

Література

1. Eike-Henner, W. K. E-Healthanditschallenges [Text] / W. K. Eike-Henner // Healthcare IT Management. – 2010. – Vol. 2, Issue 5. – P. 32–33.
2. Шульман, І. Медичні інформаційні системи: «аксіомаюзабіліті» [Електронний ресурс] / І. Шульман. – Режим доступу: <http://www.pcweek.ru/themes/detail.php?ID=73462>
3. Халафяна, А. А. Сучасністатистичніметодимедицихдосліджень [Текст] / А. А. Халафяна. – М.: ЛКІ (URSS), 2008. – 316 с.
4. Дік, В. В. Методологіяформуваннярішень в інформаційних системах та інструментальнісередовищаїхпідтримки [Текст] / В. В. Дік. – М.: Фінанси і статистика, 2001. – 297 с.
5. Тезiskonференції 3-го Міжнародного форуму «MedSoft-2006» [Текст] / М., 2006.
6. Сіманков, В. С. Системнийпідхід до розробки медичних систем підтримкиприйняттярішень [Текст] / В. С. Сіманков, А. А. Халафян // Новини вищихнавчаль-

нихзакладів. Північнокавказский регион. Технічні науки. – 2010. – № 1. – С. 29–36.

References

1. Eike-Henner, W. K. (2010). E-Health and its challenges. Healthcare IT Management, 2 (5), 32–33.
2. Shulman, I. Medical Information Systems "axiom of usability". Available at: <http://www.pcweek.ru/themes/detail.php?ID=73462>
3. Khalafyan, A. (2008). Modern statistical methods for medical research. Moscow: LKI (URSS), 316.
4. Dick, V. (2001). The Methodology of forming solutions in information systems and tools to support them. Moscow: Finances and Statistics, 297.
5. Abstracts of the 3rd conference of International Forum «MedSoft- 2006" (2006). Moscow.
6. Symankov, V. C., Khalafyan, A. A. (2010). Systemic approach for development of medical support system solutions. Proceedings of the higher educational institutions. Severokavkazskyy region. Technical science, 1, 29–30.

*Рекомендовано до публікації д-р техн. наук, професор Лисенко О. І.
Дата надходження рукопису 20.05.2015*

Антонова-Рафі Юлія Валеріївна, кандидат технічних наук, доцент, кафедра біобезпеки і здоров'я людини, Національний технічний університет України «Київський політехнічний інститут», пр. Перемоги, 37, м. Київ, Україна, 03056

E-mail: unes04@mail.ru

Московський Владислав Ігорович, кафедра біобезпеки і здоров'я людини, Національний технічний університет України «Київський політехнічний інститут», пр. Перемоги, 37, м. Київ, Україна, 03056

E-mail: im-91@mail.ru

УДК519.173, 519.173, 656.022

DOI: 10.15587/2313-8416.2015.44381

ТОПОЛОГІЧНА ЕВРИСТИКА В РОЗВ'ЯЗАННІ ПРОБЛЕМИ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ (VRP)

© І. М. Найдюнов

Роботу присвячено створенню математичної моделі топології дорожньої мережі для розв'язання задач маршрутизації транспортних засобів. Пропонується евристика «ядер і хвостів» як спосіб зниження розмірності і композиції топологічно обґрунтованих наборів точок в єдину групу. У статті розкрито 4 етапи побудови моделі. Визначено перспективи подальшого вдосконалення алгоритму з переходом до паралельного наповнення маршрутів

Ключові слова: маршрутизація, VRP, кластерна маршрутизація, CluVRP, задача комівояжера, евристики, топологія, теорія графів, «ядра і хвости»

The creation of a mathematical model of the topology of the road network to solve vehicle routing problem (VRP) is presented in the paper. We propose heuristic of "cores and tails" as a way to reduce the dimensions and to compose topological-based sets of points into a single group. It is described four stages of creating a model. The prospects of further improvement of the algorithm with the transition to parallel filling routes are outlined

Keywords: routing, VRP, cluster routing, CluVRP, traveling salesman problem, heuristics, topology, graph theory, "cores and tails"

1. Вступ

На сьогоднішній день, в умовах глобального дефіциту природних ресурсів і ситуативного підвищення цін на паливо, як ніколи актуальна проблема скорочення витрат на транспортну логістику. Для досягнення цієї мети пропонується цілий комплекс заходів організації робочого процесу, моніторингу, планування та мотивації персоналу, але ключовим з

цієї низки є автоматична побудова оптимальних маршрутів транспортних засобів.

У реальних практичних задачах вхідні умови набагато складніші, ніж в класичній проблемі комівояжера [1] – це в першу чергу розподілення задачі на наявні транспортні засоби (VRP [2]), врахування максимального об'єму та ваги транспортного засобу (CVRP [3]), дозволених часових вікон

точок доставки (VRPTW [4]), знання водіями місцевості і конкретних точок доставки, та багато інших.

