Computer tools in education, 2023
№ 4: 30–40
http://cte.eltech.ru
doi:10.32603/2071-2340-2023-4-30-40

VERSIONS OF THE "ONION HUSK" ALGORITHM IN THE PSEUDO-GEOMETRIC TRAVELING SALESMAN PROBLEM WITH A SMALL VARIANCE

Melnikov B. F.[1,2], Doct. Sci., Professor, ✉ bormel@mail.ru, orcid.org/0000-0002-6765-6800
Melnikova E. A.[3], Cand. Sc., Associated Professor, ya.e.melnikova@yandex.ru, orcid.org/0000-0003-1997-1846

[1]Shenzhen MSU – BIT University, No. 1, International University Park Road, Dayun New Town, Longgang District, Shenzhen, PRC, 517182, Guangdong Province, Shenzhen, China
[2]Center for Information Technologies and Systems of Executive Authorities, Presnensky Val street, 19-1, 123557, Moscow, Russia
[3]Russian State Social University, 4, build. 1, Wilhelm Pieck street, 129226, Moscow, Russia

Abstract

We continue to consider the pseudo-geometric traveling salesman problem. Specifically, we are considering several auxiliary algorithms needed to implement different versions of the "onion husk" algorithm. We have not found in the literature an accurate description of specific versions of algorithms for the geometric version (however, this is not necessary, since it is necessary to implement the original versions for the pseudo-geometric version), so we start with the geometric version.

Random generation of data for computational experiments corresponded to the problem being solved. For each of the some dimensional variants, some computational experiments were conducted with randomly generated input data. The following characteristics were calculated: the average number of resulting contours for the geometric variant; the ratio of the solution with contours to the optimal solution; the ratio of the solution of the pseudo-geometric version corresponding to the order of points of the geometric version to the geometric solution. The obtained results of computational experiments in general approximately correspond to the expected values.

Keywords: *optimization problems, traveling salesman problem, heuristic algorithms, "onion husk" algorithm, real-time algorithms, C++.*

Citation: B. F. Melnikov and E. A. Melnikova, "Versions of the "Onion Husk" Algorithm in the Pseudo-Geometric Traveling Salesman Problem with Small Variance," *Computer tools in education*, no. 4, pp. 30–40, 2023 (in Russian); doi: 10.32603/2071-2340-2023-4-30-40

1. INTRODUCTION

The sources of our topic are some different works presented in the collection of papers [1]. Directly the subject of this paper continues our works [2–5], as well as, partially, [6, 7]. We continue to consider the pseudo-geometric traveling salesman problem. Specifically, we are considering some auxiliary algorithms needed to implement different versions of the "onion husk" algorithm.

We have not found in the literature an accurate description of specific versions of algorithms for the geometric version. Often in publications, instead of describing the algorithm, pictures are given, like those that we gave in the previous paper; we repeat them further, see Fig. 1.

