаг 1. Инициализация $DI_j, j = 1, \dots, NS$.

Шаг 2. Цикл по $j = 1, \dots, NS$ выполнить:

Шаг 2.1. Если $time \leq T$ то выполнить:

Шаг 2.1.1. $PT_{ij} = \text{Генерация_времени_обслуживания}()$.

Шаг 2.1.2. $DI_j = DI_j + AT_{i-1} - AT_i$.

Шаг 2.1.3. Цикл по $k = 1, \dots, j - 1$.

$DI_j = DI_j + PT_{(i-1)k} - WT_{ik} - PT_{ik}$.

Шаг 2.1.4. $DI_j = DI_j - WT_{ij} - PT_{ij}$.

Шаг 2.1.5. Если $DI_j > 0$ то выполнить:

$DT_{ij} = 0,$

$WT_{ij} = DI_j,$

иначе

$DT_{ij} = -DI_j,$

$WT_{ij} = 0.$

Шаг 2.1.6. $FT_{ij} = WT_{ij} + PT_{ij}$.

Шаг 2.2. $Q_j = Q_j + 1$.

При получении эффективного оперативного плана по выпуску продукции (задача *OPER_PLAN*) предприятие приступает к решению задач операционного планирования. В статье рассматривается задача поиска наилучшего маршрута доставки готовой продукции.

ЗАДАЧА ОПЕРАЦИОННОГО ПЛАНИРОВАНИЯ

Рассмотрим задачу доставки готовой продукции различным клиентам, предварительно размещенной в автомобильные транспортные средства.

Задача поиска маршрута для доставки готовой продукции с учетом ее размещения в автомобильные ТС (реализация требования 2 задачи *OPER_PLAN*)

Имеется депо с парком автомобильных ТС с различной грузоподъемностью, расходами топлива, размерами кузова; список видов произведенной продукции, размещенной в коробки параллелепипедной формы с известными размерами и весами; список клиентов с их спросами; расстояния между клиентами. Требуется найти маршрут доставки готовой продукции различным клиентам с минимальными транспортными расходами автомобильными ТС с учетом ее размещения в ТС при выполнении следующих условий: маршрут каждого ТС должен начинаться и заканчиваться в депо; каждый клиент может иметь спрос, превышающий грузоподъемность ТС (клиент может быть посещен более одного раза, т.е. допускается раздельная доставка продукции [Kon20, Tot14, Tan21]); коробки с продукцией разрешается ставить друг на друга внутри грузового отсека; учитывается центр тяжести ТС при размещении в нем коробок с продукцией; коробки должны находиться на опоре (не допускается провисание коробок); размещение коробок в ТС осуществляется в порядке, обратном маршрутному листу.

Математическая модель задачи OPER_PLAN (реализация требования 2)

Входные данные:

$G = (V, E)$ – взвешенный граф, в котором V – множество вершин - клиентов, E – множество ребер (участков автомобильных дорог);

$V = \{i_0, i_1, \dots, i_n\}$, где i_0 – депо, i_1, \dots, i_n – клиенты; d_{ij} – расстояние между клиентами i и j , где $i, j \in V$;

$C = \{c_1, \dots, c_j, \dots, c_k\}$, где c_j – различные виды продукции, k – количество коробок с произведенной продукцией; c_{w_j} – ширина j -й коробки; c_{h_j} – высота j -й коробки; c_{l_j} – длина j -й коробки; c_{wt_j} – вес j -й коробки;

Q_i^c – объем спроса i -го клиента в продукции вида c , $i \in V - \{i_0\}$, $c \in C$; m – число автомобильных ТС v ; a_{w_v} – ширина кузова ТС v ; a_{h_v} – высота кузова ТС v ; a_{l_v} – длина кузова ТС v ; a_{f_v} – расход топлива ТС v ; a_{load_v} – грузоподъемность ТС v .

Введем следующие обозначения:

x_{ij}^v – переменная логического типа, принимающая 1, если ТС v перемещается от клиента i к клиенту j и 0 в противном случае, где $i, j \in V$; y_{ic}^v – количество продукции вида c , доставляемого ТС v клиенту i ;

(px_j, py_j, pz_j) – координаты левого нижнего дальнего угла j -й коробки параллелепипедной формы, $j \in C$;

$(px_{jk}, py_{jk}, pz_{jk})$ – координаты k -й вершины j -й коробки;

$(px_{jz,m}, py_{jz,m}, pz_{jz,m})$ – координаты центра тяжести j -й коробки по осям x, y, z , $j \in C$;

$(X_{vz,m}, Y_{vz,m}, Z_{vz,m})$ – координаты центра тяжести размещенных коробок в ТС v .

Требуется минимизировать транспортные расходы:

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^m (d_{ij} \cdot x_{ij}^v \cdot a_{f_v}) \rightarrow \min \quad (1)$$

при выполнении следующих ограничений:

- каждый клиент должен быть посещен ТС v один или более раз (раздельная доставка):

$$\sum_{i=1}^n \sum_{v=1}^m x_{ij}^v \geq 1, j = 1, \dots, n; \quad (2)$$

- число входов в графе G в вершину должно равняться числу выходов из нее:

$$\sum_{i=0}^n x_{iz}^v - \sum_{j=0}^n x_{zj}^v = 0, z = 0, \dots, n; v = 1, \dots, m; \quad (3)$$

- исключаются замкнутые подмаршруты:

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^v \leq |S| - 1, v = 1, \dots, m; S \subseteq V - \{i_0\}; \quad (4)$$

- клиент i может быть обслужен ТС v , если v проходит через клиента i :

$$y_{ic}^v \leq Q_i^c \sum_{j=1}^n x_{ij}^v, i = 1, \dots, n; v = 1, \dots, m; c = 1, \dots, k; \quad (5)$$

- спрос каждого клиента должен быть удовлетворен:

$$\sum_{v=1}^m y_{ic}^v = Q_i^c, i = 1, \dots, n; v = 1, \dots, m; c = 1, \dots, k; \quad (6)$$

- спрос клиента не должен превышать грузоподъемности ТС v :

$$\sum_{i=1}^n \sum_{c=1}^k y_{ic}^v \cdot c_{wt_c} \leq a_{load_v}, v = 1, \dots, m; \quad (7)$$

- ребра коробок ортогональны граням грузового отсека:

$$\begin{aligned} & ((px_{ik} = px_i) \vee (px_{ik} = px_i + c_{l_i})) \wedge ((py_{ik} = py_i) \\ & \vee (py_{ik} = py_i + c_{w_i})) \wedge ((pz_{ik} = pz_i) \vee (pz_{ik} = pz_i + c_{h_i})); \end{aligned} \quad (8)$$

- коробки не выходят за грани грузового отсека:

$$(px_i \geq 0) \wedge (py_i \geq 0) \wedge (pz_i \geq 0) \wedge ((px_i + c_{w_i}) \leq a_{w_v}) \wedge ((py_i + c_{l_i}) \leq a_{l_v}) \wedge ((pz_i + c_{h_i}) \leq a_{h_v}); \quad (9)$$