Всі наявні алгоритми вирішення проблем маршрутизації транспортних засобів мають один суттєвий недолік, який полягає в тому, що вони не враховують топології дорожньої мережі. Зазвичай алгоритми працюють лише з матрицею відстаней між точками доставки як моделлю транспортної мережі, за якою не очевидно, які точки знаходяться компактно в одній тій самій місцевості, наприклад, вже лежать на шляху проїзду машини (попутні), тобто точки зв'язані між собою таким чином, що витративши час на приїзд до однієї точки, необхідно обов'язково відвідати інші точки в цьому районі тим самим транспортним засобом. Для вирішення цього завдання часто використовуються алгоритми кластеризації, але вони не дають добрих результатів у неевклідовому просторі дорожньої мережі через складність визначення однозначних кластерів, в тому числі з причин того, що в залежності від напрямку проїзду відстані між точками відрізняються. У результаті цього логістичні рішення, отримані на основі наявних алгоритмів, не задовольняють вимог логістів і їх практично неможливо впровадити в реальну практику. Ключовим питанням на шляху до комплексного розв'язання проблеми є пошук евристик для математичного моделювання топології транспортної мережі.

2. Аналіз літературних даних і постановка проблеми

Пошук математичних засобів автоматизації розв'язання логістичних завдань має свою тривалу історію. Первинною задачею, яка визначає увесь клас, була задача комівояжера [1]. У класичній постановці це задача на знаходження оптимального порядку проїзду одним комівояжером між містами, з відомою матрицею відстаней між ними. Похідна задача маршрутизації транспортних засобів (VRP) оперує вже декількома транспортними засобами і постає проблема не тільки вибрати найкращий порядок, а ще і оптимально розподілити точки між одиницям транспорту.

Ця задача також має велику кількість похідних, але розвиток цих задач рухається в двох основних напрямках. Перший напрям – це врахування особливостей одиниці транспортного засобу (машини), таких як: вантажомісткість і об'єм кузова [3] можливість машини як доставляти так і підбирати грузи [5], сумісність по типам вантажу, температурним режимам та іншим параметрам. Інший напрям це врахування часових характеристик, таких як обмеження часових вікон у точках доставки, часу на обслуговування точок [4], обмеження щодо робочого часу водія та інші.

Тим не менше в переважній більшості задач робота з точками доставки іде по заданій матриці відстаней, і не враховує топологію реальних доріг. Деякі автори включають в постановку задачі і частину знаходження матриці відстаней, чи все ж таки певною мірою враховують топологію, але це носить достатньо локальний характер. Так Olli

Bräysy в одній зі своїх останніх робіт[6] враховує штрафи за повторити та заборони поворотів в деяких місцях.

Крім цього існує клас задач під назвою Кластерна проблема маршрутизації транспортних засобів (Clustered Vehicle Routing Problem, CluVRP). Особливість такого типу задач полягає в тому, що набір точок розбивається на N непересічних і неперожних підмножин, названих кластерами. Вперше цю постановку ввели Sevaux та Sorensen в 2008 році [7]. Точки в кожному кластері повинні бути відвідані послідовно, так що одиниця транспорту, яка відвідала точку в кластері, не може перейти до іншого кластера, доки всі точки в ньому не будуть відвідані. Пізніше, в 2014 Battarra M. *та ін.* [8] запропонували варіант точного рішення цієї задачі, а Vidal T. *та ін.* [9] запропонували евристичні методи, оскільки точне рішення занадто повільне для великих задач чи задач реального часу.

Але така постановка мотивована не покращенням вихідної вартості маршруту за рахунок кращого врахування топології, а іншими причинами, такими як особливості упаковки і перевозу вантажів, а також ознайомленістю водіїв з певною місцевістю. Такий підхід скоріше програє в таких показниках як кілометраж, що тим не менш може бути компенсовано знайомістю водіїв з місцевістю. Причина полягає в тому, що кластери однозначно задаються один раз на початку роботи алгоритму, і можуть скоріше суперечити особливостям реальної топології, в тому числі тому, що в такому неевклідовому просторі як дорожня мережа в залежності від напрямку проїзду відстані між точками відрізняються.

Крім того, алгоритми кластеризації сортують об'єкти на взаємовиключні категорії, але розбиття точок території на топологічні одиниці не таке однозначне – точка може знаходитися на переїзді між містами, або межі районів міста можуть зливатися тощо. Тобто, виникає необхідність замість взаємовиключних кластерів виділити таке топологічне розбиття множини точок території на зони, яке б дозволяло деякі точки трактувати одночасно належними до декількох сусідніх зон.

Практичне описання проблеми врахування топології транспортної мережі

Врахування топології мережі доріг вимагає, щоб алгоритм працював таким чином, що коли машина приїжджає в певну місцевість і включає в свій маршрут там певну кількість точок доставки, вона не переїжджала до іншої місцевості, поки не візьме в свій маршрут всі точки даної місцевості.

