УДК 629.631.554

ДИГІТАЛІЗАЦІЯ В ЛОГІСТИЦІ ЗА РАХУНОК ВВЕДЕННЯ АЛГОРИТМУ МУРАШИНОЇ ЛОГІСТИКИ

Л. А. Савченко

Національний університет біоресурсів і природокористування України, Україна.

Кореспонденція автора: Lilya_savchenko@ukr.net.

Історія статті: отримано – квітень 2019, акцептовано – серпень 2019.

Бібл. 16, рис. 3, табл. 2.

Анотація. Дигіталізація це нове поняття, яке допомагає в транспортній галузі при плануванні маршрутної мережі підвищити їхню ефективність та мінімізувати негативний вплив на навколишнє середовище.

Крім того, цифрові платформи стануть надзвичайно важливими в логістичній галузі, адже дозволять малим компаніям отримати глобальний доступ і конкурувати з великими гравцями.

В межах індустрії логістики та ланцюгів поставок спостерігається безліч втрат через ведення непродуктивної діяльності.

Важливість низьких витрат на транспортування і їх подальший вплив на вартість доставки товарів багаторазово збільшують продуктивність і заощадження компанії при інтеграції рішення VRP, тим самим викорінюючи всі сумніви в актуальності дослідження VRP.

Ключові слова: логістика, дигіталізація, транспортний процес, інформаційні технології, мурашина логістика.

Постановка проблеми

З точки зору економічної теорії, поняття транспортної логістики розглядається як планування оптимальних маршрутів вантажоперевезень з мінімізацією витрат, що є особливо актуальним для торгових компаній, щодня доставляють продукцію покупцям.

У зв'язку зі зростаючою завантаженистю автомобільних доріг, водії підприємств часто запізнюються в встановлені тимчасові інтервали доставки, що призводить до повернень продукції, зростання незадоволеності клієнтів, і, в кінцевому рахунку, до втрат виручки і збитків.

Важливість низьких витрат на транспортування і їх подальший вплив на вартість доставки товарів багаторазово збільшують продуктивність і заощадження компанії при інтеграції рішення VRP, тим самим викорінюючи всі сумніви в актуальності дослідження VRP, незважаючи на тривалість існування цієї проблеми.

Мета досліджень

Впровадження різних алгоритмів для вирішення VRP, перевірши алгоритми з теоретичної моделі в географічну карту міста і оцінити ефективність їх роботи з різними наборами реальних даних і труднощами, що виникають у сфері повсякденних послуг доставки, щоб створити додаток, що застосовує реалізовані алгоритми для створення легких для читання і застосування маршрутних листів і карт доставки товарів.

Результати досліджень

Класичну модель відрізняють такі особливості:

- наявність всього одного складу, з якого починають рух транспортні засоби;
- всі транспортні засоби є гомогенними, тобто мають однакові характеристики;
- транспортні засоби повертаються в початкову точку маршруту;
- маршрути складаються на один день.

Рис. 1. CMDOVRPTW-модель.

Fig. 1. CMDOVRPTW Model.

Натомість, CMDOVRPTW-модель описують такі особливості, які доповнюють класичну VRP-модель:

- транспортні засоби мають обмеження на вантажопідйомність;
- кожному транспортному засобу присвоюються тимчасові рамки, в які воно може здійснювати свою роботу;
- кожного замовлення також присвоюються тимчасові рамки, в які воно може бути доставлено.

Описувана в цій роботі задача є рішенням насиченої моделі CMDOVRPTW (Capacitated Multiple Depot Open Vehicle Routing Problem with Time Windows) для підприємства «Тернопільський м'ясокомбінат» у Тернополі та Тернопільській області.

Необхідно вирішити задачу розподілу наявного набору замовлень на один день по набору автомобілів і скласти маршрутні листи для кожного з кур'єрів. Позначимо через $\overline{1, n} = \{1, 2, \dots, n\}$.

Нехай A – множина автомобілів.

Задамо на множині A індексацію $I: A \rightarrow \overline{1, n}$, где $n = |A|$.