- коробки не перекрывают друг друга:

$$(px_i \geq px_j + pw_j) \vee (px_j \geq px_i + pw_i) \vee (py_i \geq py_j + pl_j) \vee (py_j \geq py_i + pl_i) \vee (pz_i \geq pz_j + ph_j) \vee (pz_j \geq pz_i + ph_i); \quad (10)$$

- коробки должны находиться на опоре (не допускается провисание коробок):

$$((pz_i + ph_i = pz_i) \wedge (px_i < px_j + c_{w_j}/2) \wedge (px_i + c_{w_i} > px_j + c_{w_j}/2) \wedge (py_i < py_j + c_{l_j}/2) \wedge (py_i + c_{l_i} > py_j + c_{l_j}/2)); \quad (11)$$

- отклонение центра тяжести загруженного ТС v от центра (точки пересечения диагоналей) должно быть минимальным:

$$\text{dev}_v = \sqrt{(X_{v_{ц.т.}} - a_{w_v}/2)^2 + (Y_{v_{ц.т.}} - a_{l_v}/2)^2 + (Z_{v_{ц.т.}} - a_{h_v}/2)^2} \rightarrow \min, \quad (12)$$

где

$$X_{v_{ц.т.}} = \sum_{j=1}^k (px_{j_{ц.т.}} \cdot c_{wt_j}), \quad Y_{v_{ц.т.}} = \sum_{j=1}^k (py_{j_{ц.т.}} \cdot c_{wt_j}), \\ Z_{v_{ц.т.}} = \sum_{j=1}^k (pz_{j_{ц.т.}} \cdot c_{wt_j})$$

(см. рис. 2);

- количество ТС с размещенными в них коробками должно быть минимальным:

$$m \rightarrow_P \min, \quad (13)$$

где P – множество возможных позиций для размещения коробок в ТС.

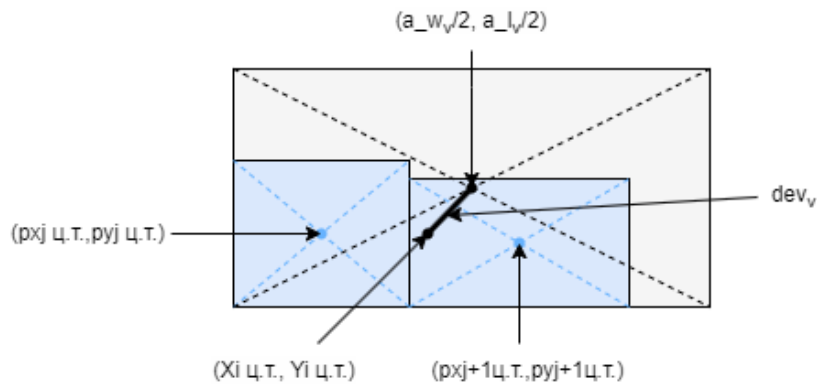


Рис. 2. Иллюстрация отклонения от центра тяжести в ТС v

Для решения задачи (1)–(13) был разработан алгоритм на базе эволюционного алгоритма $(\mu + \lambda) - EA$ (для размещения коробок с произведенной продукцией в грузовой отсек автомобильного ТС) и имитации отжига (для поиска маршрутов доставки коробок с продукцией различным клиентам) [Els20, Szu87]. Метаэвристика «имитация отжига» моделирует процесс отжига металлов, при котором твердое тело нагревается до высокой температуры, далее начинается процесс медленного охлаждения путем понижения температуры до тех пор, пока не произойдет процесс кристаллизации. Приведем общую схему алгоритма *решения задачи OPER_PLAN* (реализация требования 2). Решение задачи представляет собой объект *State*,

состоящий из следующих элементов: список ТС v , $route_v = \{i_0, i_1, \dots, i_n, i_0\}$ – маршрут ТС v , i_0 – депо, $i \in V$; $items_v = \{c_1, \dots, c_n\}$ – список видов продукции, доставляемых ТС v , $c \in C$.

Общая схема алгоритма OPER_PLAN_PV для решения задачи OPER_PLAN

Шаг 1. Инициализация:

Шаг 1.1. $i = 1$ //номер итерации.

Шаг 1.2. $t_i = t_{max}$ //задать начальную «температуру».

Шаг 1.3. t_{min} //задать минимальную «температуру».

Шаг 1.4. $State_{best} = null$ //лучшее решение.

Шаг 1.5. $State_{new} = null$ // новое решение, полученное из текущего путем случайных изменений.

Шаг 1.6. ev_count //количество итераций эволюционного алгоритма.

Шаг 1.7. $evAlgType$ //номер эволюционного алгоритма.

Шаг 1.8. μ, λ // число «потомков» и число «родителей».

Шаг 2. $State_{best} = GetInitialState()$ //генерация начального решения – маршрута

Шаг 3. Если $State_{best} = null$, то конец алгоритма (не удалось получить решение)

Шаг 4. $StateCost = GetStateCost(State_{best})$ //вычисление целевой функции (1)

Шаг 5. $solutionReceivedFlag = false$ // флаг для проверки существования допустимого решения с учетом условий (2)–(7).

Шаг 6. Пока $t_i > t_{min}$, делать:

Шаг 6.1. $State_{new} = GetNewState(State_{best})$ // генерация нового решения.

Шаг 6.2. $State_{new} = PackingProcedure(ev_count, evAlgType, State_{new})$ // размещение коробок в ТС согласно маршруту, полученному на шаге 6.1.

Шаг 6.3. Если $State_{new} \neq null$, то

Шаг 6.3.1. $\Delta E = GetStateCost(State_{new}) - StateCost$ // разница между стоимостями нового и предыдущего решения по (1).

Шаг 6.3.2. Если $GetTransitionProbability(\Delta E, t_i) \geq random(0,1)$, то

Шаг 6.3.2.1. $State_{best} = State_{new}$ // новое решение по (14).

Шаг 6.3.2.2. $StateCost_{new} = StateCost_{new}$.

Шаг 6.4. $t_i = T_{Cauchy}(t_{max}, i)$ // понижение «температуры».

Шаг 6.5. $i = i + 1$.

Шаг 7. Если $solutionReceivedFlag = false$, то конец алгоритма (решение не получено), иначе вернуть решение $State_{best}$

$$GetTransitionProbability(\Delta E, t_i) = \begin{cases} 1, & \text{если } \Delta E \leq 0, \\ e^{-\frac{\Delta E}{t_i}}, & \text{если } \Delta E > 0, \end{cases} \quad (14)$$

$$T_{Cauchy}\left(t \frac{t_{max}}{(1+i)^{max}}\right). \quad (15)$$

Рассмотрим подробнее процедуры, приведенные в алгоритме.

Процедура GetInitialState()

Шаг 1. Создать пустое решение $State$.

Шаг 2. Ввод переменных: $m, n, k, Q_i^c, c_{w_j}, c_{h_j}, c_{l_j}, c_{wt_j}, a_{w_v}, a_{h_v}, a_{l_v}, a_{load_v}$.

Шаг 3. Пока $v < m$, выполнить цикл по ТС v :

Шаг 3.1. Добавить в маршрут депо.

Шаг 3.2. Пока $i < n$, выполнить цикл по клиентам.

Шаг 3.2.1. Пока $c < k$, выполнить цикл по коробкам с продукцией.