Зазвичай для вибору порядку наповнення машини використовувався так званий «жадібний алгоритм» [10] – наступною точкою для розгляду і додавання її в маршрут машини вибиралася найближча до вже наявних в маршруті. Це призводить до таких випадків, коли алгоритм визначає маршрути для машин не оптимально – найближчі точки спрямовують фронт наповнення машини в якомусь одному напрямку, при цьому деякі інші точки, які теж повинні були потрапити в цю машину, пропускаються з причин того, що поки до цієї точки дійде черга –

машина вже не може її обслужити чи то за браком часу, чи то за браком місця.

Таким чином, якщо перші машини наповнюються достатньо компактно (кучно), то останнім доводиться їздити і збирати пропущені точки зрізних місцевостей, а то й з усієї території, що призводить до суттєвого збільшення кілометражу проїзду і навіть веде до зростання кількості необхідних машин для розв'язання логістичного завдання доставки.

Більше того, такі рішення, в яких в певну відокремлену ділянку території їде декілька машин, коли там могла б справитися й одна, ніколи не приймуть кінцеві користувачі програми – логісти. Вони з першого погляду бачать такі проблеми, і якщо програма буде давати такі результати, то все що залишиться логістам – перероблювати такі ділянки вручну, що зводить цінність програми нанівець, адже вона якраз призначена для зменшення кількості ручної роботи, в тому числі і за рахунок зручної візуалізації рішень для користувача.

Отже нам потрібно розробити таку евристику, яка дозволить розуміти зв'язки точок в задачі, щоб взявши певні точки обов'язково взяти й інші з певної топологічної ділянки цією ж машиною, оскільки вони зв'язані між собою в топології транспортної мережі. Увести таку логіку безпосередньо для точок доставки доволі складно, оскільки таких роздроблених зв'язків буде дуже багато і всі будуть «тягти» машину в різні боки одночасно. Тому постає необхідність переходу до використання якихось сукупних одиниць, які б позначали собою *топологічні одиниці території*. Цей перехід до більш крупних одиниць топології (більших, ніж окрема точка доставки) дасть змогу зменшити розмірність задачі, а отже і зменшити час на обрахунок, що є особливо важливим для задач великого об'єму.

3. Ціль та завдання дослідження

Мета дослідження – пошук евристик для забезпечення компактного (кучного) групування точок доставки в одиницю транспорту ("машину") з урахуванням реальної топології дорожньої мережі, яке легко масштабувалося б для задач великого об'єму (largescaleproblem).

Для досягнення зазначеної мети потрібно виконати наступні завдання:

– розробити топологічну евристику, що забезпечить перехід до укрупнених одиниць опису топології транспортної мережі.

– розробити способи реалізації топологічної евристики – переходу від одиничних точок доставки до укрупнених одиниць, які враховують особливості топології місцевості.

– розробити і апробувати алгоритм розв'язання логістичної задачі доставки з урахуванням топології транспортної мережі на певній місцевості.

4. Побудова математичної моделі топології дорожньої мережі на основі евристики ядер і хвостів

4.1. Опис суті евристики «ядер і хвостів»

Ідея створення топологічних зон, до якої ми дійшли за результатами аналізу літературних джерел

та практичних запитів на автоматизацію логістичних задач, реалізувалася нами в *евристичі «ядер і хвостів»*. Суть задуму полягає в тому, що ми виділяємо деякі скупчення точок як «ядра», а інші позначаємо як «хвости» до одного чи декількох ядер. Таким чином, ми одночасно можемо забезпечити і збереження інформації про те, які існують взаємні зв'язки між точками і їх групами, і при цьому ж уникаємо проблем, пов'язаних із неоднозначністю топології дорожньої мережі. Як раз однозначність топології на деяких ділянках і є для нас показником що ці ділянки носять характер ядер.

Критерій для знаходження ядер був вибраний такий: ядра це такі угруповання точок, в яких всі точки транзитивно попутні одна до одної. Тобто, це такі угруповання, в яких точки циклічно близькі між собою. Мова про циклічність і транзитивність іде від того, що точки не обов'язково повинні бути всі попарно близькі між собою. Наприклад, точки розташовані на вулицях з одностороннім рухом не можуть бути одночасно близькі до всіх сусідніх точок, але якщо ми починаємо їхати навколо кварталу, то навіть по вулицям з одностороннім рухом ми можемо об'їхати квартал і приїхати в те саме місце, по дорозі постійно рухаючись до близької точки. Це і є цикл, який свідчить про наявність топологічної особливості на цій ділянці, а значить ці точки утворюють топологічну зону, названу нами «ядро».