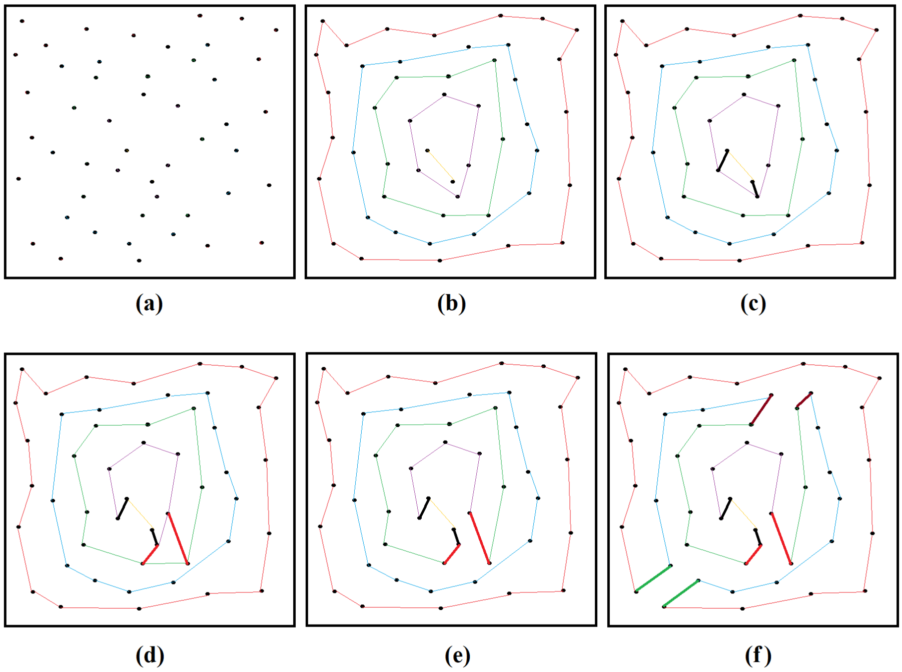

Figure 1. The general scheme of the “onion husk’ algorithm

It is clear that it is not difficult to describe a detailed algorithm based on such pictures, but our goal, as can be seen from the title of the paper, was to process a more complex case. Thus, we are going to implement the original versions for the *pseudo-geometric* version.

We should immediately note that according to the subject of the paper, we quite often use the concept of averaging for the obtained random variables. Therefore, the specific averaging algorithms that we use are of interest. A certain number of computational experiments were carried out with various randomly generated data (at least 9 times for each calculated characteristic), then the smallest and the largest obtained values of these characteristics were discarded, and the arithmetic mean was calculated for the remaining values.

Here are the contents of the paper by sections. In *Section 2*, we consider the general description of works related to the pseudo-geometric version of the traveling salesman problem. In *Section 3*, we give the brief description of the algorithms for generating input data. In *Section 4*, the descriptions of the used algorithms and some results of computational experiments is considered. *Section 5* is the conclusion: we give some possible directions of further work on the subject under consideration.

2. THE GENERAL DESCRIPTION OF WORKS RELATED TO THE PSEUDO-GEOMETRIC VERSION OF THE TRAVELING SALESMAN PROBLEM

We start with the geometric version, and therefore it is important to repeat the information from the previous paper [2] about how exactly the input data is generated. For each pseudo-geometric variant of the problem necessary for further work, we first generate a corresponding geometric variant with a given number of points; we note in advance that such a geometric variant can be used repeatedly for further actions. At the same time, the geometric variant is obtained by randomly throwing a given number of points into a unit square: both coordinates of each point are obtained by applying a uniform distribution on the segment [0, 1].

Next, each value of the resulting matrix [1] is multiplied by a next random variable; each of these random variables is obtained by applying a normal distribution with a mathematical expectation μ equal to 1 and some given variance D (or standard deviation σ). The given figure (Fig. 1, which is the extension to the figure from the previous paper [2]), shows the probability density function and the possible values of some of its characteristics for the value $\sigma = 0.2$ used in [6].

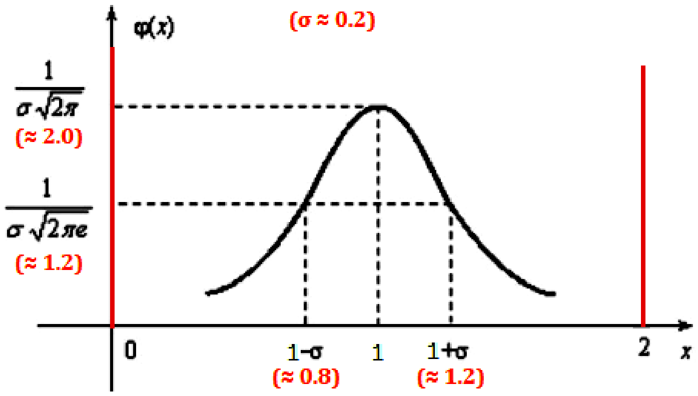

Figure 2. A graph of normal distribution and some values of its parameters with $\sigma \approx 0.2$

To the latter, such explanations are necessary.