Будемо використовувати позначення a_i для елемента a з множини A , для якого $I(a) = i$. Поставимо у відповідність кожному автомобілю a_i вектор атрибутів, що описують його:

$$(VA_i, VAMP_i, LA_i, LAMP_i, V_i, t_i^s, t_i^f),$$

де VA_i – вантажопідйомність a_i -го автомобіля, кг;

$VAMP_i$ – максимальна вага для одного товару в замовленні, що міститься в a_i -ий автомобіль, кг;

LA_i – місткість a_i -г о автомобіля в довжину, м;

$LAMP_i$ – максимальна довжина для одного товару в замовленні, що міститься в a_i -ий автомобіль, м;

V_i – швидкість пересування a_i -го автомобіля, км / год;

t_i^s – час поява кур'єра на a_i -му автомобілі на складі (час початку роботи кур'єра), г/х/с;

t_i^f – час доставки кур'єром на a_i -му автомобілі останнього за часом замовлення (час завершення роботи кур'єра), г/х/с.

Нехай Z – множина замовлень.

Задамо на множині Z індексацію

$$I: Z \rightarrow \overline{1, m},$$

де $m = |Z|$.

Будемо використовувати позначення z_j для елемента z з множини Z , для якого $I(z) = j$.

Поставимо у відповідність кожному замовленню z_j вектор атрибутів, що описують його:

$$(VZ_j, VZMP_j, LZ_j, LZMP_j, S_j, K_j, T_j^s, T_j^f),$$

де VZ_j – загальна вага всіх товарів в z_j -му замовленні, кг;

$VZMP_j$ – максимальна вага одного товару в z_j -му замовленні, кг;

LZ_j – загальна довжина всіх товарів в z_j -му замовленні, м;

$LZMP_j$ – максимальна довжина одного товару в z_j -му замовленні, м;

S_j – адреса складу розміщення z_j -го замовлення, координати;

K_j – адреса доставки z_j -го замовлення, координати;

T_j^s – верхня часова межа доставки z_j -го замовлення, г/х/с;

T_j^f – нижня часова межа доставки z_j -го замовлення, г/х/с.

Нехай $\wp$ – множина планів доставки замовлень на обраний день. План доставки замовлень складається з

множини різних листів доставки замовлень, що описують маршрут кожного з автомобілів.

$$\wp = \{P_1, P_2, \dots, P_n\},$$

Нехай P – множина різних листів доставки замовлень.

Елемент цієї множини P являє собою множину маршрутних листів.

$$P = \{M_1, M_2, \dots, M_n\},$$

де маршрутний лист M_i є упорядкованим набором $(Z_{p_1^i}, Z_{p_2^i}, \dots, Z_{p_{m_i}^i})$ замовлень, що задовольняють співвідношенню

$$p_k \neq p_j \Rightarrow Z_{p_k^i} \neq Z_{p_j^i}$$

Тут m_i – множина замовлень в маршрутному листі M_i .

Позначимо через $\mathcal{M}_i = \{Z_{p_j^i} | j \in \overline{1, m_i}\}$ множину замовлень маршрутного листа M_i .

Дані множини мають наступні властивості:

1. Всі замовлення повинні бути розподілені за маршрутними листами так, що індекс одного замовлення повинен зустрічатися не більш ніж один раз серед усіх маршрутних листів.

2. Для кожного з автомобілів сумарна вага всіх замовлень цього автомобіля не повинен перевищувати вантажопідйомність автомобіля. [дод.1]

3. Для кожного з автомобілів сумарна довжина всіх замовлень цього автомобіля не повинна перевищувати встановлену загальну довжину замовлень автомобіля.

4. Для кожного замовлення кожного з автомобілів максимальна вага будь-якого товару не повинен перевищувати допустимий показник ваги одного товару цього автомобіля.

5. Для кожного замовлення кожного з автомобілів максимальна довжина будь-якого товару не повинна перевищувати допустимий показник довжини одного товару цього автомобіля.

6. Для кожного автомобіля розрахований час доставки останнього замовлення не повинно перевищувати встановлений час роботи даного кур'єра.

7. Для кожного замовлення кожного з автомобілів розрахований час доставки замовлення повинен знаходитися в межах інтервалу часу доставки, встановленого для цього замовлення.