Шаг 3.2.1.1. Пока $h < Q_i^c$ выполнить:

Шаг 3.2.1.1.1. $newLoad = a_{load_v} - c_{wt_c}$

// оставшаяся грузоподъемность ТС с учетом веса коробки.

Шаг 3.2.1.1.2. $newVolume = (a_{w_v} \cdot a_{h_v} \cdot a_{l_v}) - (c_{w_c} \cdot c_{h_c} \cdot c_{l_c})$

// оставшийся объем кузова ТС.

Шаг 3.2.1.1.3. Если $newLoad \geq 0$ и $newVolume \geq 0$,

то загрузить коробку в ТС v ,

добавить в маршрут клиента, если его еще не было,

иначе $v = v + 1$ // переход к следующему ТС.

Если $v = m$, то сообщение «не хватает ТС»,

иначе перейти на шаг 3.

Шаг 3.2.1.1.4. $h = h + 1$ // увеличение спроса.

Шаг 3.2.1.2. $c = c + 1$ // переход к следующей коробке с продукцией.

Шаг 3.2.2. $i = i + 1$ // переход к следующему клиенту;

Шаг 3.3. Добавить в маршрут депо.

Шаг 3.4. $v = v + 1$ // переход к следующему ТС.

Шаг 4. Возвратить решение $State$.

Процедура $GetNewState(State_{best})$

Шаг 1. Сформировать решение $State_{new}$.

Шаг 2. $State_{best} \rightarrow State_{new}$.

Шаг 3. $probability = 0,5$.

Шаг 4. Если $probability \geq random(0,1)$, то $CustomerExchange(State_{new})$

// обмен спросами клиентов между ТС (рис. 3, а),

иначе $CustomerTransfer(State_{new})$

// передать спрос клиента другому ТС (рис. 3, б).

Шаг 5. Возвратить решение $State_{new}$.

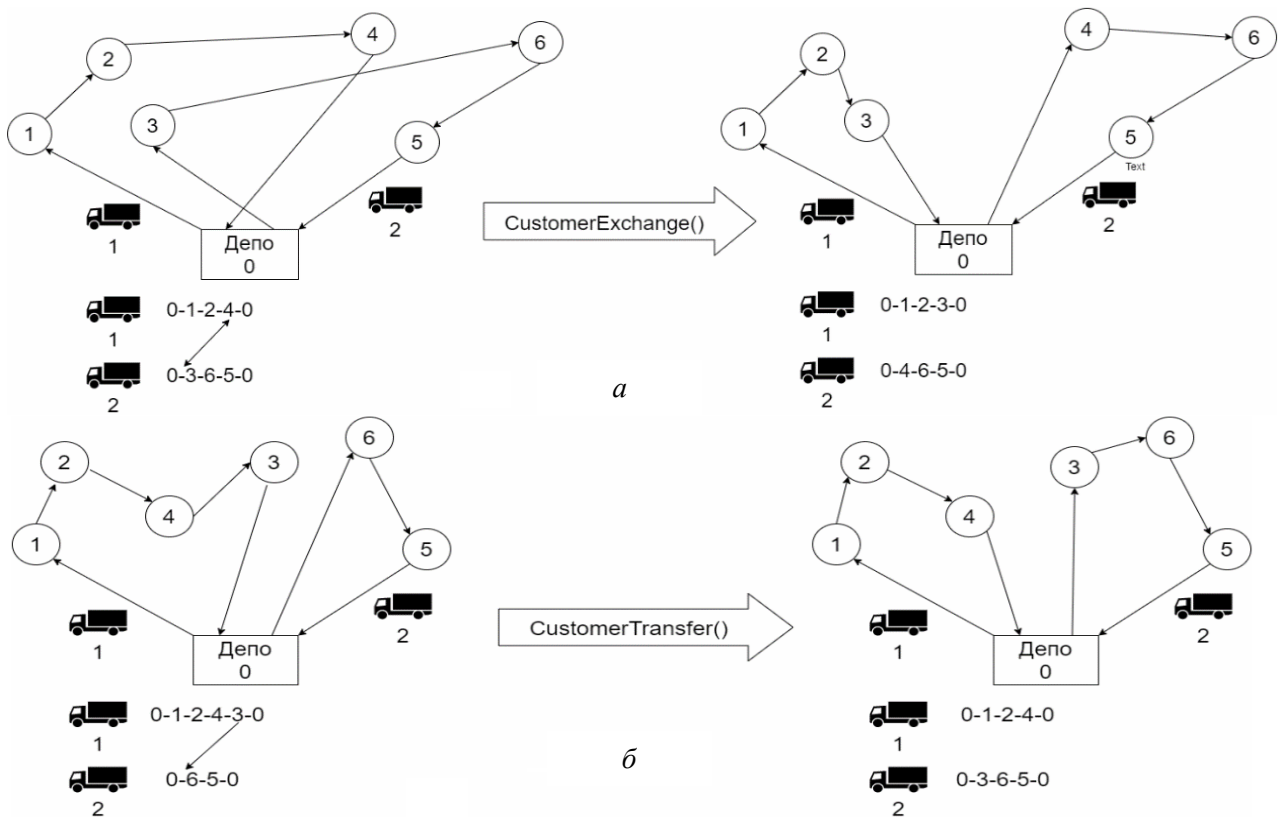


Рис. 3. Иллюстрация процедуры $CustomerExchange(State_{new})$ (а); процедуры $CustomerTransfer(State_{new})$ (б)

Процедура $PackingProcedure(ev_count, evAlgType, State_{new})$

Шаг 1. $State_{best} = State$

Шаг 2. Для $t = 1$ до ev_count выполнить:

Шаг 2.1. $State_{new} = State_{best}$

Шаг 2.2. Для каждого ТС $\in State$:

Шаг 2.2.1. Если $evAlgType = 0$, то $items_k = Mutation1(items_k)$,

// перестановка одной пары элементов в $State$

иначе

Шаг 2.2.2. Если $evAlgType = 1$, то $items_k = Mutation2(items_k)$
 // перестановка двух пар элементов в $State$.

Шаг 2.3. $State_{new}.packingCost = shared_ABLP(State_{new})$.

Шаг 2.4. Если $State_{new}.packingCost = newState.packingCost = 0$,
 то перейти к шагу 2.1.

Шаг 2.5. Если $State_{new}.packingCost < State_{best}.packingCost$,
 то $State_{best} = State_{new}$.

Шаг 2.6. Перейти к шагу 2.1.

Процедура $shared_ABLP(State_{new})$ (алгоритм-декодер)

Вход: Список ТС.

Выход: $StateCost$ – значение целевой функции (1).

Шаг 1. Для каждого ТС v :

Шаг 1.1. Поместить коробку в левый нижний дальний угол отсека ТС v ,
 совпадающий с началом координат $(0, 0, 0)$.

Шаг 1.2. Для каждой неразмещенной i -й коробки выполнить:

Шаг 1.2.1. Для каждой размещенной j -й коробки выполнить:

Шаг 1.2.1.1. Сформировать список возможных позиций P
 для размещения i -й коробки относительно размещенной
 j -й коробки.

Шаг 1.2.1.2. Удалить из списка P позиции, не удовлетворяющие
 ограничениям (8)–(13).