- Firstly, in the case of symmetric variants of the traveling salesman problem, we apply all the described actions to the elements above the main diagonal only (otherwise, to all elements except diagonal ones); in this paper we apply exactly such variants, i.e., the symmetric traveling salesman problem.
- Secondly, we are not interested in non-positive values. Therefore, we shall skip their possible appearances during generation (which is possible with high probabilities for large values). For symmetry, we also skip the possible values of random variables equal to or greater than 2.
- Thirdly, in this paper we used rather small variances ($D \leqslant 0.0016$, i.e., $\sigma \leqslant 0.04$); we expect to return to large values in the following publications.
- Fourth, of particular interest, oddly enough, is the algorithm for generating independent identically distributed random variables: despite the fact that this is a student's task for junior courses, even Habr [2] does not describe successful algorithms for such generation, and we have to use:
  — either complex algorithms using generalized variants of random variable distribution functions (and the *programmer's* work time is greatly increased),
  — or any library (and the *program's* work time is greatly increased [3]).

[1] Note such an obvious fact, which, however, clearly does not follow from the text given here. At the input of the algorithms for solving the traveling salesman problem, it is precisely such matrices that are served, and not the initial locations of the points.

[2] https://habr.com/ru/all/.

[3] For example, when using Python to write the programs we discussed in the previous paper [2], the total running time increased several times compared to C++ (sometimes by almost 10 times).

At the same time, such additional explanations are even more interesting (although they have little to do with the material of this paper), partially already considered in [2]. With very small values of variance, we get a version of the problem that practically coincides with the geometric version (for $D = 0$ this fact is obvious). Conversely, for very large values of variance, we can assume that a so-called random variant of the problem is obtained [4].

However, in practice, to solve the problem in the general formulation, we have a distance matrix only, and the following question arises: how to distinguish the generation options, i.e., how to find out which algorithm was used to generate the particular case of the problem in question? Of course, some statistical methods are possible for such a situation when a large set of input data is given, i.e., many special cases of the problem. But, firstly, we have not yet dealt with such statements at all (although *this would be a very interesting task*), and, secondly, we can always assume that the given variant was randomly generated for a pseudo-geometric formulation of the problem with any variance (other than 0 and ∞). At the same time, an important goal of this paper (and its possible sequels) is that the possible knowledge of the variance value (or σ) will allow us to describe much more successful algorithms for solving the general version of the traveling salesman problem.

3. THE BRIEF DESCRIPTION OF THE ALGORITHMS FOR GENERATING INPUT DATA

Random generation of data for computational experiments corresponded to the problem being solved. It has already been noted above that we have not found in the literature successful algorithms for generating independent identically distributed random variables with a normal distribution law. We also repeat that we did not find them on Habr [5]. At the same time, we limit ourselves to small values of variance (specifically, $D \approx 0.02$, i.e., $\sigma \approx 0.14$), and for such values, we can take advantage of the fact [8], that most auxiliary statistical characteristics practically do not distinguish between:

- — normally distributed random variables,
- — and sums of a sufficiently large number of terms, each of which is a uniformly distributed random variable with the same distribution law.

The above is fully consistent with the fact that we are currently limited to a variance of no more than 0.02, in contrast to [6] where a much larger variance (0.2) was chosen; however, the last value leads to more complicated algorithms. It is also important to add that using large variances in [6], we considered there the very first versions of all algorithms only. Using of the first versions of all algorithms consistent with the fact that only the most preliminary comparisons for very different approaches; it is understandable from the subject of that paper.

Thus, first it seems successful to consider such an *example*. Let us sum up 50 uniform independent identically distributed random variables with a *uniform* distribution law on the segment $[0, 1]$. For the resulting random variable (sum), we get $M = 25$ and $D = 50 \cdot (1/12) = 50/12$ (since $1/12$ is D-value for the uniform random variable on $[0, 1]$). Next, we divide the resulting random variable by 25 and we get a “practically normal” random variable with $M = 1$, this is what we need for our tasks. To find D, we take $1/25$ out of the bracket and square this value, we get the coefficient of $1/625$, which we multiply by the previously found value of $50/12$; as a result, we get the variance

$$\frac{50}{12 \cdot 25 \cdot 25} = \frac{1}{150}.$$

[4] The latter, in our opinion, is best solved using the method of branch and bound, which was the main subject of [2].

[5] See, for example, https://habr.com/ru/post/263993/ and some other Habr’s pages.

