УДК 004.9, 519.8

ПОБУДОВА ПОЧАТКОВОГО РІШЕННЯ ДЛЯ ЗАДАЧІ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ З ПІДБОРОМ ТА ДОСТАВКОЮ

Молчановський О.І.
Національний технічний університет України «Київський Політехнічний Інститут»,
Любонько А.Л.
ПП «Українські Інтелектуальні Технології», м. Київ

В *роботі розглянута задача маршрутизації транспортних засобів з підбором та доставкою (Vehicle Routing Problem with Pickup and Delivery – VRPPD). Ця задача є варіантом окремої задачі маршрутизації транспортних засобів (VRP). Наведено формулювання задачі та її математична модель. Запропоновано двоступеневий алгоритм побудови початкового рішення для даної задачі. Результати експериментів на відомих еталонних прикладах порівняно з найкращими відомими на даний момент. Отримані результати є кращими у порівнянні із відомими результатами побудови початкового рішення для VRPPD.*

Ключові слова: маршрутизація транспортних засобів, математична модель, оптимізація, початкове рішення, кластеризація.

Вступ. У задачах маршрутизації транспортних засобів (Vehicle Routing Problem – VRP) розглядаються проблеми оптимальної доставки товарів між складами та кінцевим клієнтами. Задачі VRP вперше були сформульовані у 1959 році [1], але найбільшого розвитку вони отримали останні 20 років. У світі існує декілька наукових шкіл, які займаються цілеспрямовано цим типом задач. Проте, в Україні можна говорити про відсутність особливої цікавості до даної тематики. Задачі VRP мають великий потенціал до практичного впровадження, в першу чергу у галузі логістики. Прикладами застосувань таких задач у реальному світі є автоматизація роботи підприємств з великою кількістю транспортних засобів, збір сміття на вулицях, маршрутизація шкільних автобусів, транспортування людей з фізичними вадами, доставка товарів клієнтам магазинів [2].

Існує достатньо велика кількість задач VRP [3], які відрізняються одна від одної врахуванням тих або інших характеристик вузлів, транспортних засобів, а також введенням додаткових обмежень. Наявність різних обмежень визначається типом задач, а часто також тим, що реальні задачі є дуже складними і в таких випадках у першу чергу розглядаються спрощені задачі. Найпростішим варіантом VRP є задача з одним складом та однотипними транспортними засобами (Capacitated VRP). До обмежень задач VRP відносяться: наявність часових вікон вузлів (VRP with Time Windows); використання декількох типів транспортних засобів (VRP with Heterogeneous Fleet); врахування різних типів вузлів, наприклад, коли в одному вузлі необхідно забрати певні товари і доставити їх згодом в інший вузол (VRP with Pickup and Delivery); попереднє обмеження спрощується, коли у деяких вузлах необхідно забрати товари та доставити їх на склад (VRP with Backhauls); кожний вузол-клієнт може обслуговуватись тільки транспортними засобами певного типу (Site Dependent VRP).

У даній роботі розглядається задача маршрутизації транспортних засобів з підбором та доставкою (VRPPD). Дана задача є однією з найскладніших екземплярів задач VRP [3]. Одними з перших, хто розглянув цю задачу були Savelsbergh, Sol [4]. Першими задачу VRPPD з часовими вікнами розв'язали Li, Lim [5]. Вони також ввели еталонні приклади, які заснували на загальноприйнятих вже на той час прикладах Solomon [6].

Загальноприйнятим є підхід до вирішення задачі, при якому розв'язок будується в два етапи: спочатку генерується початкове рішення, а потім це рішення покращується алгоритмами на основі мета-евристик. Більшість літератури з даної тематики не звертає особливої уваги на побудову початкового рішення, роблячи акцент на етапі покращення, за виключенням роботи [10]. Але від якості отриманого на першому етапі рішення залежить й загальний результат, адже відомо, що метаевристичні алгоритми є чутливими до якості вхідних сформованих рішень. Серед розповсюджених мета-евристичних методів розв'язку задачі VRPPD поширені метод симуляції відпалу [7], методи пошуку у великому просторі сусідів (Large Neighborhood Search) [8], генетичні алгоритми [9]. З точки зору побудови початкового рішення можна згадати статтю [10].