Шаг 1.2.2. Если список P не пустой, то

Шаг 1.2.2.1. Поместить i -ую коробку в лучшую позицию из списка P .

Шаг 1.2.2.2. Перейти к следующей коробке,

иначе

Шаг 1.2.3. Выйти из алгоритма с сообщением: «Не удалось найти допустимую
 позицию размещения коробки».

Шаг 2. Вычислить значение целевой функции $StateCost$ (1).

Шаг 3. Возвратить $StateCost$.

На рис. 4 представлена иллюстрация работы декодера $shared_ABLP(State_{new})$.

Лучшая позиция в процедуре $shared_ABLP(State_{new})$ при размещении коробок послойно определяется как минимум координат pz_j^k, py_j^k, px_j^k , где k – номер позиции (рис. 5, а), а при размещении коробок в колонну – как минимум координат py_j^k, px_j^k, pz_j^k (рис. 5, б).

ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Для реализации имитационного алгоритма $OPER_PLAN_IM$, позволяющего выполнить первое требование задачи $OPER_PLAN$, разработано программное обеспечение на языке программирования C# на платформе .NET Framework в среде программирования Microsoft Visual Studio 2019. Была проведена верификация программной реализации этого алгоритма при следующих входных данных: продолжительность планового периода $T = 10$, количество производственных процессов $NS = 2$, число заказов 7.

В результате работы программы в течение $T = 10$ получены времена прибытия каждого заказа: 0.53, 0.92, 1.28, 4.3, 4.42, 5.1, 8.8; времена обслуживания на первом устройстве каждого заказа: 1.12, 2.06, 3.55, 1.56, 1.8, 2.82, 0.54; времена обслуживания на втором устройстве каждого заказа: 0.66, 0.87, 0.94, 0.5, 1.56, 1.63, 0.87; общее время обслуживания на двух устройствах каждого заказа: 1.78, 2.93, 4.49, 2.06, 2.67, 4.45, 1.41. При этом общее среднее время, в течение которого заказы обрабатывались в системе, оказалось равным 2.827 у.е. Расчеты, проведенные вручную, показали, что времена, проведенные заказами в системе, равны 1.76, 2.88, 4.51, 2.1, 2.59, 4.42, 1.5, а общее среднее время оказалось равным

$$\frac{1.76+2.88+4.51+2.1+2.59+4.42+1.5}{7} = 2.822.$$

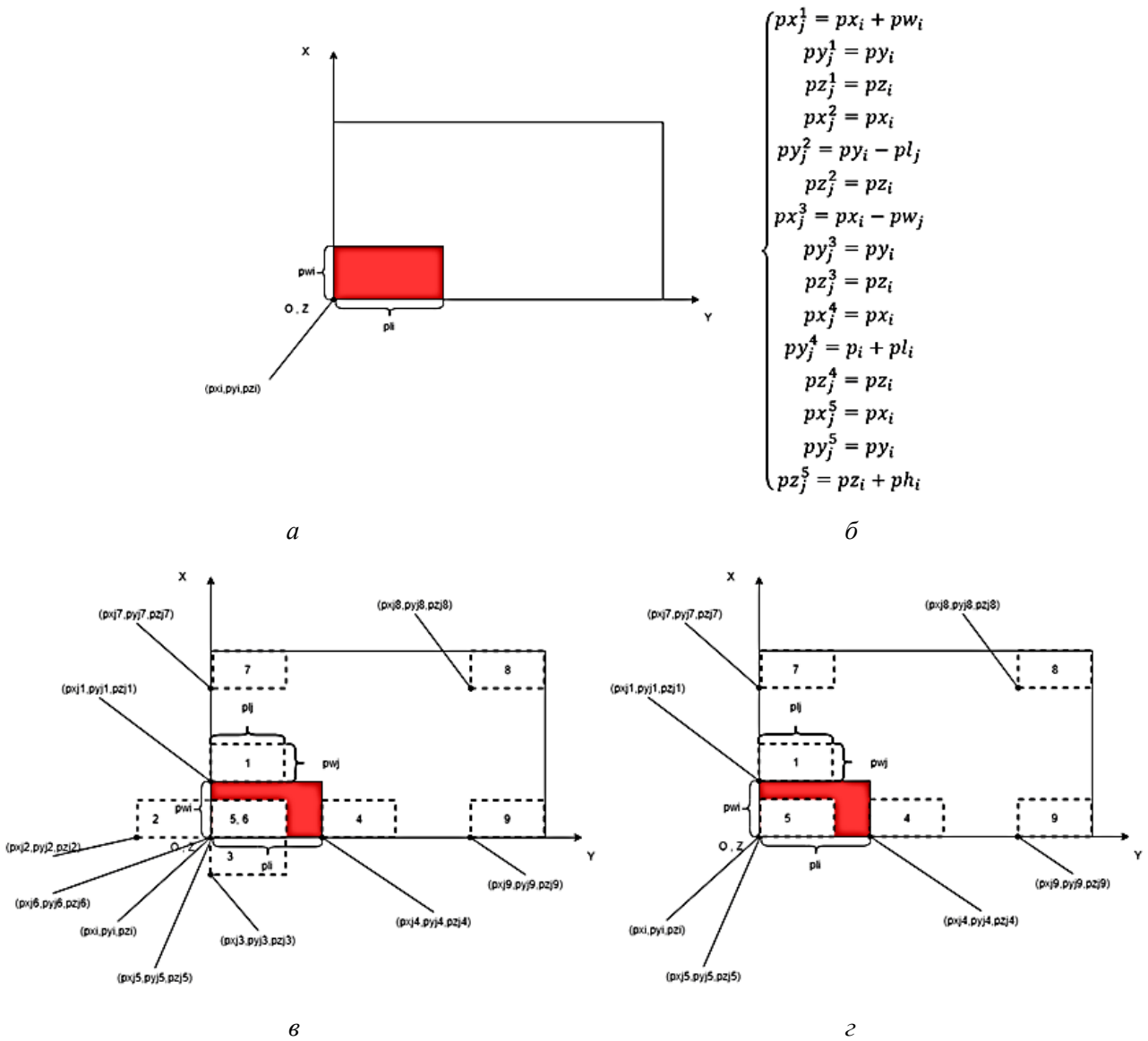


Рис. 4. Иллюстрация работы декодера $shared_ABLP(State_{new})$:
a – размещение первой коробки в начало координат; *б* – формулы для расчета координат возможных позиций; *в* – расчет координат возможных позиций размещения коробок; *г* – допустимые позиции размещения коробок