Постановки завдання можуть бути описані такими способами в залежності від застосовуваного критерію оптимізації:

1. Найменший час доставки усіх замовлень

$$K(P_0) = \max_{i \in \overline{1, n}} (T_i^s + \frac{dis_i(m_i)}{V_i})$$

2. Найменша кількість кур'єрів, використаних для доставки замовлень

$$isListEmpty(M_i) = \begin{cases} 1, & |M_i| = \emptyset \\ 0, & \text{у решті випадків} \end{cases}$$

Алгоритми локального пошуку (Local Search Algorithms) полягають у випадковій ініціалізації початкового рішення і поступового перетворення даного рішення у все більш і більш оптимальне. На кожному циклі виконання алгоритму береться поточне рішення і розглядається набір сусідніх рішень (рішення в околиці, neighborhood solutions), в

які поточне рішення може бути переведено за допомогою одного елементарного зміни (рух, move). З рухів вибирається найбільш оптимальний (що дає максимальне значення оціночної функції, що відповідає обраному критерію оптимізації - мінімальний час доставки замовлень, мінімальне число задіяних транспортних засобів, мінімальне покриття відстань). Якщо воно задовольняє критеріям (є прийнятним, acceptable), властивим конкретному алгоритму локальної оптимізації, то поточне рішення замінюється на вбрання оптимальне з околиці (крок, step) і цикл виконання алгоритму запускається заново для нового рішення.

Локальні алгоритми мають загальний вигляд, описаний в схемі 1.

Input: Cost() – function, determining cost of the solution, which algorithm is trying to maximize
Output: solution X_{best}
 1: **function** LocalSearch(Cost())
 2: $X_{best} \leftarrow$ randomly create initial solution;
 3: **while** not terminated **do**
 4: $neighbors \leftarrow$ choose Neighborhood solutions;
 5: $X_{cur} \leftarrow$ choose solution with the best Cost() from $neighbors$;
 6: **if** (Cost(X_{cur}) is acceptable comparing to Cost(X_{best}))
 7: $X_{best} \leftarrow X_{cur}$;
 8: **else**
 9: **return** X_{best} ;
 10: **end if**
 11: **end while**
 12: **end function**

Схема 1. Local Search Algorithm

Scheme 1. Local Search Algorithm

Для ініціалізації первинного рішення використовується евристичний алгоритм First Fit (пошук першого підходящого), що полягає в поступовому розподілі всіх наявних замовлень за маршрутними листами так, щоб мінімізувати обраний для основного алгоритму пошуку рішення критерій оптимізації

Hill Climbing Algorithm (Simple Local Search Algorithm) є локальним оптимізаційним алгоритмом максимізації оцінної функції. Даний алгоритм бере всі кроки з поточної околиці, вибирає з них оптимальний варіант.

Псевдокод алгоритму наведено в схемі 2.

Input: Cost() – function, determining cost of the solution, which algorithm is trying to maximize
Output: solution X_{best}
 1: **function** HillClimbing(Cost)
 2: $X_{best} \leftarrow$ randomly create initial solution;
 3: **while** true **do**
 4: $neighbors \leftarrow$ GetAllNeighborSolutions(X_{best});
 5: $X_{cur} \leftarrow$ select solution with highest cost from $neighbors$;
 6: **if** (Cost(X_{cur}) $\geq$ Cost(X_{best}))
 7: $X_{best} \leftarrow X_{cur}$;
 8: **else**
 9: **return** X_{best} ;
 10: **end if**
 11: **end while**
 12: **end function**

Схема 2. Hill Climbing Algorithm

Scheme 2. Hill Climbing Algorithm

Tabu Search (метод табу-пошуку) так само, як і Hill Climbing Algorithm, шукає оптимальне значення оціночної функції на всіх рішеннях з околиці поточного, але виключає з них ті, руху до яких

зачіпають об'єкти, що містяться в табу-списку. Табу-список являє собою набір об'єктів, що не мають право брати участь в русі. У список об'єкти потрапляють після вибору відповідного кроку, що зачіпає їх, на поточному циклі виконання алгоритму.

Псевдокод алгоритму наведено в схемі 3.