Точні методи розв'язку (такі як, наприклад, метод гілок та меж) з практичної точки зору можна застосовувати до задач із кількістю вузлів не більше 100. Серед робіт в даному напрямку можна виокремити [11]. В роботі [12] проведена спроба поєднати точні методи розв'язку з евристичними.

Опис математичної моделі. Опис математичної моделі базується на роботах [13] та [10]. Постановка задачі VRPPD включає в себе такі елементи: вузли замовлень, термінальні вузли та транспортні засоби (ТЗ). Загальна кількість замовлень n. Кожне замовлення r_i включає в себе вузол підбору p_i та вузол доставки d_{i+n}. Таким чином, маємо множину вузлів підборів $P=\{1,...,n\}$ та доставок $D=\{n+1,...,2n\}$. Нехай $N=P\cup D$. K – множина всіх ТЗ, $|K|=m$. Нехай T, $|T|=2m$ – множина термінальних вузлів. $\tau_k=2n+k$ та $\tau'_k=2n+k+m$, де $k\in K$ – відповідно початковий термінальний та кінцевий термінальний вузол для ТЗ k. Зазначимо, що як вузли замовлень, так і термінальні вузли можуть співпадати між собою. Отже може бути один термінальний вузол для всіх ТЗ (як, наприклад, в еталонних тестах Li-Lim [5]) та/або деякі вузли одночасно можуть бути й вузлами підбору, й вузлами доставки. Граф $G=(V,A)$ складається з вузлів $V=N\cup\{\tau_1,...,\tau_m\}\cup\{\tau'_1,...,\tau'_m\}$ та дуг $A=V\times V$. Для кожної дуги $(i,j)\in A$ визначається відстань $d_{ij}\geq 0$ та час подорожі $t_{ij}\geq 0$. Передбачається, що значення відстаней та подорожей задовольняють нерівностям трикутника: $d_{ij}\leq d_{il}+d_{lj}$ та $t_{ij}\leq t_{il}+t_{lj}$ для всіх $i,j,l\in V$.

Кожний вузол $i\in V$ має час обслуговування s_i та часове вікно $[a_i,b_i]$, $a_i\leq b_i$. Час обслуговування – це час необхідний для завантаження чи розвантаження ТЗ у відповідному вузлі. Часове вікно вказує часові межі для вузла, коли в ньому може відбуватись обслуговування. ТЗ може прибути у вузол раніше часу a_i, але тоді йому доведеться чекати відкриття часового вікна для початку обслуговування. Для кожного вузла $i\in P$ визначається кількість вантажу $q_i>0$, який необхідно підібрати у цьому вузлі. Так само для кожного вузла $j\in D$ визначається кількість вантажу $q_j>0$, який необхідно доставити до цього вузла. При чому, якщо вузли i та j належать до одного замовлення $(j=i+n)$, то $q_i=q_j$. Вважається, що кожний вузол доставки чи підбору відвідується лише один раз. Умова передування вимагає, щоб вузли підбору передували відповідним вузлам доставки у сформованих маршрутах.

Місткість ТЗ $k\in K$ дорівнює C_k, C – множина різних значень C_k. Для будь-якого сформованого маршруту вимагається, щоб його ТЗ на будь-якій ділянці маршруту не був завантажений більше наведеної місткості C_k. З кожним ТЗ також пов'язані вартості: α_k – вартість за одиницю шляху, β_k – вартість за одинцю часу, γ_k – одноразова вартість використання транспортного засобу.