In general, we shall get the necessary variance in a similar way. We consider the sum of N such uniform independent identically distributed random variables on $[0,1]$; we get a random variable with $M = N/2$ and $D = N/12$. Next, we divide the resulting random variable by $N/2$ and get a "practically normal" random variable with $M = 1$. In the same way, we take $1\big/(N/2)$ out of the bracket and square this value, we get the coefficient $1\big/(N^2/4)$, which we multiply by the previously found value $N/12$. As a result, we get

$$\frac{4\cdot N}{12\cdot N\cdot N} = \frac{1}{3\cdot N}.$$

But in reality we have to solve a trivial "inverse problem", i.e., to get the desired variance. Note that we can hardly be interested in the exact values of the variance; we are interested in approximate values. To obtain a given variance of approximately D, it is necessary that $1\big/(3\cdot N)$ is approximately equal to D, so the number of terms of N should be approximately equal to

$$N \approx \frac{1}{3\cdot D}.$$

Within $D \leqslant 0.02$, this is done easily, and the calculations give the following table (Tab. 1):

Table 1. The required number of terms for the required variance value for $D \leqslant 0.2$

σ	D	N
≈ 0.14	≈ 0.02	17
≈ 0.12	≈ 0.0144	23
≈ 0.10	≈ 0.01	33
≈ 0.08	≈ 0.0064	52
≈ 0.06	≈ 0.0036	93
≈ 0.04	≈ 0.0016	208
≈ 0.02	≈ 0.0004	833
0	0	—

It is clear that in order to obtain data corresponding to the geometric traveling salesman problem (which happens when $D = 0$), additional calculations of normally distributed random variables are not required.

Note also that a large variance (to say, $D \geqslant 0.1$) is hardly interesting for the complex of problems under consideration; and therefore only *fast* algorithms for obtaining random variables with a variance within $D \in [0.02, 0.1]$ can present a small complexity. However, we have not yet practically started to consider the problems obtained by applying such variance variants.

4. THE DESCRIPTIONS OF THE USED ALGORITHMS AND SOME RESULTS OF COMPUTATIONAL EXPERIMENTS

We can say that in this section we are considering only the description of computational experiments. However, of course, each of the implemented procedures should find application in some other, more complex projects; the general description of some of these projects was briefly given above (or follows from the text above). All the data based on the random generation are calculated of input data, also briefly described above.

First of all, we build the contours of the "onion husk" algorithm. For the software implementation of subsequent projects, it is important for us to know a random variable depending on random generation, which is the average number of resulting contours. At the same time, we do not consider the possibility of *improving* the algorithms for constructing these contours (since it is easy to give an example when using “intersecting contours” for the entire traveling salesman problem gives better results); the main reason for not considering these issues is as follows: as we said before, we mainly focus on the pseudo-geometric version of generating special cases of the problem, and this version is unlikely to be needed such scrupulousness for the initial task of constructing contours.

```
0     001  564   1
1     193  809   2
2     585  480   3
3     350  896   1
4     823  747   1
5     174  859   1
6     710  514   2
7     304  015   1
8     091  364   2
9     147  166   2
10    988  446   1
11    119  005   1
12    009  378   1
13    532  571   3
14    602  607   2
15    166  663   3
16    451  352   3
17    057  608   2
18    783  803   1
19    520  302   2
0    ++++  311  590  482  842  342  711  627  219  424  994  571  186  530  602  193  497  071  818  581
1     311 ++++  511  180  633  054  595  801  456  644  874  807  469  413  455  148  524  243  590  603
2     590  511 ++++  478  357  559  130  543  507  539  405  665  585  106  128  457  185  543  379  189
3     482  180  478 ++++  496  180  525  882  591  758  781  921  620  372  383  297  553  411  443  618
4     842  633  357  496 ++++  658  259  897  825  891  344 1023  894  340  261  662  542  778  069  538
5     342  054  559  180  658 ++++  638  854  501  694  913  856  509  459  496  196  577  277  612  656
6     711  595  130  525  259  638 ++++  643  637  662  286  780  715  188  143  564  306  660  298  285
7     627  801  543  882  897  854  643 ++++  409  218  809  185  468  601  663  663  368  642  922  359
8     219  456  507  591  825  501  637  409 ++++  206  901  361  084  486  565  308  360  246  819  433
9     424  644  539  758  891  694  662  218  206 ++++  886  164  253  559  633  497  356  451  900  397
10    994  874  405  781  344  913  286  809  901  886 ++++  975  982  474  419  851  546  945  412  490
11    571  807  665  921 1023  856  780  185  361  164  975 ++++  389  701  772  660  480  606 1038  499
12    186  469  585  620  894  509  715  468  084  253  982  389 ++++  557  636  326  443  235  883  517
13    530  413  106  372  340  459  188  601  486  559  474  701  557 ++++  079  377  234  476  342  269
14    602  455  128  383  261  496  143  663  565  633  419  772  636  079 ++++  439  296  545  267  316
15    193  148  457  297  662  196  564  663  308  497  851  660  326  377  439 ++++  421  122  633  505
16    497  524  185  553  542  577  306  368  360  356  546  480  443  234  296  421 ++++  469  560  085
17    071  243  543  411  778  277  660  642  246  451  945  606  235  476  545  122  469 ++++  752  555
18    818  590  379  443  069  612  298  922  819  900  412 1038  883  342  267  633  560  752 ++++  566
19    581  603  189  618  538  656  285  359  433  397  490  499  517  269  316  505  085  555  566 ++++

konturov: 3
```