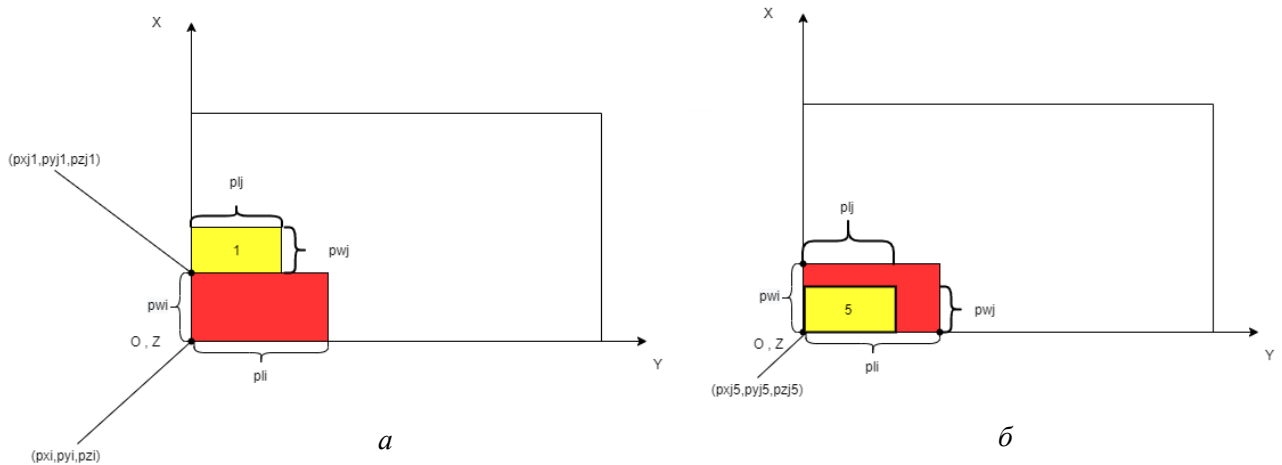


Рис. 5. Размещение коробок в лучшую позицию при послойном размещении (*a*) и при размещении в колонну (*б*)

Таким образом, выходные данные имитации отличаются от результатов, полученных вручную, на 0.005у.е. (в силу округления), что говорит о правильности работы программы, реализующей алгоритм *OPER_PLAN_IM*.

Численные эксперименты проводились на данных предприятия, производящего керамическую плитку, и реализующего заказы на нее различным клиентам (торговым организациям). Производство плитки состоит из пяти основных производственных процессов, иллюстрация которых приведена на рис. 6.



Рис. 6. Иллюстрация производственных процессов при изготовлении керамической плитки

При проведении численных экспериментов было рассмотрено четыре варианта оперативного плана со следующими параметрами: продолжительность планового периода составляла $T = 365$ дней, за единицу заказа принималась одна паллета с $93,8\text{ м}^2$ плитки, стоимость единицы заказа 40000руб, ожидаемые темпы производства $E(Q_j)$, $j = 1, \dots, 4$, (взяты на основе данных эксперта) в день для каждого процесса изготовления плитки приведены на рис. 7.

Параметры алгоритма *OPER_PLAN* для четырех производственных планов

План	Полные затраты, C (руб)	$E(Q_1)$ Формовка	$E(Q_2)$ сушка	$E(Q_3)$ нанесение глазури	$E(Q_4)$ обжиг	$E(Q_5)$ сортировка
1	13000000	0.051	0.41	0.02	0.1	0.02
2	13500000	0.048	0.44	0.023	0.13	0.016
3	12900000	0.045	0.45	0.021	0.09	0.022
4	12500000	0.049	0.4	0.019	0.12	0.02

Рис. 7. Исходная информация для алгоритма *OPER_PLAN*: параметры алгоритма для четырех производственных планов

На рис. 8 приведены результаты моделирования (количество реплик $N = 200$ для каждого плана): среднее время простоя в ожидании заказа, ожидаемая и средняя прибыли; среднее время выполнения процесса «формовка», число поступивших заказов 1192, число выполненных заказов 766, ожидаемая прибыль отличается от средней прибыли на 1.5–2.5%. Наиболее эффективным оказался план 4.

Для реализации алгоритма *OPER_PLAN_PV*, который позволил выполнить второе требование задачи *OPER_PLAN*, разработано программное обеспечение на языке программирования *Golang*, клиентская часть – на языке JavaScript в среде разработки Visual Studio Code, пользовательский интерфейс программы написан с использованием JavaScript библиотеки React, для отображения маршрутов использовалось Open Street Map.

Для поиска рациональных маршрутов в алгоритме *OPER_PLAN_PV* использовались параметры температуры: t_{\min} , t_{\max} . Были проведены численные эксперименты, в результате которых определялось влияние операции обмена клиентами между различными ТС на расход топлива, и, как следствие, на стоимость маршрута. Обмен клиентами из разных маршрутов в случайно выбранные ТС происходил с различными вероятностями (варианты 1 и 2) и с равными вероятностями (вариант 3).

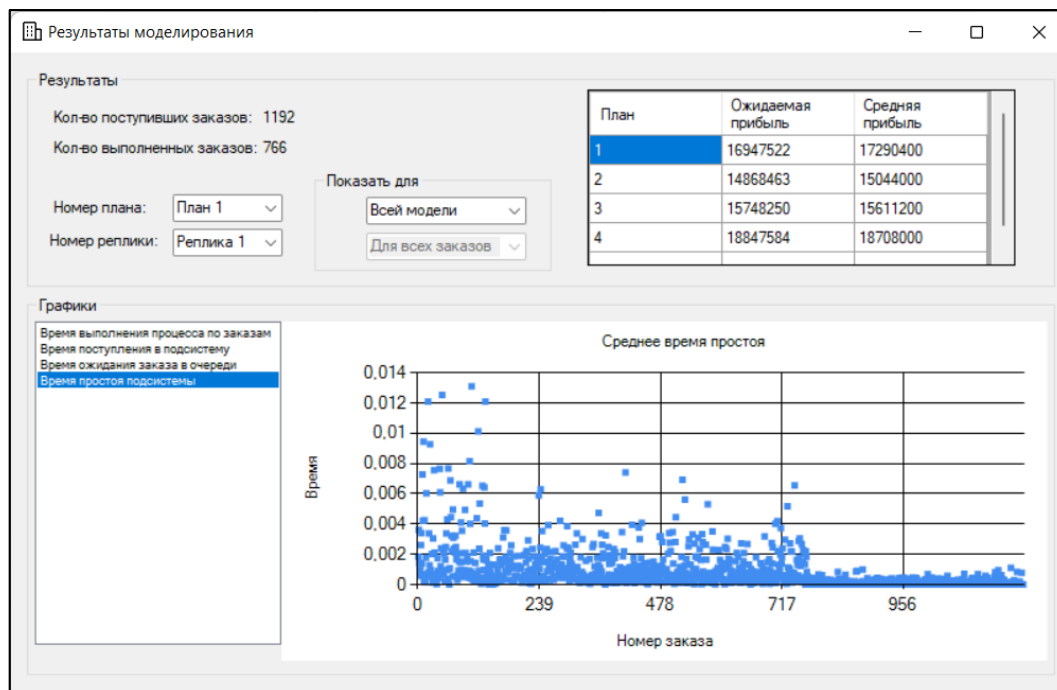


Рис. 8. Результаты моделирования: среднее время простоя в ожидании заказа, ожидаемая и средняя прибыли; среднее время выполнения процесса «формовка»

Для решения задачи размещения коробок с произведенной продукцией внутри грузового отсека автомобильного ТС проводилось тестирование разработанного эволюционного алгоритма $(\mu + \lambda) - EA$ с двумя видами мутаций *Mutation1*, *Mutation2* (шаги 2.2.1 и 2.2.2 в процедуре *PackingProcedure()*) с параметрами $\mu = 1, \lambda = 1$. При этом проводился численный эксперимент для сравнения двух вариантов алгоритмов – декодеров алгоритма $(1+1)-EA$. Приведем один из экспериментов (таблица).