Цільова функція може варіюватись. Загалом можуть мінімізуватись наступні характеристики: загальна вартість, загальна відстань та/або час, загальна кількість маршрутів, кількість замовлень, які не вдалось обслужити. Таким чином задача зводиться до мінімізації цільової функції:

$$\min \sum_{k\in K}\alpha_k \sum_{(i,j)\in A} d_{ij}x_{ijk} + \sum_{k\in K}\beta_k\left(S_{\tau'_k,k} - a_{\tau_k}\right) + \sum_{k\in K}\gamma_k y_k + \delta\sum_{i\in P} z_i\,, \tag{1}$$

де x_{ijk} – булева змінна, яка дорівнює 1, якщо дуга $(i,j)\in A$ обслуговується ТЗ $k\in K$; $S_{\tau'_k,k}$ – час прибуття ТЗ k у термінал τ'_k; y_k – булева змінна, яка дорівнює 1, якщо ТЗ $k\in K$ використовується в остаточному рішенні; z_i – булева змінна, яка дорівнює 1, якщо замовлення $i\in P$ не було розподілено; δ – штраф за один нерозподілений вузол.

Метод побудови початкового рішення. Запропонований метод складається з двох алгоритмів: перший знаходить рішення задачі VRPPD для обмеженої кількості ТЗ, другий – для необмеженої кількості ТЗ. При чому перший алгоритм викликає на виконання другий. На вхід першого алгоритму подається множина вузлів замовлень N, термінальних вузлів T та множина транспортних засобів K; на виході алгоритму – множина сформованих маршрутів M, кожен з яких є послідовністю вузлів, в якій перший та останній – термінальні вузли відповідного ТЗ, решта – вузли підбори та доставки із виконанням умови передування.

Алгоритм побудови рішення задачі VRPPD для обмеженої кількості ТЗ.

1. Поточна множина замовлень $N' := N$. Множина доступних ТЗ $K' := K$. Множина маршрутів $M := \{\ \}$.
2. Знайти рішення для N' із необмеженою кількістю ТЗ, використовуючи наявні типи ТЗ у K'. Множина M' складається з нових знайдених маршрутів.
3. Впорядкувати множину нових знайдених маршрутів M' за важливістю.
4. Спробувати знайти для всіх маршрутів M', починаючи з найбільш важливих, вільні ТЗ серед K'.
5. Видалити відібрані у п. 4 ТЗ з множини K' та розподілені замовлення з множини N'. Додати до множини M множину M': $M := M \cup M'$.
6. Якщо множина $N' \neq \{\ \}$, то перейти до п. 2. Інакше кінець роботи.

Термін «важливість маршруту» містить наступний зміст: маршрут m_i вважається більш важливим за маршрут m_j, якщо сумарна кількість вантажу, який буде доставлений всередині маршруту m_j, є більшою за сумарну кількість вантажу у m_i.

Другий алгоритм приймає на вхід множину вузлів замовлень N та термінальних вузлів T, а також множину типів машин C; на виході – множина сформованих маршрутів M. Даний алгоритм за своєю суттю відноситься до жадібних алгоритмів.

Алгоритм побудови рішення задачі VRPPD для необмеженої кількості ТЗ.

1. Встановити $M := \{\ \}$.
2. Для кожної пари вузлів (p_i, d_{i+n}) замовлення i знайти найкращий тип ТЗ $k\in C$ та створити новий маршрут $m = (\tau_k, p_i, d_{i+n}, \tau'_k)$. $M := M \cup \{m\}$.
3. Серед усіх маршрутів M знайти пару m_i та m_j, об'єднання яких дає найбільший виграш по вартості. Якщо таких маршрутів не знайдено, то кінець роботи. Інакше перейти до п. 4.
4. Об'єднати маршрути m_i та m_j у новий маршрут $m_k = m_i \cup m_j$. $M := M \cup \{m_k\} \setminus \{m_i, m_j\}$. Перейти до п. 3.

У п. 2 алгоритму шукається найбільш вигідний тип ТЗ для майбутнього маршруту. Цей вибір робиться на основі вартості маршруту:

$$\min_{k\in C} F(\tau_k, p_i, d_{i+n}, \tau'_k)\,, \tag{2}$$

де $F(\cdot)$ – функція вартості маршруту.

