

Эффективный метод решения задачи обхода мегаполисов при ограничениях предшествования специального типа

А.Г. Ченцов
agchentsov@mail.com

Д.М. Хачай
daniil.khachay@gmail.com

ИММ УрО РАН (Екатеринбург)
УрФУ (Екатеринбург)

Аннотация

Рассматривается задача обхода мегаполисов с фиксированным числом городов и специальным образом заданными отношениями предшествования, известная также как асимметрическая обобщенная задача коммивояжера (AGTSP). Для поиска оптимального решения задачи приводится схема динамического программирования, развивающая схему Хелда–Карпа, разработанную для классической постановки задачи коммивояжера; обосновываются условия, при которых трудоемкость предложенного алгоритма полиномиально, в частности, линейно зависит от числа мегаполисов.

1 Введение

Обобщенная задача коммивояжера (GTSP) является расширением классической задачи коммивояжера (TSP), в которой множество городов разделено на непересекающиеся между собой мегаполисы (кластеры), причем каждый кластер должен быть посещен искомым маршрутом в точности один раз. Задача обладает широким спектром приложений, например в задачах маршрутизации [9], в задаче демонтажа отработанного реактора на АЭС [5], а также при моделировании высокоточной резки металла [18].

Существует множество подходов для нахождения оптимальных и субоптимальных решений этой задачи. На первый взгляд, наиболее естественным представляется подход, состоящий в сведении обобщенной задачи коммивояжера к подходящей постановке обычной TSP. В самом деле, известна [17] полиномиальная сводимость задачи GTSP к подходящей асимметрической задаче TSP, сохраняющая ее стоимость, так что для решения GTSP могут применяться многочисленные алгоритмы и эвристики, разработанные для классической TSP [12, 14]. К сожалению, результирующая задача TSP теряет многие полезные свойства исходной постановки, и для ее аппроксимации не удается применить такие известные эффективные алгоритмы, как алгоритм Кристофидеса [6] для метрической TSP, полиномиальную приближенную схему (PTAS) Ароры для Евклидовой TSP [1], ее обобщение на случай задачи о нескольких коммивояжерах [15, 16].

Еще один подход связан с адаптацией некоторых эволюционных методов, как, например, генетических алгоритмов [4, 10], эвристик муравьиной колонии [13] и т.д. Результаты численных экспериментов

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A.A. Makhnev, S.F. Pravdin (eds.): Proceedings of the 47th International Youth School-conference “Modern Problems in Mathematics and its Applications”, Yekaterinburg, Russia, 02-Feb-2016, published at <http://ceur-ws.org>

показывают, что в некоторых случаях данный подход действительно позволяет эффективно получить решения, близкие к оптимальным. Однако основным его недостатком является отсутствие теоретических оценок точности аппроксимации и вычислительной сложности для всех перечисленных выше алгоритмов.

В рамках третьего подхода изучаются нетривиально полиномиально разрешимые подклассы исследуемой задачи. В то время как для классической постановки задачи коммивояжера известно большое количество таких подклассов (см., например, [8]), для задачи GTSP подобные результаты нам не известны.

Данная работа посвящена описанию полиномиально разрешимого подкласса задачи GTSP, отягощенного дополнительными ограничениями предшествования специального вида. Полученные результаты являются естественным обобщением результатов, представленных в [2] для классической TSP.

Рассмотрим наиболее общий случай задачи GTSP. В данной постановке для любой пары смежных узлов u и v , стоимости прямых и обратных перемещений не одинаковы. Чтобы подчеркнуть данную *асимметрию*, будем называть такую постановку Асимметрической Обобщенной Задачей Коммивояжера (AGTSP).

Ограничения предшествования естественны для AGTSP и восходят к конкретным прикладным задачам. В нашем случае эти ограничения определяют порядок посещения мегаполисов и могут быть легко интерпретированы.