Input: Cost() – function, determining cost of the solution, which algorithm is trying to maximize;
 tabuSize – number of elements in tabu list.
Output: solution X_{best}
 1: **function** HillClimbing(Cost, tabuSize)
 2: $X_{best} \leftarrow$ randomly create initial solution;
 3: $tabuList \leftarrow \{\}$;
 4: **while** true **do**
 5: $neighbors \leftarrow$ GetAllNeighborSolutions(X_{best});
 6: $notTabuedNeighbors \leftarrow neighbors \setminus tabuList$;
 7: $X_{cur} \leftarrow$ select solution with highest cost from $notTabuedNeighbors$;
 8: **if** (Cost(X_{cur}) $\geq$ Cost(X_{best}))
 9: $X_{best} \leftarrow X_{cur}$;
 10: $tabuList \leftarrow tabuList \setminus tabuList[size(tabuList) - tabuSize]$;
 11: $tabuList \leftarrow tabuList \cup X_{cur}$;
 12: **else**
 13: **return** X_{best} ;
 14: **end if**
 15: **end while**
 16: **end function**

Схема 3. Tabu Search.

Scheme 3. Tabu Search.

На відміну від двох попередніх алгоритмів, Simulated Annealing розглядає не всі рухи в околиці, що значно підвищує швидкість виконання цього алгоритму. Перегляд рухів переривається при знаходженні першого підходящого руху, яке і стає оптимальним кроком на цьому циклі. Рух вважається придатним, якщо він призводить до вирішення, що не зменшує значення оціночної функції щодо поточного рішення, або ж зменшує, але проходить випадкову перевірку.

Late Acceptance Hill Climbing Algorithm так само, як і Simulated Annealing, розглядає тільки частину рухів в околиці. Рух може бути вибрано як крок у випадку, якщо воно призводить до вирішення, що не зменшує значення оціночної функції для поточного оптимального рішення, або ж до вирішення з не меншим значенням рахунки, ніж у оптимального рішення певного числа циклів назад.

Step Counting Hill Climbing Algorithm відрізняється від Late Acceptance Hill Climbing Algorithm тим, що замість порівнювання можливих рухів з рахунком оптимального рішення певне число циклів назад даний алгоритм зберігає одну порогову оцінку протягом декількох циклів, вибираючи наступні руху тільки в тому випадку, якщо вони не знижують даний поріг.

Описаний далі генетичний алгоритм оптимізації відрізняється від попередніх алгоритмів тим, що є глобальним, а не локальним алгоритмом оптимізації.

Порівняння результатів застосування реалізованих алгоритмів для вирішення VRP-задач, може бути описано наступним чином:

1) Hill Climbing Algorithm

Алгоритм показав найбільш швидке отримання результату серед інших алгоритмів, але результати вийшли гіршими за якістю через нездатність алгоритму обходити локальні максимуми.

2) Tabu Search - Simulated Annealing - Late Acceptance Hill Climbing Algorithm - Step Counting Hill Climbing Algorithm

Результати тестування даних алгоритмів об'єднані через схожість їх роботи, так як все є локальними алгоритмами оптимізації, зосередженими на вирішенні проблеми уникнення локальних максимумів і пошуку глобального максимуму значення оціночної функції. Через додаткових умов вибору чергового рішення в якості оптимального на кожному циклі алгоритми працювали довше, ніж простий алгоритм локального пошуку, а й отримані результати виявилися набагато ефективніше за значеннями оціночної функції. Кращі результати за значенням оціночної функції були показані алгоритмом Simulated Annealing (але цим алгоритмом потрібно більше ітерацій для знаходження рішення, що виражалося в більшій тривалості роботи), результати ж інших трьох алгоритмів не сильно відрізнялися один від одного по ефективності і швидкості при повторних запусках на одних і тих же даних.

3) Genetic Algorithm.

Результати застосування генетичного алгоритму оптимізації виявилися гіршими результатів всіх інших алгоритмів, за винятком Hill Climbing Algorithm. Відбулося це на більший часу пошуку рішення, що пояснюється необхідністю оцінювати і перевіряти на кожному циклі зміни популяції велике число рішень.