У п. 3 шукається найкраще об’єднання двох маршрутів, яке робиться на основі різниці вартості двох окремих маршрутів та нового об’єднаного маршруту:

$$\max_{i,j\in M}\left(F(m_i)+F(m_j)-F(m_i\cup m_j)\right), \tag{3}$$

де $m_i \cup m_j$ – маршрут, який отримано в результаті об’єднання маршрутів m_i та m_j. Об’єднання враховує пошук найкращого типу ТЗ $k \in C$, щоб вартість $F\left(m_i \cup m_j\right)$ була мінімальною. Під час об’єднання двох маршрутів послідовно розглядаються всі можливі результати об’єднання. Нехай маємо два маршрути для об’єднання $m_i = (v_0^i, v_1^i, ..., v'^i_0)$ та $m_j = (v_0^j, v_1^j, ..., v'^j_0)$, які мають відповідно n_i та n_j нетермінальних вузлів. Тоді загальна кількість можливих варіантів об’єднання цих двох маршрутів дорівнює:

$$C_{n_i+n_j}^{n_i} = \frac{\left(n_i + n_j\right)!}{n_i! \cdot n_j!}. \tag{4}$$

Зрозуміло, що при достатньо великих маршрутах час на перевірку всіх варіантів є неприйнятно великим. Тому ми використовуємо кластеризацію для того щоб скоротити загальну кількість варіантів.

Кластеризація використовується для розбиття вузлів підбору та доставки на множини-кластери. І хоча на практиці нам відомі фактичні координати вузлів на карті, але ми використовуємо методи розбиття вузлів на кластери, які ґрунтуються виключно на матрицях відстані чи часу. Це обумовлено тим, що в загальному випадку ми розв’язуємо задачу у неевклідовому просторі. Використаний нами метод [14], приймаючі на вхід матриці відстані або часу, повертає ймовірність приналежності кожного вузла до кожного кластеру у вигляді матриці $\mathrm{P} = \left\|\rho_{ik}\right\|$, де ρ_{ik} – величина приналежності вузла i до кластеру k. Кількість кластерів n_C задається як вхідний параметр цього методу.

При створенні маршруту $m_i = (v_0^i, v_1^i, ..., v'^i_0)$ визначаються ймовірності його приналежності P_{ik} до кожного кластеру k, як середнє арифметичне приналежності його вершин до даного кластеру:

$$P_{ik} = \frac{1}{|m_i|}\sum_{v\in m}\rho_{vk}, \tag{5}$$

де $|m_i|$ – кількість нетермінальних вершин у маршруті m_i. При чому кожному маршруту ставиться у відповідність множина K_i, що включає кластери, до яких маршрут m_i належить з ймовірністю більше деякого порогу κ, тобто:

$$K_i = \left\{ j \mid P_{ij} \geq \kappa \right\}. \tag{6}$$

Під час об’єднання двох маршрутів m_i та m_j вираховується відношення спільних кластерів для обох маршрутів до загальної кількості кластерів, яким приналежать вершини маршрутів (з урахуванням значення порогу κ). Об’єднання відбувається за умови, що дане відношення більше значення порогу λ:

$$\frac{\left|K_i \cap K_j\right|}{\left|K_i \cup K_j\right|} \geq \lambda. \tag{7}$$

При проведенні експериментів значення κ та λ бралися 0,2 та 0,5 відповідно.

На етапі вибору кращого варіанту об’єднання маршрутів можуть враховуватись додаткові характеристики маршрутів, які не включаються у функцію вартості $F(\cdot)$. Нами було розглянуто вплив трьох таких характеристик: кількість перетинів всередині маршруту, кількість вузлів в маршруті, запас часу на маршрут.