Таблица

Сравнение результатов работы алгоритма $(1+1)-EA$ при 100% заполненности грузового отсека ТС

Алгоритм-декодер	Послойная упаковка				Упаковка в колонну			
	$(1+1)-EA$ <i>Mutation1</i>		$(1+1)-EA$ <i>Mutation2</i>		$(1+1)-EA$ <i>Mutation1</i>		$(1+1)-EA$ <i>Mutation2</i>	
Итерации	100	400	100	400	100	400	100	400
Прогоны								
1	0.9434	0.9719	1.0246	0.9689	1.0177	0.7117	0.8670	0.6761
2	0.9784	0.9541	1.0609	0.9056	1.0719	0.6861	0.9710	0.9683
3	1.2525	0.8549	1.0968	1.0048	1.1172	0.9495	0.9941	1.0617
4	1.0538	0.9654	1.0661	0.7749	1.2085	0.5791	1.5011	1.0935
5	0.9677	0.9243	1.2842	0.9805	1.0225	0.7133	1.1548	0.9299
6	1.0494	0.7490	0.9776	0.9015	1.3942	0.9841	1.0250	0.8319
7	1.0927	0.9884	0.8551	0.9551	0.8884	0.6382	1.0494	0.7701

Целью эксперимента было исследование отклонения центра тяжести загруженного грузового отсека ТС, имеющего параллелепипедную форму, от центра пересечения диагоналей. Было сгенерировано три тестовых набора данных, соответствующих 30% заполнению отсека коробками, 60% заполнению, полностью заполненному грузовому отсеку ТС (100%) со следующими параметрами: длина отсека $L = 80$ у. е., ширина $W = 230$ у. е., высота $H = 24$ у. е.

Параметры коробок варьировались в пределах $80 \leq l \leq 120, 70 \leq w \leq 100, 60 \leq h \leq 100$.

Как видно из результатов, представленных в таблице, лучшие результаты были получены алгоритмом (1+1)-EA с декодером «упаковка в колонну». При 30% заполненности отсека ТС лучшие результаты показал алгоритм (1+1)-EA с декодером «послойная упаковка», при 60% заполненности отсека ТС лучшие результаты показал алгоритм (1+1)-EA с декодером «упаковка в колонну».

Численные эксперименты проводились на данных, взятых с сайта компании Lasselsberger Group⁴, которая занимается производством кафельной плитки для различных клиентов.

Приведем результаты одного из экспериментов, в котором:

1) данные о коробках с размещенными в них различными видами кафельной плитки, следующие (ширина (м) × высота(м) × длина (м) × вес (кг)):

- керамогранит «Вестанвинд» имеет размеры $0,2 \times 0,3 \times 0,6 \times 18$;
- керамогранит «Блюм» – $0,3 \times 0,3 \times 0,6 \times 27$;
- керамогранит «Лофт Стайл» – $0,45 \times 0,3 \times 0,45 \times 27$;

2) данные об автомобильных ТС (название, количество, грузоподъемность (кг),

- размеры: ширина (м) × высота (м) × длина (м),
- расход топлива (л/100 км): микроавтобус, 3, 3000, $2 \times 2 \times 3$, 10;

3) данные о клиентах и их спросах:

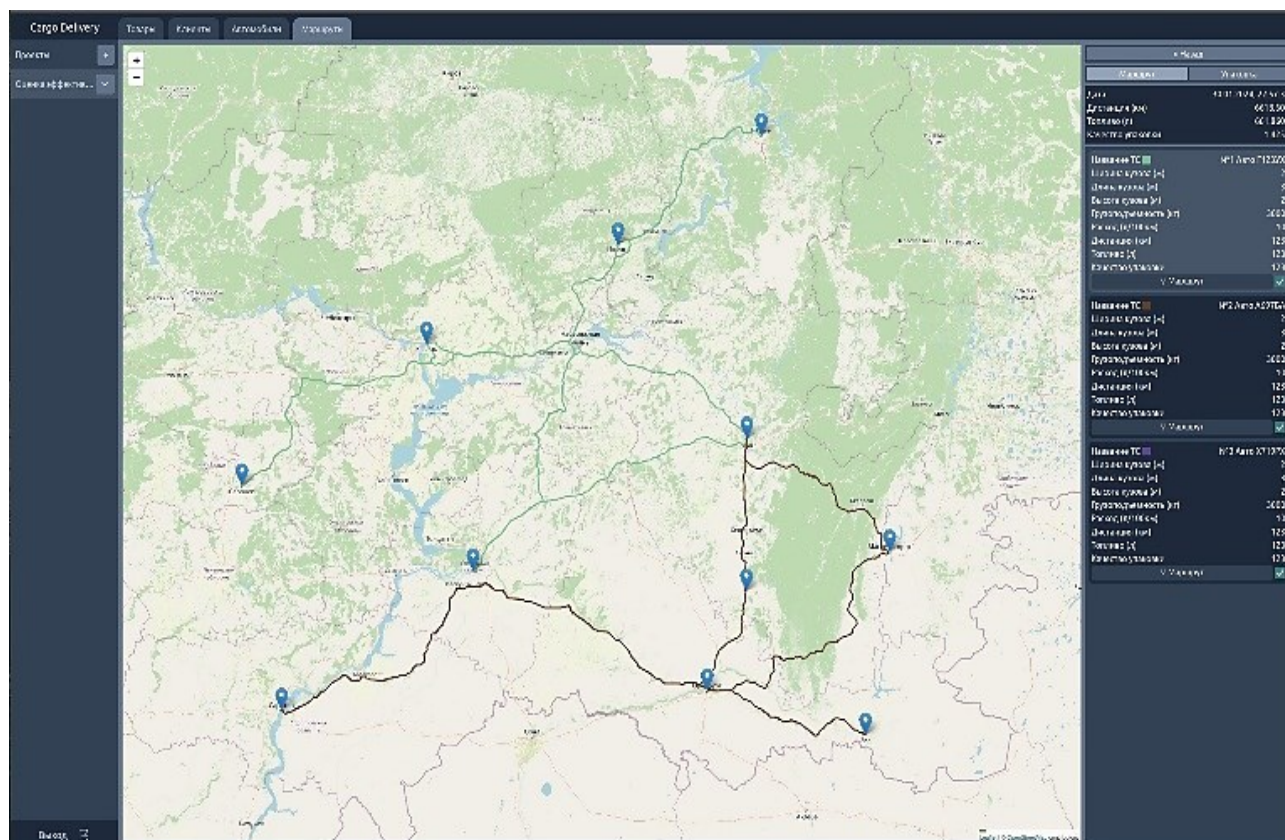
- Оренбург: керамогранит «Вестанвинд», 5 шт;
- Самара: керамогранит «Вестанвинд», 43 шт;
- Казань: керамогранит «Блюм», 24 шт;
- Мелеуз: керамогранит «Блюм», 7 шт; керамогранит «Вестанвинд», 10 шт;
- Саранск: керамогранит «Блюм», 7 шт; керамогранит «Вестанвинд», 13 шт;
- Пермь: керамогранит «Вестанвинд», 21 шт;
- Ижевск: керамогранит «Вестанвинд», 11 шт;
- Саратов: керамогранит «Лофт Стайл», 6 шт;
- Орск: керамогранит «Вестанвинд», 21шт;
- Магнитогорск: керамогранит «Блюм», 17шт; керамогранит «Вестанвинд», 24шт.