Для написання програми були використані мови PHP, JavaScript і мову розмітки HTML. Вибір обґрунтований специфікою даних мов, спрямованої на створення web-додатків, що було основною вимогою по створенню інтерфейсу додатку. Також важливу роль зіграли реалізації API зовнішніх систем, з якими додатком потрібно працювати, а саме GoogleMaps і геокодер Google, мають реалізацію якраз для вищеприказаних мов.

Для написання програми використовувалися php-бібліотеки "curl" і "SimpleXML", що забезпечують роботу з HTTP-запитами і xml-форматом даних відповідно. HTTP-запити використовувалися для роботи з методами відомого API, а xml-формат використовувався для роботи з Excel-документами.

Модель спроектованої бази даних

Середовищем розгортання даної моделі став MS SQL Server 2014 року, який використовується компанією-замовником в силу своєї простоти в

обслуговуванні і підтримці, а також легкості розгортання і масштабованості.

Як приклад, розглянемо один з «проблемних» днів з 35 контрагентами (включаючи фірму-постачальника). Аналізований день був проблемним для компанії, оскільки компанія зазнавала збитків з вини одного з водіїв в розмірі 5430 гривень.

Контрагенти, у даному випадку це фірмові магазини, торгові мережі та ресторани.

Водій, не встигнувши до покупця, був оштрафований поверненням продукції в повному розмірі. Компанія намагалася зробити дане замовлення пріоритетним, що призвело до запізненнь до наступним покупцям в центрі міста. Таким чином, фірма має такі проблеми, які будуть вирішені в ході проведеного дослідження:

- Формування послідовності обслуговування покупців;
- Скорочення втрат продукції, як в досліджуваній заявці, так і в ряді інших;
- Оптимальний розподіл завантаженості водіїв.

Рис. 2. Початковий маршрут.

Fig. 2. Hiking route.

Рис. 2. Оптимізований маршрут.

Fig. 2. Optimization Route.

Таблиця 1. Маршрут доставки товару контрагентам по місту.**Table 1.** Route of delivery of goods to counterparties by city.

№	Контрагент	Початок	Кінець	Прибуття	Відбуття	Розвантаження	Простій	Запізнення
1	ТОВ «Тернопільський м'ясокомбінат»	-	-	11:00	11:00	-	-	-
2	Піцерія «Фламінго»	11:00	11:30	11:12	11:19	00:07	-	-
3	Фірмовий магазин №1	11:00	12:00	11:25	11:28	00:03	-	-
4	Студія м'яса №1	11:00	12:00	11:37	11:45	00:08	-	-
5	Ресторан «Хуторок»	12:30	13:00	12:20	12:33	00:03	-	00:17
6	Фірмовий магазин №2	11:00	19:00	13:03	13:09	00:06	-	-
7	Фірмовий магазин №3	12:30	13:30	13:11	13:14	00:03	-	-
8	Фірмовий магазин №4	12:00	13:30	13:19	13:28	00:09	-	-
9	Студія м'яса №1	13:00	14:00	13:34	13:39	00:05	-	-
10	ТЦ «Оріон»	11:00	14:00	13:48	13:51	00:03	-	-
11	NOVUS	14:00	15:00	13:55	14:05	00:05	-	00:08
12	Ресторан "Західний"	13:30	15:30	14:08	14:15	00:07	-	-
13	Студія м'яса №2	13:30	17:30	14:20	14:28	00:08	-	-
14	Студія м'яса №3	14:00	20:00	14:35	14:43	00:08	-	-
15	Студія м'яса №4	13:30	18:30	14:47	14:50	00:03	-	-
16	ТЦ «Подoliaни»	15:00	16:00	14:57	15:07	00:07	00:03	-
17	ТОВ «Еверест»	14:00	17:00	15:11	15:18	00:07	-	-
18	ТЦ Північний	13:00	18:00	15:30	15:35	00:05	-	-
19	ТОВ «Клуб м'яса»	15:00	19:00	16:50	16:54	00:04	-	-
20	Студія м'яса №5	12:00	18:00	17:06	17:11	00:05	-	-
21	Кафе «Склянка»	15:00	19:00	17:15	17:24	00:07	-	-

Висновки

1. Генетичний алгоритм розвиває покоління (набір) індивідумів, представлених у вигляді хромосом, створюючи нові покоління нащадків за допомогою внесення змін в дані, до тих пір, поки не будуть досягнуті певні критерії параметрів завершення алгоритму. До таких критеріїв в описуванні реалізації алгоритму відноситься значення максимального числа поколінь. По завершенню виконання алгоритму, найкраща вироблена хромосома в поколінні розшифровується і вибирається в якості шуканого рішення.