Перетином в даній постановці вважається візуальний перетин ділянок маршруту на площині в точках, відмінних від вузлів маршруту. Даний підхід до задачі VRPPD був запропонований у [15]. В даній роботі вводиться поняття «візуальної привабливості маршруту». Вважається, що візуально більш привабливі маршрути, тобто такі, які мають меншу кількість перетинів, можуть бути оптимальнішими за своїми головними ознаками: довжина, час, вартість. З іншого боку, саме такі маршрути в першу чергу створюють в ручному режимі фахівці-логісти. Тому слід намагатись зменшити кількість перетинів у створюваних маршрутах. Цього можна досягти, додавши до функції вартості штраф за кожний перетин.

Під час синтезу маршрутів бажано зменшити загальну кількість ТЗ, які будуть використані. Хоча, загалом, це є другорядним завданням – мінімізація ТЗ, проте може бути бажаним надавати перевагу маршрутам із більшою кількістю вузлів. Таким чином, очікуючи, що це призведе до зменшення загальної кількості ТЗ у остаточному розв’язку. Цей чинник можна увести до функції вартості шляхом зменшення загальної вартості на величину залежну від кількості вузлів у маршруті. Нами було використана квадратична залежність.

Останнім евристичним міркуванням щодо покращення якостей маршрутів є запас часу по відношенню до всіх вузлів маршруту. Ми тут керуємось таким міркуванням, що чим щільніше в часі розташовані вузли в маршруті, тим краще, адже це в перспективі може дозволити додати більше нових вузлів до даного маршруту. Що в свою чергу призведе до зменшення загальної вартості розв’язку.

З урахуванням наведених вище трьох характеристик функція вартості маршруту m_i набуде наступного вигляду:

$$F'(m_i) = F(m_i) + \omega_1 \cdot crosses_i - \omega_2 \cdot |m_i|^2 + \omega_3 \cdot \frac{b_{\tau'} - a_\tau}{|m_i|}, \tag{8}$$

де $crosses_i$ – кількість перетинів в маршруті; $|m_i|$ – кількість 260е термінальних вершин у маршруті; a_τ, $b_{\tau'}$ – час відкриття часового вікна у стартовому терміналі та час закриття вікна у кінцевому терміналі відповідно; ω_1, ω_2, ω_3 – керуючі параметри. При чому штраф ω_1 за перетини може бути абсолютним (в одиницях вартості) або відносним (у відсотках від вартості). Тож тепер у формулі (3) $F(\cdot)$ замінюється на модифіковану функцію вартості $F'(\cdot)$.

Результати експериментів. Запропонований метод побудови початкового рішення задачі VRPPD було перевірено на еталонних прикладах з роботи [5]. Дані прикладі є модифікацією прикладів з роботи [6] для задачі VRPTW. Множина прикладів містить 56 задачі по 100 вузлів. Задачі розбиваються на 3 класи: LR (вузли розташовані випадково), LC (вузли розбиті на кластери), LRC (є вузли як в кластерах, так і поза кластерами). При чому в кожному класі є по два підкласи: з ТЗ меншої та більшої місткості.

При проведенні експериментів значення керуючих параметрів обирались в наступних діапазонах:

- кількість кластерів n_C – від 5 до 12;
- відносні значення ω_1 : $\{0; 0{,}1; 0{,}5; 1\}$ та абсолютні значення: $\{5; 30; 80; 300\}$;
- ω_2: $\{0; 2; 5; 10\}$;
- ω_3: $\{0; 0{,}5; 1; 2; 5\}$.