Хвости, в свою чергу, – це всі інші точки, які не ввійшли в ядра, при чому для кожного хвоста зберігається інформація про те, до якого (чи яких) ядер він може відноситись, а для ядер – відповідно інформація про те, які в них є хвости. Таким чином, відвідавши всі точки ядра ми маємо інформацію про те, що ми не можемо поїхати до наступного ядра до тих пір, доки не розглянемо всі хвости цього ядра. Оскільки хвости можуть відноситись як до одного, так і до декількох ядер, то ми можемо ввести поняття *обов'язкових хвостів*, тобто тих, які відносяться лише до одного ядра, або тих, всі ядра яких уже відвідані. Таким чином, розглядаючи хвости кожного наступного ядра, ми повинні відвідати всі обов'язкові хвости ядра тією ж машиною, що і саме ядро.

Тобто результатом введення такої моделі є те, що ми переходимо від рівнозначного набору всіх точок задачі, для яких у нас немає жодної інформації про їхню взаємну топологію, крім матриці попарних відстаней між ними, до певної структури ядер і хвостів, які відображають топологію дорожньої мережі.

Більше того, маючи зв'язки між ядрами і хвостами, ми отримуємо можливість при роботі алгоритму обмежитися перебором лише серед сусідніх об'єктів. Це дозволяє суттєво знизити асимптотичну складність алгоритму, перейшовши від параметра N , заданого як кількість точок, до параметру кількості транспортних засобів, оскільки ми можемо уникнути переборів по всьому масиву точок. Таким чином, для приблизно однорідного транспортного парку, не залежно від асимптотичної складності розгляду однієї точки (яка залежить від кількості наявних в рейсі точок) ми можемо прирівняти час розгляду однієї одиниці транспорту до константного, а отже для ве-

ликих задач отримати складність задачі складність задачі амортизовано лінійну від кількості транспорту.

4.2. Етапи створення топологічної моделі «ядер і хвостів»

Перший етап – створення *графу попутності*

Матрицю відстаней між точками можна трактувати як повний граф зв'язків між всіма точками. Для того, щоб побудувати ланцюжок точок, за якими рухається машина, ми проріджуємо повний граф, залишаючи для кожної точки тільки первинні зв'язки – такі, в яких найкоротший шлях по дорожній мережі не проходить через інші точки. Опосередковані зв'язки, в яких досягнення точки відбувається тільки за умови проїзду через іншу точку, видаляються з повного графу. Таким чином, отримуємо граф попутності.

Другий етап – створення *графу сусідства*

Для побудови графу сусідства ми повинні ще раз прорідити отриманий граф попутності, залишивши для кожної точки лише декілька її найближчих сусідів. Питання в тому, як визначити поріг, за яким визначати зв'язані точки як близькі або далекі? Підібрати цей поріг як константу неможливо, адже навіть в одній задачі масштаби відстаней між точками досить різняться, а для різних задач вони можуть виявитись взагалі несумісні. Тому поріг – *радіус сусідства* – знаходиться для кожної точки свій. Радіус сусідства – це відстань до k -го найближчого сусіда помножений на *коефіцієнт надмірності*. Цей коефіцієнт дозволяє в деяких випадках густо розташованих точок, взяти більше за k найближчих сусідів.

Ця ідея була запозичена з алгоритму магистральних ієрархій, розробленого Р. Sanders та D. Schultes [11] в 2005–2007 рр., для швидкої побудови найкоротших шляхів у графах великого розміру, таких як дорожня мережа всього світу. В цій роботі радіус сусідства використовується не для побудови графу сусідства, а просто як проміжний етап для позначення околу, при перевищенні якого потрібно переходити на вищий рівень ієрархії. Крім того, алгоритм працює суто за самим графом дорожньої мережі, де вершинами позначені повороти і перехрестя, а ребрами – прямі ділянки шляху.

Величина параметру k кардинально впливає на розв'язок задачі. Якщо параметр дорівнює одиниці, то в кожній точці ми знаходимо лише одного найближчого сусіда, що еквівалентно жадібному алгоритму. Якщо ми збільшуємо цей параметр, то щільність графу швидко зростає, аж до повного графу і відповідно повного перебору.

У цілому можна сказати, що розв'язок задачі VRP лежить в балансі між цими двома крайностями, де жадібний алгоритм виконується швидко, але легко попадає в локальний мінімум, а повний перебір ігнорує локальні мінімуми і знаходить точний результат, але виконується дуже повільно (адже повний перебір – це трансобчислювальна задача, тобто така, що лежить за межами сучасних обчислювальних можливостей).

На практиці допустимі результати, прийнятні для подальшої обробки, дають лише значення $k=2$ та

іноді 3. Цей параметр кладає баланс між розміром ядер, їх кількістю, відсотком точок, які ввійшли до ядер, і відсотком точок, які лишилися хвостами. Для нормальної подальшої роботи найкраще мати якомога більше маленьких ядер, і приблизно рівне співвідношення точок у ядрах і хвостів. Параметр $k=1$ дає занадто розріджений граф, на якому практично не знаходиться ядер і абсолютна більшість точок попадають в хвости, а $k>3$ навпаки дає занадто щільний граф, в якому вся територія розпізнається, як одне-два величезних ядра без хвостів.