2. Для вирішення завдання VRP за допомогою генетичного алгоритму оптимізації була обрана модель представлення кожного індивідуума в поколінні (що означає одне рішення) тільки однієї хромосомою, де кожна хромосома виглядає як ланцюжок замовлень і транспортних засобів. Транспортні засоби є роздільниками між маршрутними листами, а замовлення, розташовані після таких роздільників, показують свою приналежність до певного маршруту.

3. Створення нового покоління включає в себе три стадії: селекція частини індивідуумів з поточного покоління з найкращими показниками оціночної

функції; випадковий вибір набору пар індивідуумів з поточного покоління і рекомбінація генів в хромосомах (операція кросинговеру) обраних батьків всередині пари для отримання двох нащадків; мутація генів нащадків для збільшення диверсифікації індивідуумів (рішень).

4. Описані стадії створюють зі старого покоління нове, з одного боку зберігаючи якість у вигляді набору кращих хромосом, а з іншого - додаючи операцією кросинговеру і мутацією диверсифікацію в набір рішень нового покоління.

5. Перше покоління виходить аналогічно стартовому рішенням для алгоритмів локального пошуку, описаних раніше, за допомогою евристики First Fit, яку запускає певну кількість разів для списків замовлень в різних послідовностях. Псевдокод алгоритму наведено в додатку 3.

6. В результаті проведеного дослідження компанія вирішила такі питання:

а) формування послідовності обслуговування покупців;

б) скорочення втрат продукції, як в досліджуваній заявці, так і в ряді інших;

в) оптимальний розподіл завантаженості водіїв.

Список літератури

1. *Arbelaitz O., Rodriguez C.* Low cost parallel solutions for the VRPTW optimization problem. *International Journal of Computational Science and Engineering*. 2005. T. 1. №. 2-4. С. 175-182.
2. *Blasum U., Hochstättler W.* Application of the branch and cut method to the vehicle routing problem. *Zentrum für Angewandte Informatik Köln Technical Report zpr2000-386*. 2000.
3. *Bräysy O., Hasle G., Dullaert W.* A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*. 2004. T. 159. №. 3. С. 586-605.
4. *Burke E. K., Bykov Y.* The late acceptance hill-climbing heuristic. *University of Stirling, Tech. Rep.* 2012.
5. Local search algorithms [Електронний ресурс] – Режим доступа: https://www.cs.unc.edu/~lazechnik/fall10/lec06_local_search.pdf.
6. SimpleXML [./en/book.simplexml.php].
7. Vehicle routing & scheduling [Електронний ресурс]. Режим доступа: <http://www.opendoorlogistics.com/software/vehicle-routing-scheduling>.
8. *Jaszkiewicz A., Kominek P.* Genetic local search with distance preserving recombination operator for a vehicle routing problem. *European Journal of Operational Research*. 2003. T. 151. №. 2. С. 352-364.
9. *Аникин Б. А.* Логистика: учебник. Под ред. Б. А. Аникина. Москва. Инфра-М, 2006.
10. *Иванов, Д. А.* Управление цепями поставок. Издательство Политехнического университета, 2009. 659 с.
11. *Пелих С. А.* Организация производства в условиях переходной экономики. Право и экономика, 2009. 576 с.
12. *Стерлигова А. Н.* Управление запасами в цепях поставок: учебник. ИНФРА-М, 2009. 428 с.
13. *Гаджинский А. М.* Основы логистики: учебное пособие. Москва. ИВЦ «Маркетинг», 2002.
14. *Бажин И. И.* Логистика: компакт-учебник. Харьков. Консум, 2003.
15. *Новиков О. А., Уваров С. А.* Логистика: учебное пособие. Санкт-Петербург. Бизнес-пресса, 1999. 208 с.
16. *Неруш Ю. М.* Логистика: учебник. Москва. 2008. 517 с.
4. *Burke E. K., Bykov Y.* (2012). The late acceptance hill-climbing heuristic. *University of Stirling, Tech. Rep.*
5. Local search algorithms [Electronic resource] - Access mode: https://www.cs.unc.edu/~lazechnik/fall10/lec06_local_search.pdf
6. SimpleXML [./en/book.simplexml.php].
7. Vehicle routing & scheduling [Electronic resource]. Access mode: <http://www.opendoorlogistics.com/software/vehicle-routing-scheduling>.
8. *Jaszkiewicz A., Kominek P.* (2003). Genetic local search with distance preserving recombination operator for a vehicle routing problem. *European Journal of Operational Research*. T. 151. No. 2. 352-364.
9. *Anikin B. A.* (2006). *Logistika: Textbook* / Ed. B.A. Anikina. Moscow. Infra-M.
10. *Ivanov D. A.* (2009). *Supply chain management*. The Publishing House of the Polytechnic University. 659.
11. *Pelikh S. A.* (2009). *Organization of production under the conditions of a transitional economy*. Law and Economics. 576.
12. *Sterligova A. N.* (2009). *Management of supplies in supply chains: a textbook*. INFRA-M. 428.
13. *Gadzhinsky A. M.* (2002). *Fundamentals of Logistics: A Study Guide*. Moscow. IEC "Marketing".
14. *Bazhin I. I.* (2003). *Logistics: Compact tutorial*. Kharkiv. Consum.
15. *Novikov O. A., Uvarov S. A.* (1999). *Logistics: Study. help*. St. Petersburg. Business Press. 208.
16. *Nerush Y. M.* (2008). *Logistika: textbook*. 517.