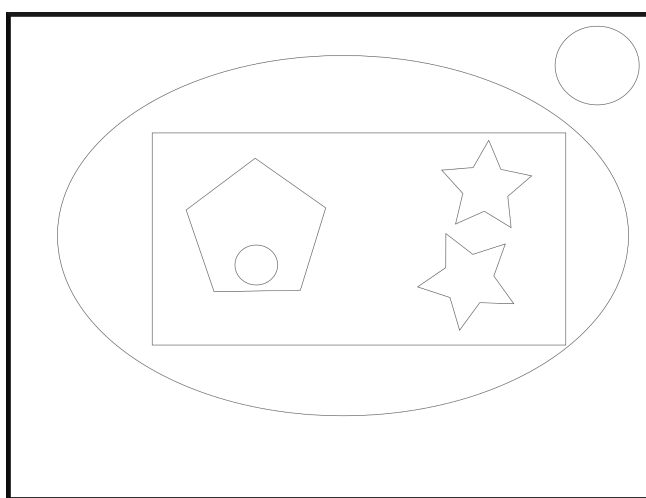


Рис. 1: Задача фигурной резки металла

Например, в задаче высокоточной резки металла из исходного стандартного листа требуется вырезать детали различной формы. В соответствующей постановке AGTSP, каждой детали сопоставляется конечный мегаполис, состоящий из точек врезки и выключения инструмента, размещенных по ее контуру (рис. 1). Как видно, на исходном листе одни детали могут оказаться вложенными в другие. Технология резки, согласно которой вложенные детали должны быть вырезаны раньше объемлющих, задает естественные ограничения предшествования (рис. 2).

Наша работа структурирована следующим образом. В разделе 2 мы остановимся на математической постановке асимметрической обобщенной задачи коммивояжера. Далее, в разделе 3 мы напомним известную схему динамического программирования Хелда–Карпа (ДП), используемую для нахождения точного решения нашей задачи. В рассматриваемой нами постановке стоимости перемещения и посещения городов предполагаются зависящими от частичных маршрутов, но процедура ДП успешно справляется с этими затруднениями. В разделе 4 мы приведем эквивалентное описание этой процедуры в терминах нахождения кратчайших $s-t$ путей в соответствующем взвешенном ациклическом ориентированном графе (известном как *орграф состояний*), поскольку обоснование наших результатов, представленных в разделе 5, удобно проводить именно в этих терминах. Наконец, в разделе 6 мы остановимся на одном содержательном примере, иллюстрирующем проведенные рассуждения.

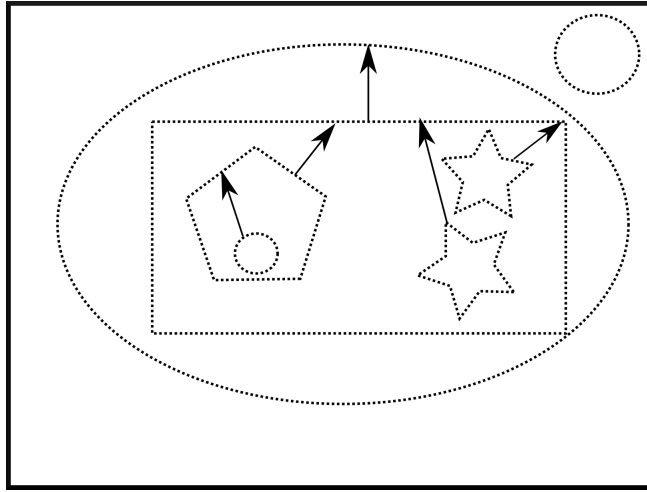


Рис. 2: Порядок резки задает ограничения предшествования

2 Постановка задачи

Рассмотрим постановку расширенной асимметрической обобщенной задачи коммивояжера (AGTSP) (рис. 3). Пусть заданы конечные дизъюнктные множества M_1, \dots, M_n , называемые *мегаполисами*, и стартовая точка x_0 , не принадлежащая ни одному из них. Без ограничения общности, полагаем множества M_j равномоными и определяемыми равенствами

$$M_j = \{g_{j1}, \dots, g_{jp}\}.$$

Вместе с транспортными издержками $c(g_{l\sigma}, g_{j\tau})$ для произвольных индексов

$$j \neq l \in \mathbb{N}_n = \{1, \dots, n\} \text{ и } \sigma, \tau \in \mathbb{N}_p,$$