Параметр коефіцієнту надмірності впливає не так кардинально і дозволяє більш тонко налаштувати щільність графу.

Третій етап – *знаходження ядер*

Оскільки граф сусідства показує нам зв'язки між точками, які являються сусідами, і проїзд між якими зручний в котромусь із напрямів, ми можемо звести задачу пошуку ядер до задачі пошуку компонент сильної зв'язності в орієнтованому графі. Таким чином ми знаходимо такі групи точок, які є транзитивно сусідами одна одній.

Компонента сильної зв'язності орієнтованого графу – це найбільший підграф, такий що в ньому існує шлях з будь-якої вершини підграфу до кожної з інших його вершин. Цикл в орієнтованому графі є сильно зв'язним, і кожна нетривіальна компонента сильної зв'язності містить щонайменше один цикл, отже орієнтований граф є ациклічним тоді і лише тоді, коли він не має компонент сильної зв'язності з більш як однією вершиною. Найпростіший алгоритм пошуку компонент сильної зв'язності полягає в тому, щоб побудувати неорієнтований граф на основі транзитивного замикання даного орієнтованого графу, а потім скористатися звичайним алгоритмом пошуку компонент зв'язності в неорієнтованому графі. Крім цього існує 3 алгоритми для вирішення цієї задачі за лінійний час. Алгоритм Косараджу [12], алгоритм Тарджана [13] і алгоритм компонент сильної зв'язності по шляхах (алгоритм Габова) [14]. Всі дієво знаходять компоненти сильної зв'язності орієнтованого графу, але алгоритм Тарджана і алгоритм по шляхах сприятливіші на практиці, бо вони працюють швидше і більше підходять для задач з великою кількістю точок, оскільки потребують лише один пошук у глибину замість двох. Нами було обрано алгоритм Тарджана.

Оскільки алгоритми пошуку сильних компонент зв'язності розбивають на такі компоненти всю задачу, то в них попадають і хвости. Але на відміну від ядер, компоненти зв'язності, яким відповідають хвости, зазвичай дуже маленькі. Ми вводимо пороговий параметр, який задає мінімальний розмір компоненти зв'язності, для того, щоб вона могла вважатися ядром. Співвідношення кількості точок у ядрах та хвостів є таким, що задовольняє вимоги алгоритму у випадку, якщо задати параметр рівним 3 або 2. Значення 1 не задовольняє умови, бо тоді всі точки попадуть у ядра, що знизить нашу розмірність, але не дасть уявлення про обов'язкові точки при відвідуванні певної території, а значення параметру більші за 3 призводять до знаходження

малої кількості ядер і великої кількості хвостів, такої, що одна машина не може розвезти всі обов'язкові хвости навіть одного ядра.

Четвертий етап – *знаходження хвостів, зв'язаних з ядрами*.

Для знаходження хвостів ядра ми позбавляємо наш граф інформації про орієнтованість, перетворюючи всі зв'язки на двоспрямовані. Це робиться для збільшення кількості зв'язаних сусідніх ядер у хвоста, щоб гарантувати що хвіст буде відомий для всіх ядер, які можуть мати до нього відношення.

Отримавши неорієнтований граф сусідства, ми виконуємо в ньому частковий пошук в глибину з кожної точки кожного ядра, записуючи ядро з якого ми почали пошук в тимчасовий список для кожної розглянутої точки, як можливий кандидат на ядро, до якого належить цей хвіст. Критерієм зупинки пошуку є розглядання точки, яка є членом будь-якого ядра. Така точка не потрапляє до списку, і з неї не продовжується пошук по суміжним вершинам графу.

Отримавши списки можливих кандидатів ядер для кожного хвоста ми повинні відсіяти в цьому списку зайві варіанти. Оскільки ми виконуємо пошук в глибину зупиняючись тільки досягнувши ядер, то ми цілком можемо опинитись в ситуації, коли по зв'язкам між хвостами ми минули всі ядра і досягли якогось віддаленого ядра, яке насправді ніяк не впливає на даний хвіст. Той факт, що ми працювали з неорієнтованою версією графа, тільки збільшує таку ймовірність.

Для відсіювання ядер ми задаємо радіус зв'язності з ядрами для кожного хвоста. Це порогове значення, яке визначає максимально допустиму відстань до найближчої точки ядра, для того, щоб можна було вважати, що хвіст зв'язаний з цим ядром. Це значення визначається як відстань до найдалшої точки найближчого ядра.

4.3. Алгоритм використання топологічної моделі

Для побудови маршрутів використовується алгоритм послідовної обробки транспортних засобів, для включення точки в маршрут використовуються евристика локального пошуку *ort3*[]. Розроблена евристика ядер і хвостів впливає в першу чергу на порядок розгляду точок.