В среднем, маршруты с наименьшим расходом топлива, а, следовательно, наименьшей стоимости, были получены в варианте 1 операции обмена клиентов.

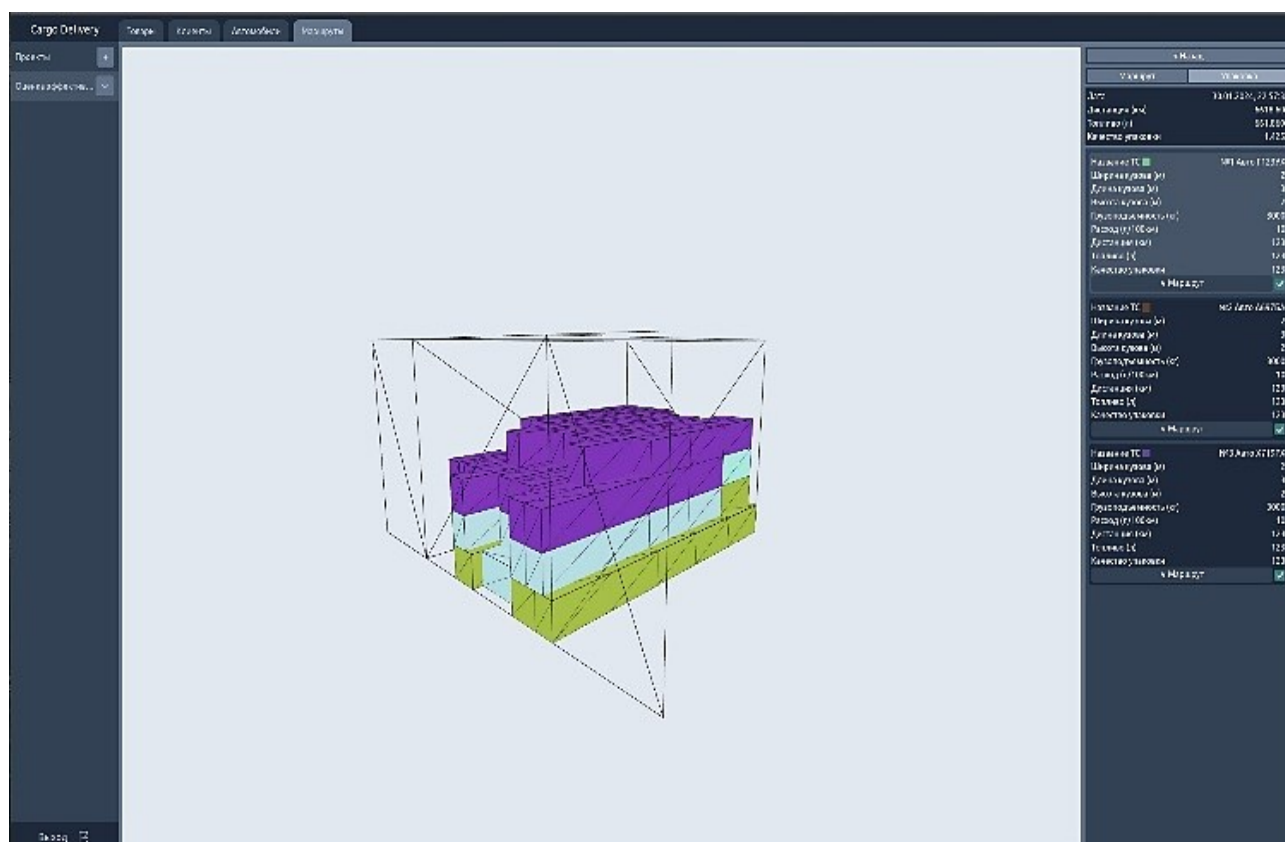
Для поиска рационального размещения коробок с кафельной плиткой в алгоритме OPER_PLAN_PV был реализован эволюционный алгоритм (1+1)-EA с двумя видами мутаций: перестановка одной пары элементов (мутация №1) и перестановка двух пар элементов (мутация № 2). При этом рассматривались различные виды размещений коробок: послойное размещение и размещение в колонну. Исследовалось отклонение центра тяжести упакованного коробками ТС от центра (точки пересечения диагоналей). На рис. 9 приведен фрагмент разработанного программного обеспечения с найденными наилучшими маршрутом и размещением коробок с плиткой.

Проведенные численные эксперименты показали, что интеграция разработанных алгоритмов OPER_PLAN и OPER_PLAN_PV в системы SAP TM, SAP TSW, являющейся частью системы SAP ERP, позволит эффективно спланировать производство для получения наибольшей прибыли и доставить заказы клиентам точно в срок наилучшими маршрутом и размещением заказов клиентов внутри автомобильных ТС.

⁴ LB Ceramics. URL: <https://lb-ceramics.ru> (дата посещения 15.04.2026).



а



б

Рис. 9. Пример рациональных маршрутов (данные с сайта <https://lb-ceramics.ru>, депо в Уфе) (а); пример размещения коробок с плиткой в грузовой отсек автомобильного ТС (б)

ЗАКЛЮЧЕНИЕ

В статье была предложена схема принятия логистических решений, включающая актуальные и трудные логистические задачи, одна из которых связана с выбором эффективного плана развития производства некоторой продукции, другая – с ее доставкой различным клиентам наилучшими маршрутами с учетом различных ограничений. Приведены математические модели и алгоритмы решаемых задачах. Поскольку задача выбора наилучшего производственного плана содержит случайные параметры, для ее решения был разработан имитационный алгоритм, базирующийся на известном алгоритме, описывающем деятельность фирмы с k производственными процессами для выпуска некоторой продукции. При этом были решены задачи маршрутизации и размещения трехмерных объектов (коробок с заказами клиентов) в ТС, являющиеся NP-трудными задачами комбинаторной оптимизации. Для их решения были разработаны алгоритмы на базе известных метаэвристик. Как показали численные эксперименты на реальных данных и оценки экспертов, приведенные алгоритмы при интеграции в автоматизированную систему управления ресурсами предприятия SAP ERP, позволят предприятию получить в среднем наибольшую прибыль.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