заданы стоимости $\hat{c}(x_0, g_{j\tau})$ и $\check{c}(g_{j\tau}, x_0)$ перемещения из точки x_0 в каждую из точек $g_{j\tau}$ (и обратно). Далее, фиксируется стоимость посещения (внутренних работ) $c'(g_{j\tau})$ каждого из мегаполисов. Задача состоит в нахождении маршрута наименьшей стоимости, начинающегося и заканчивающегося в точке x_0 и посещающего все мегаполисы (в точности по одному разу).

С математической точки зрения требуется найти перестановку

$$\pi : \mathbb{N}_n \rightarrow \mathbb{N}_n,$$

задающую порядок посещения мегаполисов, называемую далее *маршрутом*, и конечную последовательность $g_{\pi(1)\tau(1)}, \dots, g_{\pi(n)\tau(n)}$, именуемую *трассой*, так, чтобы

$$\hat{c}(x_0, g_{\pi(1)\tau(1)}) + \sum_{i=1}^{n-1} (c'(g_{\pi(i)\tau(i)}) + c(g_{\pi(i)\tau(i)}, g_{\pi(i+1)\tau(i+1)})) + \check{c}(g_{\pi(n)\tau(n)}, x_0) \rightarrow \min. \quad (1)$$

Основные отличия между стандартной постановкой AGTSP и рассматриваемой нами в работе состоят в следующем:

- 1) для любых вершин $g_{l\sigma}$ and $g_{j\tau}$ стоимости перехода $c(g_{l\sigma}, g_{j\tau})$ и стоимость посещения мегаполиса $c'(g_{j\tau})$ зависят от выбранной трассы, соединяющей x_0 и вершину $g_{l\sigma}$;
- 2) на множестве мегаполисов определен один из двух типов ограничений предшествования (по аналогии с приведенными в работе [2] для классической TSP):

Тип I. Для натурального числа $k \leq n$, любая допустимая перестановка π удовлетворяет ограничению

$$\forall i, j \in \mathbb{N}_n \quad (j \geq i + k) \Rightarrow (\pi(i) < \pi(j)). \quad (2)$$

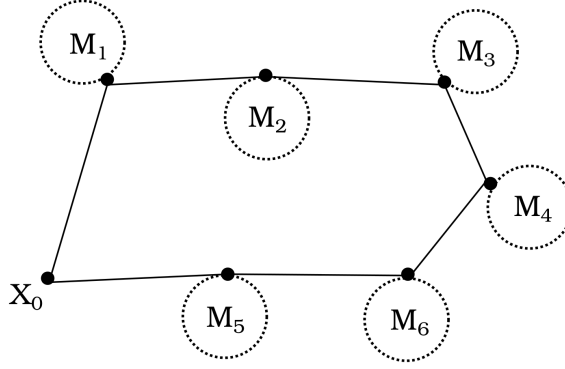


Рис. 3: Пример AGTSP для $n = 6$.

Тип II. Для любых натуральных значений $1 \leq k(1), \dots, k(n) \leq n$ и произвольной допустимой перестановки π справедливо соотношение

$$\forall i, j \in \mathbb{N}_n (j \geq i + k(i)) \Rightarrow (\pi(i) < \pi(j)). \quad (3)$$

3 Схема динамического программирования

Традиционный подход к решению задачи (1) восходит к классическим работам Беллмана [3] и Хелда и Карпа [11] и опирается на метод динамического программирования.