Алгоритм наповнення кожної машини складається з чотирьох етапів, які циклічно повторюються до тих пір, поки машина не буде заповнена майже повністю:

1. Вибір наступного найближчого ядра жадібним способом, за найменшою відстанню між точками.

2. Відвідування всіх точок ядра. Якщо якась із точок не може бути відвідана, то і все ядро вважається не відвіданим.

3. Відвідування всіх обов'язкових хвостів цією ж машиною.

4. Якщо якісь із хвостів також зв'язані з іншими ядрами, що потрапили в інші машини – спробувати розмістити їх в тих машинах.

Алгоритм реалізовано на мові C++, щ дало змогу перевірити ефективність його використання для розв'язання логістичних задач.

5. Результати застосування евристики «ядер і хвостів»

Алгоритм застосований для розв'язання модельної задачі зі 120 точок, які мають бути розвезені за допомогою 5 транспортних засобів. Оцінка ефективності дії алгоритму проводилася як порівняння результатів із результатами, отриманими при застосуванні жадібного алгоритму. При застосуванні жадібного порядку розгляду точок, маршрут кожної наступної машини виходив менш компактним. При застосуванні алгоритму, в якому порядок розгляду оснований на евристиці ядер і хвостів, всі маршрути для п'яти машин формувалися компактно.

Другий етап оцінки результатів застосування евристики проводився на реальних виробничих логістичних задачах, характерних для компаній доставки питної води в місті. При тестуванні на задачах великого розміру спостерігалось не тільки покращення компактності результату, а й прискорення швидкості обрахунку маршруті транспортних засобів.

6. Ефективність врахування топології транспортних мереж зі застосуванням евристики ядер і хвостів

Застосування топологічної евристики ядер і хвостів, дає покращення показників розв'язання логістичних задач за рахунок врахування топології транспортної мережі. Проте варто відмітити, що при значному збільшенні масштабу задачі все ще залишається не достатньо ефективним. Зокрема може виникнути ситуація, коли на одну топологічну територію приходиться декілька ядер, одне з яких обслуговується однією машиною а решта – іншою, що зменшує економічну вигоду рішення. Доцільно продовжити дослідження топологічних характеристик транспортної мережі і пошук топологічних евристик в цілому. Перспективним видається також напрям застосування евристики ядер і хвостів до паралельного алгоритму наповнення декількох машин одночасно, що може стати наступним кроком удосконалення способів розв'язання проблеми маршрутизації для задач великого масштабу.

7. Висновки

Таким чином, запропоновано нову евристику ядер і хвостів, яка забезпечує компактне групування точок доставки в одиницю транспорту з урахуванням реальної топології дорожньої мережі.

Розроблена топологічна евристика дає змогу перейти до укрупнених одиниць опису топології транспортної мережі і враховує невизначеність реальної топології місцевості, яку не вдавалося подолати застосуванням кластерного підходу.

На основі теорії графів розроблено спосіб побудови математичної моделі топології місцевості, який складається з чотирьох етапів: визначення графу присутності, графу сусідства, ядер і хвостів.

Розроблено алгоритм циклічної побудови маршрутів відвідування точок доставки транспортними засобами для певної території з урахуванням її топології. Апробація алгоритму показала його переваги перед жадібним алгоритмом за параметрами компактності маршрутів для усіх одиниць транспортних засобів і швидкості рішення.

Перспективу подальшого дослідження складає застосування евристики ядер і хвостів до розв'язання проблеми маршрутизації для задач великого масштабу.