Таблиця 1 – Найкращі отримані результати експериментів

Клас	D_{best}	T_{best}	N_{best}
LR101	1679,21	3561,23	19
LR102	1608,17	3272,3	17
LR103	1568,22	2948,94	16
LR104	1290,83	2391,83	13
LR105	1521,7	3008,44	17
LR106	1356,33	2585,19	14
LR107	1251,41	2405,66	12
LR108	1144,97	2211,42	11
LR109	1359,4	2570,39	14
LR110	1292,76	2439,56	13
LR111	1247,07	2420,87	12
LR112	1193,27	2276,58	12
LR201	1469,5	3485,58	5
LR202	1658,48	3574,06	5
LR203	1446,66	3046,83	4
LR204	1435,97	2592,69	3
LR205	1552,27	3051,43	5
LR206	1413,72	2840,3	4
LR207	1569,05	2706,54	4
LR208	1364,93	2370,31	3
LR209	1331,71	3160,2	5
LR210	1635,88	3061,77	4
LR211	1378,48	2581,53	4
LC101	828,94	9828,94	10
LC102	828,94	9828,94	10
LC103	939,87	10013,68	10
LC104	1008,8	10072,83	11
LC105	828,94	9828,94	10
LC106	828,94	9828,94	10
LC107	828,94	9828,94	10
LC108	829,28	9829,28	10
LC109	828,94	9828,94	10
LC201	871,35	17931,2	6
LC202	968,63	17894,7	7
LC203	923,34	15734,59	6
LC204	1077,61	14912,47	7
LC205	823,11	14904,71	7
LC206	955,99	17339,46	7
LC207	954,34	15966,63	6
LC208	886,47	14741,8	6
LRC101	1846,17	3376,99	17
LRC102	1953,73	3495,57	18
LRC103	1503,2	2701,02	13
LRC104	1395,06	2495,41	13
LRC105	1785,34	3176,33	16
LRC106	1662,21	2827,12	14
LRC107	1445,97	2601,45	14
LRC108	1287,29	2404,98	12
LRC201	2380,48	6984,54	10
LRC202	2234,89	6819,05	10
LRC203	1866,87	5401,61	8

Продовження табл. 1

LRC204	1461,57	4822,27	8
LRC205	2199,96	6157,39	9
LRC206	2240,99	5348,96	9
LRC207	2296,42	5765,71	12
LRC208	1632,69	4767,37	10

В таблиці 1 наводяться результати експериментів по всім 56 еталонним прикладам. Наводяться найкраща отримана сумарна відстань (D_{best}), найкращий сумарний час (T_{best}) та найкраща кількість маршрутів (N_{best}). При чому ці значення можуть не належити одному розв'язку, а бути отриманими при різних керуючих параметрах.

За результатами експериментів виявилось, що оптимальна кількість кластерів n_C для задач з класу LC дорівнює від 8 до 10; для задач з класу LR – від 5 до 7; для задач з класу LRC – від 8 до 12. Окремо слід зазначити на значне пришвидшення пошуку розв'язку при використанні кластерів – від одного до двох порядків.

Також виявилось що врахування кількості перетинів (параметр ω_1) не дає гарних результатів. Тільки в одиничних випадках найкраще рішення знаходилось при невеликих (для абсолютного – 5; 30, для відносного – 0,1) значеннях цього параметру. Значення параметрів ω_2 та ω_3 для найкращих розв'язків набули широких діапазонів в залежності від задачі. Це свідчить про необхідність проведення подальших досліджень щодо впливу даних параметрів на якість отриманих рішень.

В таблиці 2 наводяться середні по класах відхилення отриманих результатів від відомих найкращих результатів (http://www.top.sintef.no/vrp/pdp_bknown100.html) для загальної відстані сформованих маршрутів Δ_{dist} та кількості маршрутів Δ_{routes}. Також для порівняння наводяться аналогічні величини з роботи [10] – стовпці Δ^1_{dist} та Δ^1_{routes}. Загальний час обрахунку одного екземпляру задачі в середньому становив 0,5 – 1 сек.

Таблиця 2 – Порівняння результатів експериментів з найкращими відомими

Клас	Δ_{dist}, **%**	Δ_{routes}, **%**	Δ^1_{dist}, **%**	Δ^1_{routes}, **%**
LR1	13,6	21,5	60	45
LC1	4,1	4,9	125	25
LRC1	16,5	27,2	65	58
LR2	54,9	56,8	175	103
LC2	58,3	116,7	370	92
LRC2	81,5	196,9	190	115

Висновки. Було розглянуто метод побудови початкового рішення для задачі VRPPD. Результати експериментів на еталонних прикладах показали, що метод непогано себе поводить для задач із ТЗ малої місткості і гірше при більшій місткості. Найгірші результати виявлено для задач з підкласу LRC2, в яких маршрути складаються з 30-50 пар вузлів. Отримані результати виявились відчутно кращими (в рази) по відношенню до методу з [10]. Подальша робота буде спрямована у бік створення методів покращення початкового рішення та більш детального аналізу керуючих параметрів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Dantzig, A. G. B., Ramser, J. H. The Truck Dispatching Problem Stable // Management Science. – 1959. – 6 (1). – P. 80-91.