Пусть оптимальная трасса, выйдя из точки x_0 , посетив за первые $i - 1$ шагов мегаполисы с номерами из подмножества $J \subset \mathbb{N}_n$, в i -й позиции посещает мегаполис M_j в точке $g_{j\tau(i)} \in M_j$. Обозначим стоимость этой части трассы через $C(J, i, j, g_{j\tau(i)})$. Справедливы следующие рекуррентные соотношения

$$C(\emptyset, 1, j, g_{j\tau(1)}) = \hat{c}(x_0, g_{j\tau(1)}), \quad (4)$$

$$C(J, i, j, g_{j\tau(i)}) = \min_{l \in J} \min_{g_{l\tau(i-1)} \in M_l} \{C(J \setminus \{l\}, i-1, l, g_{l\tau(i-1)}) + c(g_{l\tau(i-1)}, g_{j\tau(i)}) + c'(g_{j\tau(i)})\}, \quad (5)$$

причем оптимальное значение задачи (1) может быть вычислено по формуле

$$C^* = \min_{j \in \mathbb{N}_n} (C(\mathbb{N}_n \setminus \{j\}, n, j, g_{j\tau(n)}) + \check{c}(g_{j\tau(n)}, x_0)). \quad (6)$$

Оптимальная трасса восстанавливается с помощью процедуры обратного поиска.

4 Граф состояний

Рекуррентная процедура (4)-(6) допускает эквивалентную формулировку в терминах теории графов. В самом деле, пользуясь известным подходом (см., например, [7]), сопоставим исходной задаче задачу поиска кратчайшего s - t пути в подходящем реберно-взвешенном $(n + 2)$ -дольном ориентированном графе $G^*[p] = (V^*[p], A^*[p], w^*[p])$. Через $V_i^*[p]$ обозначим подмножество вершин, составляющих i -ю долю, задаваемое равенствами

$$V_0^*[p] = \{s\}, \quad V_{n+1}^*[p] = \{t\}, \\ V_i^*[p] = \{(J, i, j, \tau) : j \in \mathbb{N}_n \setminus J, g_{j\tau} \in M_j, J \subset \mathbb{N}_n, |J| = i - 1\} \quad (i \in \mathbb{N}_n). \quad (7)$$

При этом вершины s и t соответствуют стартовой точке x_0 , а произвольная вершина (J, i, j, τ) (обычно именуемая *состоянием*) соответствует части трассы длины i , посещающей мегаполисы с номерами из множества $J \cup \{j\}$, последним среди которых — мегаполис M_j (в точке $g_{j\tau}$). Смежными в графе $G^*[p]$ являются лишь вершины соседних долей $V_i^*[p]$ и $V_{i+1}^*[p]$. Дуги $G^*[p]$ соединяют вершину s с произвольной

вершиной доли $V_1^*[p]$, произвольную вершину доли $V_n^*[p]$ с вершиной t , а также произвольные вершины (J, i, l, σ) и $(J', i + 1, j, \tau)$, удовлетворяющие условиям

$$|J| = i - 1, \quad J' = J \cup \{l\}, \quad j \notin J', \quad \sigma, \tau \in \mathbb{N}_p. \quad (8)$$

Определим множество дуг, соединяющих вершины, принадлежащие $V_i^*[p]$ и $V_{i+1}^*[p]$, как $A_{i,i+1}^*[p]$. Веса дуг определяются соотношениями:

$$w^*[p](s, (\emptyset, 1, j, \tau)) = \hat{c}(x_0, g_{j\tau}), \quad w^*[p]((\mathbb{N}_n \setminus \{j\}, n, j, \tau), t) = \check{c}(g_{j\tau}, x_0),$$

$$w^*[p]((J, i, l, \sigma), (J', i + 1, j, \tau)) = c(g_{l\sigma}, g_{j\tau}) + c'(g_{j\tau}).$$

Нетрудно убедиться в том, что множество допустимых трасс в задаче (1) взаимно однозначно отображается на множество s - t путей в графе $G^*[p]$, следовательно, кратчайшей трассе (оптимальному решению задачи (1)) соответствует s - t путь наименьшего веса. Более того, схема (4)-(6) эквивалентна модификации известного алгоритма Форда–Беллмана для поиска кратчайшего пути в бесконтурной сети (см., например, [7]).