Література

- Applegate, D. L. The traveling salesman problem: a computational study [Text] / D. L. Applegate, R. E. Bixby, V. Chvátal, W. J. Cook. – Princeton University Press, 2007. – 608 p.
- Dantzig, G. B. The Truck Dispatching Problem [Text] / G. B. Dantzig, J. H. Ramser // Management Science. – 1959. – Vol. 6, Issue 1. – P. 80–91. doi: 10.1287/mnsc.6.1.80
- Baldacci, R. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts [Text] / R. Baldacci, N. Christofides, A. Mingozzi // Mathematical Programming. – 2007. – Vol. 115, Issue 2. – P. 351–385. doi: 10.1007/s10107-007-0178-5
- Ellabib, I. An Experimental Study of a Simple Ant Colony System for the Vehicle Routing Problem with Time Windows [Text] / I. Ellabib, A. B. Otman, P. Calamai // Lecture Notes in Computer Science. – 2002. – Vol. 2463. – P. 53–64. doi: 10.1007/3-540-45724-0_5
- Bianchessi, N. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery [Text] / N. Bianchessi, Righini // Computers & Operations Research. – 2007. – Vol. 34, Issue 2. – P. 578–594. doi: 10.1016/j.cor.2005.03.014
- Bräysy, O. The mixed capacitated general routing problem with turn penalties [Text] / O. Bräysy, E. Martínez, Y. Nagata, D. Soler // Expert Systems with Applications. – 2011. – Vol. 38, Issue 10. – P. 12954–12966. doi: 10.1016/j.eswa.2011.04.092
- Sevaux, M. Hamiltonian paths in large clustered routing problems [Text] / M. Sevaux, K. Sørensen // Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing. – Troyes, France, 2008.
- Battarra, M. The clustered vehicle routing problem [Text] / M. Battarra, G. Erdoğan, D. Vigo // Operations Research. – 2014. – Vol. 62. – P. 58–71.
- Thibaut, V. Hybrid Metaheuristics for the Clustered Vehicle [Text] / T. Vidal, M. Battarra, A. Subramanian, G. Erdoğan // Computers & Operations Research. – 2015. – Vol. 58, Issue 1. – P. 87–99. doi: 10.1016/j.cor.2014.10.019
- Кормен, Т. Жадные алгоритмы. изд. 2-е. [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн // Алгоритмы: построение и анализ / Под ред. И. В. Красикова. – Москва: Вильямс, 2005. – 1296 с.
- Sanders, P. Engineering Highway Hierarchies [Text] / P. Sanders, D. Schultes // 9th Workshop on Algorithm Engineering and Experiments. – Mehlhorn: SIAM, 2007. – P. 36–45. doi: 10.1137/1.9781611972870.4
- Sharir, M. A strong connectivity algorithm and its applications to data flow analysis [Text] / M. Sharir // Computers and Mathematics with Applications. – 1981. – Vol. 7, Issue 1. – P. 67–72. doi: 10.1016/0898-1221(81)90008-0
- Tarjan, R. E. Depth-first search and linear graph algorithms [Text] / R. E. Tarjan // SIAM Journal on Computing. – 1972. – Vol. 1, Issue 2. – P. 146–160. doi: 10.1137/0201010
- Gabow, H. N. Path-based depth-first search for strong and biconnected components [Text] / H. N. Gabow // Information Processing Letters. – 2000. – Vol. 74, Issue 3–4. – P. 107–114. doi: 10.1016/s0020-0190(00)00051-x

References

- Applegate, D. L., Bixby, R. M., Chvátal, V., Cook, W. J. (2007). The Traveling Salesman Problem: a computational study. Princeton University Press, 608.
- Dantzig, G. B., Ramser, J. H. (1959). The Truck Dispatching Problem. Management Science, 6 (1), 80–91. doi: 10.1287/mnsc.6.1.80
- Baldacci, R., Christofides, N., Mingozzi, A. (2007). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming, 115 (2), 351–385. doi: 10.1007/s10107-007-0178-5
- Ellabib, I., Otman, A. B., Calamai, P. (2002). An Experimental Study of a Simple Ant Colony System for the Vehicle Routing Problem with Time Windows. Lecture Notes in Computer Science, 2463, 53–64. doi: 10.1007/3-540-45724-0_5
- Bianchessi, N., Righini (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research, 34, 578–594. doi: 10.1016/j.cor.2005.03.014
- Bräysy, O., Martínez, E., Nagata, Y., Soler, D. (2011). The mixed capacitated general routing problem with turn penalties. Expert Systems with Applications, 38 (10), 12954–12966. doi: 10.1016/j.eswa.2011.04.092
- Sevaux, M., Sørensen, K. (2008). Hamiltonian paths in large clustered routing problems. Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing. Troyes, France.
- Battarra, M., Erdoğan, G., Vigo, D. (2014). The clustered vehicle routing problem. Operations Research, 62, 58–71.
- Vidal, T., Battarra, M., Subramanian, A., Erdoğan, G. (2015). Hybrid metaheuristics for the Clustered Vehicle Routing Problem. Computers & Operations Research, 58, 87–99. doi: 10.1016/j.cor.2014.10.019
- Feed, T., Leiserson, C., Rivest, R., Stein, K. (2005). Chapter 16: Greedy algorithms. Introduction to Algorithms (ed. 2nd). Moscow: Williams, 1296.
- Sanders, P., Schultes, D. (2007). Engineering Highway Hierarchies. 9th Workshop on Algorithm Engineering and Experiments, SIAM. Mehlhorn, 36–45. doi: 10.1137/1.9781611972870.4
- Sharir, M. (1981). A strong connectivity algorithm and its applications to data flow analysis. Computers and Mathematics with Applications, 7 (1), 67–72. doi: 10.1016/0898-1221(81)90008-0
- Tarjan, R. E. (1972). Depth-first search and linear graph algorithms. SIAM Journal on Computing, 1 (2), 146–160. doi: 10.1137/0201010
- Gabow, H. N. (2000). Path-based depth-first search for strong and biconnected components. Information Processing Letters, 74 (3–4), 107–114. doi: 10.1016/s0020-0190(00)00051-x