ДИГИТАЛИЗАЦІЯ В ЛОГІСТИКЕ ЗА РАХУНОК ВВЕДЕНИЕ АЛГОРИТМА МУРАВЬИНОЙ ЛОГИСТИКИ

Л. А. Савченко

Аннотация. Дигитализация это новое понятие, которое помогает транспортной отрасли при планировании маршрутной сети повысить их эффективность и минимизировать негативное влияние на окружающую среду. Кроме того, цифровые платформы станут чрезвычайно важными в логистической отрасли, ведь позволят малым компаниям получить глобальный доступ и конкурировать с крупными игроками. В рамках индустрии логистики и цепях поставок наблюдается множество потерь из-за ведения непродуктивной деятельности. Важность низких затрат на транспортировку и их последующее влияние на стоимость доставки товаров многократно увеличивают производительность и сбережения компании при интеграции решения VRP, тем самым искореняя все сомнения в актуальности исследования VRP.

Ключевые слова: логистика, дигитализации, транспортный процесс, информационные технологии, муравьиная логистика.

DIGITALIZATIONS IN LOGISTICS BY INTRODUCTION OF MURALOG LOGISTICS ALGORITHM

L. A. Savchenko

Abstract. Digitization is a new concept that has helped the transport industry in planning its route network

References

1. *Arbelaitz O., Rodriguez C.* (2005). Low cost parallel solutions for the VRPTW optimization problem. *International Journal of Computational Science and Engineering*. T. 1. No. 2-4. 175-182.
2. *Blasum U., Hochstättler W.* (2000). Application of the branch and cut method to the vehicle routing problem. *Zentrum für Angewandte Informatik Köln Technical Report zpr2000-386*.
3. *Bräysy O., Hasle G., Dullaert W.* (2004). A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*. T. 159. No. 3. 586-605.

to increase their efficiency and minimize the environmental impact. In addition, digital platforms will become extremely important in the logistics industry, allowing small businesses to gain global access and compete with big players. Within the logistics industry and the supply chain, there are many losses due to unproductive activities. The importance of low transportation costs and their subsequent impact on the cost of delivering goods repeatedly increases the productivity and cost savings of a company when integrating a VRP solution, thus eliminating any doubts about the relevance of the VRP study.

Key words: logistics, digitalization, transport process, information technologies, ant logistics.

Л. А. Савченко ORCID 0000-0002-4336-4416.