К сожалению, для общего случая задачи AGTSP число дуг в орграфе $G^*[p]$ растет экспоненциально с ростом числа мегаполисов n , что влечет экспоненциальную же оценку трудоемкости метода динамического программирования для задачи (1). В самом деле, для произвольного $n \geq 2$

$$|V^*[p]| > |V_2^*[p] \cup \dots \cup V_n^*[p]| \geq pn2^{n-2}.$$

При этом входящая степень произвольной вершины $(J, m, j, \tau) \in V_m^*[p]$ при $m \geq 2$ удовлетворяет соотношению $\deg^-(J, m, j, \tau) = (m - 1)p \geq p$. Следовательно, $|A^*[p]| = \Omega(np^22^n)$.

Тем не менее, введение в условие задачи дополнительных ограничений, например, ограничений предшествования на множестве мегаполисов, может существенно снизить суммарную трудоемкость процедуры поиска оптимального решения (см., например, [19]). Ниже мы воспользуемся подходом, предложенным Баласом [2] для классической TSP, в рамках которого схема (4)-(6) обладает линейной по n трудоемкостью.

5 Оценки трудоемкости

Начнем с описания частного случая графа $G^*[p]$, соответствующего ограничениям предшествования типа I и типа II, описанным выше. Покажем, что структура графа $G^*[p]$ определяется структурой графа $G^*[1]$.

Лемма 1. Для любого $p > 1$:

$$V_i^*[p] = V_i^*[1] \times \mathbb{N}_p \quad (i \in \mathbb{N}_n) \quad (9)$$

$$A_{0,1}^*[p] = A_{0,1}^*[1] \times \mathbb{N}_p, \quad A_{n,n+1}^*[p] = A_{n,n+1}^*[1] \times \mathbb{N}_p \quad (10)$$

$$A_{i,i+1}^*[p] = A_{i,i+1}^*[1] \times \mathbb{N}_p^2 \quad (i \in \mathbb{N}_{n-1}) \quad (11)$$

В самом деле, задавшись произвольным $p > 1$, рассмотрим отображение $\Gamma: V^*[p] \rightarrow V^*[1]$, задаваемое соотношениями

$$\Gamma(s) = s, \quad \Gamma(t) = t, \quad \Gamma((J, i, j, g_{j\tau})) = (J, i, j).$$

Поскольку смежность вершин в графе $G^*[p]$ при произвольном p определяется соотношением (8), отображение Γ является гомоморфизмом. Более того, вершины $(J, i, l, g_{l\sigma})$ и $(J \cup \{l\}, i + 1, j, g_{j\tau})$ смежны тогда и только тогда, когда смежны вершины (J, i, l) и $(J \cup \{l\}, i + 1, j)$ в графе $G^*[1]$. По построению,

$$\Gamma^{-1}((J, i, j)) = \{(J, i, j, g_{j1}), \dots, (J, i, j, g_{jp})\},$$

что влечет справедливость соотношений (9)-(11). Лемма доказана.

Следствие 1. Для любого $p > 1$

$$|A^*[p]| \leq |A^*[1]| \cdot p^2.$$

В работе [2] была описана структура графа $G^*[1]$, определяющего динамическую процедуру для классической TSP с дополнительными ограничениями предшествования (2) и (3). Представим эти результаты в Теореме 1.

Теорема 1.

1. В случае ограничений предшествования (2)

$$|A^*[1]| = O(n \cdot k^2 2^{k-2}).$$

2. В случае ограничений (3)

$$|A^*[1]| = O\left(\sum_{i=1}^n k^*(i)(k^*(i) + 1)2^{k^*(i)-2}\right)$$

для

$$k^*(i) = \max\{k(j) : i - k(j) + 1 \leq j \leq i\}.$$

Наши основные результаты для оценки вычислительной сложности следуют из Леммы 1 и Теоремы 1.