2. P. Toth, D. Vigo, The vehicle routing problem // Society for Industrial Mathematics, 2002. – Vol. 9

3. Eksioglu, B., Vural, A. V., & Reisman, A. The vehicle routing problem: A taxonomic review // Computers & Industrial Engineering. – 2009. – 57 (4). – P. 1472-1483.

4. Savelsbergh, M. W. P., Sol, M. The General Pickup and Delivery Problem // Transportation Science. – 1995. – 29 (1). – P. 17-29.

5. Li, H., Lim, A. A Metaheuristic for Pickup and Delivery Problem with Time Windows. In: Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence. Dallas, TX, USA, 2001. – P. 160-167.

6. Solomon, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints // Operations Research. – 1987. – 5 (2). – Pp. 254-265.

7. Bent, R., Hentenryck, P. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows // Computers & Operations Research. – 2006. – 33 (4). – P. 875-893.

8. Ropke, S., Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows // Transportation Science. – 2006. – 40 (4). – Pp. 455-472.

9. M. I. Hosny and C. L. Mumford. Investigating genetic algorithms for solving the multiple vehicle pickup and delivery problem with time windows. In MIC2009 // Metaheuristic International Conference, July 2009.

10. Hosny, M. I., Mumford, C. L. Constructing initial solutions for the multiple vehicle pickup and delivery problem with time windows // Journal of King Saud University, 2011.

11. Ropke, S., Laporte, G., Cordeau, J.-F. Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows // Networks. – 2007, 49 (4). – P. 258-272.

12. Holborn, P. L., Thompson, J. M., Lewis, R. Combining Heuristic and Exact Methods to Solve the Vehicle Routing Problem with Pickups and Deliveries and Time Windows, In: Proceedings of the main European events on Evolutionary Computation. – Malaga, Spain, 2012. – P. 63-74.

13. Ropke, S., Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows // Transportation Science. – 2006. – 40 (4). – P. 455-472.

14. Hathaway, R. J., Bezdek, J. C., Davenport, J. W. (1996). On relational data versions of c-means algorithms // Pattern Recognition Letters, 5 (96). – Pp. 607-612.

15. Lu, Q., Dessouky, M. M. A New Insertion-based Construction Heuristic for Solving the Pickup and Delivery Problem with Time Windows // European Journal of Operational Research. – 2006. – 175 (2). – P. 672-687.

Молчановский А.И., Любонько А.Л. ПОСТРОЕНИЕ НАЧАЛЬНОГО РЕШЕНИЯ ДЛЯ ЗАДАЧИ МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ С ПОДБОРОМ И ДОСТАВКОЙ

В работе рассмотрена задача маршрутизации транспортных средств с подбором и доставкой. Приведена формулировка задачи и ее математическая модель. Предложен двухэтапный алгоритм построения начального решения для данной задачи. Эксперименты на известных эталонных примерах показали значительно более качественные результаты в сравнении с известными методами.

Ключевые слова: маршрутизация транспортных средств, математическая модель, оптимизация, начальное решение, кластеризация.

Molchanovskyi O.I., Lyubonko A.L. CONSTRUCTING INITIAL SOLUTIONS FOR VEHICLE ROUTING PROBLEM WITH PICKUP AND DELIVERY

In this paper we look at Vehicle Routing Problem with Pickup and Delivery. Problem description and mathematical model were presented. The two-stage initial solution construction algorithm was proposed. The algorithm was tested on the known benchmarks and shown better performance compared with known methods.

Keywords: vehicle routing problem, mathematical model, optimization, initial solution, clusterization.