Figure 3. An example of issuing the first stage of data processing

Fig. 3 shows the output of the first of the programs related to the project, it is given for processing small-scale data. First the coordinates of the points are printed (all data is given as 3 digits after the decimal point, so each of the coordinates is given as an integer between 0 and 1000), to the right of the coordinates is the calculated contour number (they are numbered starting from 1, from external to internal), below is the distance matrix for the future solution of the general traveling salesman tasks (ranging from 0 to 1414), and the last output is the total number of counted contours.

We built each regular contour in a trivial way, and we did not strive for the effectiveness of the contour construction function. Exactly, we selected a set of segments from points not yet included in the contours, for each of which all the remaining unconnected points lie on the same side of the straight line on which the segment in question lies. For small dimensions (up to 500, as in our situation), such an algorithm is quite acceptable, but for large dimensions (to say, 5000 and more) it is hardly acceptable. Moreover, the construction of a contour is quite an interesting problem, in our opinion, it is not fully described in the literature, and the algorithms already described can be improved; we expect to return to this problem in some future publications.

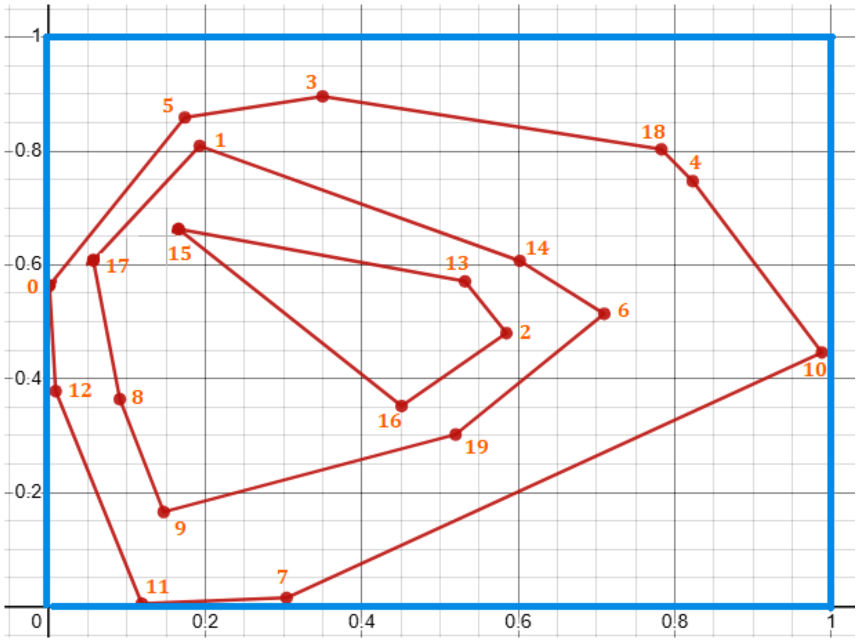

Figure 4. The location of the points to the above output example and the corresponding contours

Fig. 4 shows the location of the points to the above output example and the corresponding contours. The possible construction of a solution of the traveling salesman problem based on the contours found is shown in the following Fig. 5: between each neighboring contours, we select the two closest pairs of points [6]; the added and the removed edges are shown in green. At the same time, the algorithms for such a choice are simple, but some difficulties may arise when choosing any point of such a pair for two contours: the external one and the internal one for it. The detailed solution we have applied to the described small problem is hardly of interest. Moreover, for further calculations, we did not use the found solution to the traveling salesman problem, but the sum of the lengths of the constructed contours; we explain this simplification by the fact that it is the characteristics of the contours that are one of the main subjects of this paper.