Теорема 2. Пусть для рассматриваемой задачи AGTSP имеют место ограничения предшествования (2). Тогда схема динамического программирования (4)-(6) позволяет получить оптимальное решение (для данной задачи) за время

$$O(n \cdot p^2 k^2 2^{k-2}). \quad (12)$$

Теорема 3. Если любая постановка задачи AGTSP удовлетворяет следующим ограничениям предшествования (3), тогда схема (4)-(6) будет выполнена за время

$$O\left(p^2 \sum_{i=1}^n k^*(i)(k^*(i) + 1)2^{k^*(i)-2}\right). \quad (13)$$

В теоремах 2 и 3 утверждается, что в случае дополнительных ограничений предшествования (2) или (3), решение задачи AGTSP может быть найдено эффективно. В самом деле, легко видеть, что верхняя оценка (12) (в случае оценки (13) аналогично) является линейной по n для любых фиксированных k и p и остается полиномиальной для $p = O(\text{poly}(n))$ и $k = O(\log(n))$. Таким образом, процедура динамического программирования в обоих случаях позволяет найти оптимальное решение за время, линейно зависящее от числа мегаполисов n .

6 Содержательный пример

Обсудим применимость ограничений предшествования. На первый взгляд, ограничения (2)-(3) могут показаться излишне строгими. Тем не менее, даже ограничение (2), при k от 1 до n , охватывает все возможные случаи задачи AGTSP.

Рассмотрим следующий содержательный пример. Нужно составить план эвакуации из многоэтажного здания при пожаре. Здание состоит из m этажей. На t -м этаже находится q_t комнат. Каждая комната (M_i) имеет несколько дверей (рис. 4). Команда спасателей может начать выполнение своего задания, высадившись на любом этаже (стоимостью доставки на этаж можно пренебречь). После выполнения своего задания команда может быть эвакуирована также с любого этажа. Однако основным ограничением является возможность перехода между этажами только с помощью лифтов: стоимость такого перемещения намного выше, чем стоимость перемещения по этажу. Сформулировав данную задачу математически, мы получим следующие ограничения предшествования. Для комнаты M_i определим

$$k(i) = \begin{cases} q_1 + 1 - i, & \text{при } 1 \leq i \leq q_1, \\ q_1 + q_2 + 1 - i, & \text{при } q_1 + 1 \leq i \leq q_1 + q_2, \\ \dots & \\ \sum_{l=1}^{m-1} q_l + 1 - i, & \text{при } \sum_{l=1}^{m-2} q_l + 1 \leq i \leq \sum_{l=1}^{m-1} q_l. \end{cases}$$

Эти ограничения означают, что здание должно быть эвакуировано или сверху вниз, или наоборот. Следовательно, ограничения предшествования (3) являются вполне естественными для данного класса задач.

7 Заключение

В статье предложена процедура динамического программирования для нахождения оптимального решения задачи AGTSP. Для двух типов ограничений предшествования показано, что процедура является эффективной. В этом случае трудоемкость такой процедуры линейна по n для любых фиксированных k и p и полиномиальна при $k = O(\log n)$ и $p = O(\text{poly}(n))$.

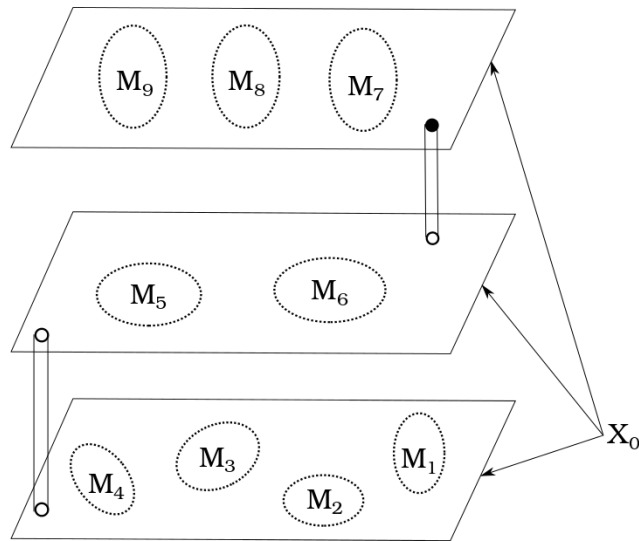


Рис. 4: Иллюстрация эвакуационного плана при пожаре для $m = 3$, $q_1 = 4$, $q_2 = 2$, $q_3 = 3$

Благодарности

Исследования поддержаны Российским научным фондом, грант № 14-11-00109.