*Рекомендовано до публікації д-р техн. наук, професор Биков В. Ю.
Дата надходження рукопису 19.05.2015*

Найдьонов Іван Михайлович, кафедра біомедичної кібернетики, Національний технічний університет України «Київський Політехнічний Інститут», пр. Перемоги, 37, м. Київ, Україна, 03056
E-mail: samogot@gmail.com

УДК 004.021

DOI: 10.15587/2313-8416.2015.44357

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ОЦІНКИ СТАНУ ЗДОРОВ'Я СТУДЕНТІВ

© Ю. В. Антонова-Рафі, М. В. Нікітенко

У статті розглядаються питання щодо використання інформаційних технологій оцінки стану здоров'я студентів вищих навчальних закладів. Також розглянуто методи оцінки рівня фізичного здоров'я по методиці Апанасенко та Науменко, дослідження проби Мартіне при аналізі стану студентів при фізичних навантаженнях. Було пораховано і приведено в індекс маси тіла, життєвий індекс, силовий індекс (динамометрія руки), індекс Робінсона і час відновлення ЧСС

Ключові слова: програмне забезпечення, оцінка фізичного рівня здоров'я, проба Мартіне, моніторинг стану здоров'я студентів

The article deals with the use of information technology for health assessments of students. Also, the evaluation methods of physical health by Apanasenko-Naumenko method, research of Martin test at analysis of the students during physical exercises. It was counted and given a body mass index, life index, strength index (hand dynamometry), the Robinson index and the recovery of heart rate

Keywords: software, evaluation of physical health, Martin test, monitoring of the health of students

1. Вступ

На даний момент аналіз ситуації стосовно фізичного навантаження студентів є на недорозвинутому рівні. Вплив фізичних вправ на педагогічні результати опосередкований фізіологічними і біологічними механізмами. Інакше кажучи, навантаження є причиною тих адаптаційних змін в організмі, від характеру і величини яких залежить результат.

Найціннішими для фізичного виховання студентів є великі (тобто розвиваючі) і середні (закріплюючі) навантаження, використання їх дозволяє вчителю забезпечити оздоровчу спрямленість занять і управління розвитком організму студентів з урахуванням вимог їх всебічного фізичного вдосконалення.

Потрібно дослідити можливості та задачі моніторингу фізичного стану студентів, визначити та реалізувати найбільш ефективні методи тестування та аналізу за допомогою існуючих методів. Визначити динаміку фізичної підготовленості молодших школярів із різною руховою активністю.

2. Постановка проблеми

Кінець ХХ – початок ХХІ ст. ознаменувався переходом від індустріального до інформаційного суспільства. Інформаційні технології проникають і всі сфери людської діяльності, суттєво змінюючи життя кожного. Розробка й використання інформаційних систем і систем управління у сфері фізичного виховання та спорту можуть розглядатися, як один із перспективних напрямів у зв'язку з погіршенням здоров'я нації та необхідністю виявляти негативні тенденції у молоді до моменту виявлення серйозних патологій. Крім того, застосування сучасних технологій дозволяє не тільки визначити функціональний стан людини, а й обирати оптимальний шлях для його корекції. Саме тому, аналіз існуючих інформаційних систем у фізичному вихованні і спорті, вивчення основних сфер застосування інформаційних технологій і розробка рекомен-

дацій із використанням інформаційних систем є особливо актуальним.

Предметом дослідження є стан, особливості та динаміка змін фізичного та фізіологічного здоров'я студентів, засоби й методи моніторингу здоров'я, організаційні та методичні заходи щодо застосування технологій оздоровлення та здорового способу життя, засобів фізичної культури та спорту як невід'ємних складових гармонійного розвитку особистості.

Метою є розробка і впровадження інформаційного та медико-інженерного забезпечення супроводження моніторингу здоров'я студентської молоді НТУУ «КПІ» на базі інформаційних технологій. Розробка програмного продукту на основі даних методів і реалізація їх у вигляді ProgramApplication.

Реалізація цієї мети передбачає виконання наступних завдань:

1) Розробка інформаційного та медико-інженерного забезпечення супроводження моніторингу стану організму студентів на базі інформаційних технологій;

2) Розрахунки та їх аналіз:
– дослідження проби Мартіне при аналізі стану студентів при фізичних навантаженнях;

– визначення оцінки рівня фізичного здоров'я по методиці Апанасенко та Науменко;

– дослідження індекса маси, життєвий індекс, силовий індекс, індекс Робінсона, час відновлення ЧСС та загальна оцінка рівня фізичного здоров'я.

3) Впровадження системи контролю за динамікою змін у функціонуванні основних систем організму студентів протягом навчання;

4) Впровадження системи медико-інженерних технологій оздоровчого спрямування на базі отриманих результатів.

3. Оцінка рівня здоров'я

Загальна оцінка здоров'я визначається сумою балів і дозволяє розподілити всіх практично здорових