- [Bit25] Bitto B., Karthigai P. Ch. AI's Impact on Vehicle Growth and Ownership Patterns in the Automotive Industry // Systems Engineering and Information Technologies. 2025. Vol. 7, No. 3(22). P. 40-47. [QCQXLS](#).
- [Els20] Elshaer R., Award H. A Taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants // Comput. Ind. Eng., 2020, Vol.140, pp. 106-242. [10.1016/j.cie.2019.106242](#).
- [Grz19] Grzeškowiak P., Ciesielski W., Cwin W. Mastering SAP ABAP. Pact Publishing, 2019. 553 p.
- [Kon20] Konstantakopoulos G-D., Gayialis S.P., Kechagias E.P. Vehicle routine problem and related algorithms for logistics distribution: a literature review and classification // Oper. Res. Int. J. 2020, p.1-30. [10.1007/s12351-020-00600-7](#).
- [Kuz19] Kuz Andrea Diaz. Uncover the Secrets of SAP SD. Azul Publishing, 2019. 297 p.
- [Lan05] Langevin A., Riopel D. Logistics Systems: Design and Optimization, USA, Springer, 2005. 388 p.
- [She23] Sheldom M. Ross. Simulation. Elsevier Science, 6 edition. 2023. 336 p.
- [Szu87] Szu H. H., Hartley R. L. Fast simulated annealing // Physical Letters, 1987. No. 122, pp. 157-162.
- [Tan21] Tan S.-Y., Yeh W.-C. The vehicle routing problem: state of the art classification and review // Applied Sciences, 2021, 11, no. 21: 10295. [10.3390/app112110295](#).
- [Tot14] Toth P., Vigo D. Vehicle Routine: Problems, Methods and Applications. Philadelphia, PA, USA: SIAM, 2014. 463 p.
- [Van20] Vandeput N. Inventory Optimization. Models and Simulations. De Gruyter Publishing, 2020. 318 p.
- [Вал24] Валеев Р. С., Юсупов А. Р., Кашаев Ш. З., Валеева А. Ф. Об одной задаче обеспечения эффективного товарооборота дистрибьютерной компании // СИИТ. 2024. Т. 6, № 4(19). С. 38-50. [KSRBGK](#). [[Valeev R. S., Yusupov A. R., Kashaev Sh. Z., Valeeva A. F. On one problem of ensuring effective turnover of a distribution company // SIIT. 2024. Vol. 6, No. 4(19). P. 38-50. (In Russian).]]
- [Вох23] Вохминцев А. В. Методология решения проблемы одновременной навигации и построения карты на основе комбинирования визуальных и семантических характеристик окружающей среды // СИИТ. 2023. Т. 5, № 3(12). С. 136-155. [EFOFFO](#). [[Vokhmintsev A. V. Methodology for solving the problem of simultaneous navigation and map construction based on a combination of visual and semantic characteristics of the environment // SIIT. 2023. Vol. 5, No. 3(12). P. 136-155. (In Russian).]]
- [Ней75] Нейлор Т. Имитационные эксперименты с моделями экономических систем, Москва, Мир, 1975. 500 с. [[Naylor, Thomas H. Computer Simulation Experiments with Models of Economic Systems. New York: Wiley, 1971.]]
- [Шай25] Шаймарданов А. Ф., Сазонова Е. Ю., Сметанина О. Н. Построение маршрутов с учетом спрогнозированного риска дорожно-транспортных происшествий на основе исторических данных // СИИТ. 2025. Т. 7, № 5(24). С. 109-123. [TRRSBA](#). [[Shaimardanov A. F., Sazonova E. Yu., Smetanina O. N. Construction of routes taking into account the predicted risk of road accidents based on historical data // SIIT. 2025. Vol. 7, No. 5(24). P. 109-123. (In Russian).]]

ОБ АВТОРАХ | ABOUT THE AUTHORS

ВАЛЕЕВ Руслан Сагитович

Уфимский университет науки и технологий, Россия.
ruslan_valeev@inbox.ru ORCID: [0000-0002-0550-4676](#).

Ведущий консультант ООО «Вентра Ай-Ти». Доц. каф. вычислительной математики и кибернетики, канд. техн. наук по сист. анализу, управлению и обработке информации.

VALEEV Ruslan Sagitovich

Ufa University of Science and Technology, Russia.

ruslan_valeev@inbox.ru ORCID: [0000-0002-0550-4676](#).

Leading specialist LLC "Ventra IT". Assoc. Prof. of the Department of Computational Mathematics and Cybernetics, Ph.D. in systems analysis, management and information processing.

ЗАЙНУЛЛИН Салават Саянович
ООО «Компания «Тензор», Россия.
salavatzaynullin67@gmail.com
Инженер-программист.

ХАМИТОВ Айдар Ильфатович
ООО «Фродекс», Россия.
aidar607@gmail.com
Инженер-программист.

ФЕДОРОВ Максим Юрьевич
ООО «Озна – Диджитал Солюшнс», Россия.
maxim.fedorov28@yandex.ru
Ведущий инженер-программист.

ВАЛЕЕВА Аида Фаритовна
Уфимский университет науки и технологий, Россия.
aida_val2004@mail.ru. ORCID: 0000-0003-3687-4925.
Проф. каф. вычислительной математики и кибернетики.
Д-р техн. наук, профессор.

ZAYNULLIN Salavat Sayanovich
LLC "Company Tenzor", Russia.
salavatzaynullin67@gmail.com
Software engineer.

KHAMITOV Aidar Ilfatovich
LLC "Frodex", Russia.
aidar607@gmail.com
Software engineer.

FIODOROV Maksim Yurevinch
LLC "Ozna – Digital solutions", Russia.
maxim.fedorov28@yandex.ru
Leading Software engineer.

VALEEVA Aida Faritovna
Ufa University of Science and Technology, Russia.
aida_val2004@mail.ru. ORCID: 0000-0003-3687-4925.
Prof., Computational Mathematics and Cybernetics Dept.
Dr. of Tech. Sci.

МЕТАДААННЫЕ | METADATA

Заглавие: Стратегическое и операционное планирование в задачах логистики.

Авторы: Валеев Р. С., Зайнуллин С. С., Хамитов А. И., Федоров М. Ю., Валеева А. Ф.

Аннотация: Рассматривается ряд логистических задач, а именно: задача стратегического планирования для предприятия, выпускающего некоторую продукцию с целью нахождения эффективного производственного плана, и задача маршрутизации (задача операционного планирования), позволяющая определить наилучшие маршруты для доставки произведенной продукции различным клиентам с учетом их размещения в автомобильные транспортные средства. Перечисленные задачи предлагается интегрировать в систему SAP TM, являющейся одной из систем автоматизированной системы управления ресурсами предприятия SAP ERP. Представлены математические модели, алгоритмы рассматриваемых задач, приведен практический пример.

Ключевые слова: Задача стратегического планирования; задача операционного планирования; маршрутизация; SAP TM; SAP ERP.

Язык: Русский.

Статья поступила в редакцию 28 марта 2026 г.

Title: About certain tasks of the strategic and operational planning when making logistic decisions.

Authors: Valeev R. S., Zainullin S. S., Khamitov A. I., Fedorov M. Yu., Valeeva A. F.

Abstract: A number of logistic problems are considered, namely: the strategic planning problem for an enterprise that produces certain products in order to search the effective production plan, and the routing problem (operational planning problem) which allows to determine the best routes for the delivering of manufactured products to various customers with account of their placement in transport vehicles (TS). The enlisted tasks are proposed to be integrated into the SAP TM system which is one of the systems of the automated enterprise resource management system SAP ERP. The mathematical models and algorithms for the problems under consideration are presented, the practical example is presented.

Key words: Strategic planning problem; the operational planning problem; the routing; SAP TM; SAP ERP.

Language: Russian.

The article was received by the editors on 28 March 2026.