Список литературы

- [1] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45, 1998.
- [2] E. Balas. New classes of efficiently solvable generalized traveling salesman problems. *Annals of Operations Research*, 86(0):529–558, 1999.
- [3] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9(1):61–63, 1962.
- [4] Boris Bontoux, Christian Artigues, and Dominique Feillet. A memetic algorithm with a large neighborhood crossover operator for the generalized traveling salesman problem. *Computers & Operations Research*, 37(11):1844 – 1852, 2010.
- [5] A.A. Chentsov and A.G. Chentsov. Dynamic programming method in the generalized traveling salesman problem: The influence of inexact calculations. *Mathematical and Computer Modelling*, 33(8–9):801 – 819, 2001.
- [6] N Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. In *Symposium on New Directions and Recent Results in Algorithms and Complexity*, page 441, 1975.
- [7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT, 3ed. edition, 2009.
- [8] Vladimir G. Deineko, Bettina Klinz, Alexander Tiskin, and Gerhard J. Woeginger. Four-point conditions for the tsp: The complete complexity classification. *Discrete Optimization*, 14:147–159, 2014.
- [9] Emanuele Garone, Jean-Francois Determe, and Roberto Naldi. Generalized traveling salesman problem for carrier-vehicle systems. *Journal of Guidance Control and Dynamics*, 37:766–774, 2014.
- [10] Gregory Gutin and Daniel Karapetyan. A memetic algorithm for the generalized traveling salesman problem. 9(1):47–60, 2010.

- [11] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. In *Proceedings of the 1961 16th ACM National Meeting*, ACM '61, pages 71.201–71.204, New York, NY, USA, 1961. ACM.
- [12] Keld Helsgaun. Solving the equality generalized traveling salesman problem using the lin–kernighan–helsgaun algorithm. *Mathematical Programming Computation*, pages 1–19, 2015.
- [13] Kan Jun-man and Zhang Yi. Application of an improved ant colony optimization on generalized traveling salesman problem. *Energy Procedia*, 17, Part A(0):319 – 325, 2012.
- [14] D. Karapetyan and G. Gutin. Lin–kernighan heuristic adaptations for the generalized traveling salesman problem. *European Journal of Operational Research*, 208(3):221 – 232, 2011.
- [15] M.Yu. Khachai and E.D. Neznakhina. A polynomial-time approximation scheme for the euclidean problem on a cycle cover of a graph. *Proceedings of the Steklov Institute of Mathematics*, 289(1):111–125, 2015.
- [16] M. Khachay and K. Neznakhina. Approximability of the minimum-weight k-size cycle cover problem. *Journal of Global Optimization*, 2015.
- [17] Gilbert Laporte, Hélène Mercure, and Yves Nobert. Generalized travelling salesman problem through n sets of nodes: the asymmetrical case. *Discrete Applied Mathematics*, 18(2):185 – 197, 1987.
- [18] A.A. Petunin, A.G. Chentsov, and P.A. Chentsov. About a routing problem of the tool motion on sheet cutting. *Model. Anal. Inform. Sist.*, 22, 2015.
- [19] George Steiner. On the complexity of dynamic programming for sequencing problems with precedence constraints. *Annals of Operations Research*, 26(1):103–123, 1990.

Efficient solution method for Generalized Travelling Salesman Problem with precedence constraints of a special type

Alexander G. Chentsov, Daniel M. Khachay

Krasovskii Institute of Mathematics and Mechanics (Yekaterinburg, Russia)

Ural Federal University (Yekaterinburg, Russia)

Keywords: Asymmetric Generalized Traveling Salesman Problem (AGTSP), NP-hard problem, dynamic programming.

We consider the combinatorial optimization problem of visiting clusters of a fixed number of nodes (cities) under the special type of precedence constraints. This problem is a kind of the Generalized Traveling Salesman Problem (GTSP). To find an optimal solution of the problem, we propose a dynamic programming based on algorithm extending the well known Held and Karp technique. In terms of special type of precedence constraints, we describe subclasses of the problem, with polynomial (or even linear) in n upper bounds of time complexity.