We compared the total length of the contours (which differs little from the current solution obtained based on the contours) with the optimal solution calculated according to [2]. However, as we have already noted, we did not use large dimensions here.

We do not provide relevant illustrations for the description of further calculations (they would probably be uninformative), so we shall describe them in more detail. Based on the generated variants of the geometric version, according to the explanations given above, we generate pseudo-geometric variants using two values of the standard deviation: $\sigma = 0.04$ and $\sigma = 0.08$. For each generated version of the problem, we "adjust to the answer", i.e., to the solution of the traveling salesman problem; exactly, when generating the solution, we use *the same sequences* that were found for the original geometric variant; in both cases (geometric and pseudo-geometric ones), the sum of the contour lengths is considered for some simplification. At the same time, we fixed the result of the ratio of the solution of the pseudo-geometric variant obtained in this way to the solution of the corresponding geometric variant.

[6] We shall not discuss the optimality of the solution constructed in this way. A little more about this, see later in this paper.

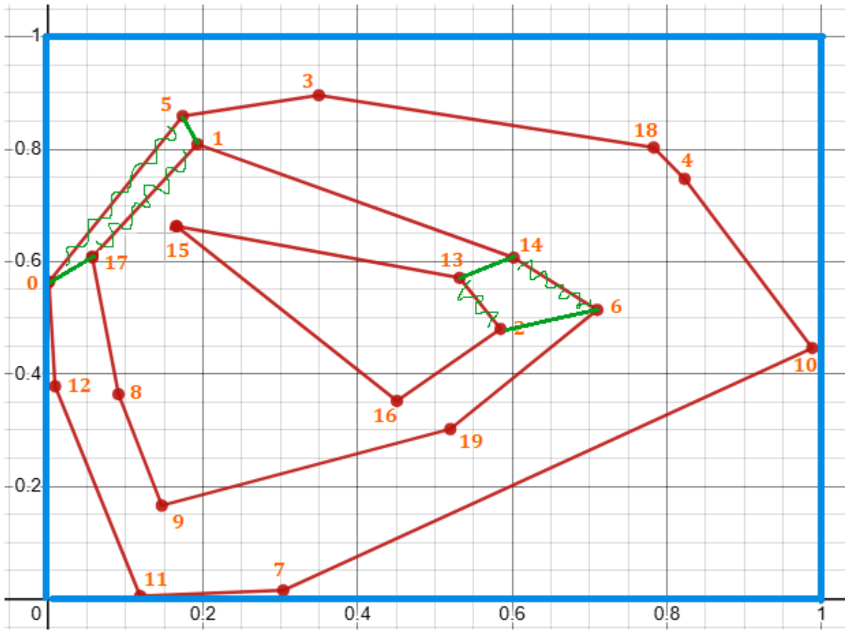

Figure 5. The possible construction of a solution of the traveling salesman problem

Let us describe the structure of Table 2 given below. The dimension is indicated in the table of calculation results for rows. As we already said, all other results (in each cell of the table) are averaged as follows: a certain number of computational experiments were carried out with randomly generated data, then the smallest and largest values obtained were discarded, the arithmetic mean was calculated for the remaining values.

Table 2. The main results of the computational experiments

	K	R	0.04	0.08
20	3.71	–0.73	2.09	4.17
45	6.17	–1.02	3.11	6.82
79	8.71	–2.11	4.87	9.13
199	16.71	—	9.22	20.01
499	30.29	—	34.34	70.34

- The “K” column shows the average number of resulting contours for the geometric variant.
- The “R” column is the ratio of the solution with contours to the optimal solution. For this column and its cells, the following additional comments are needed.
  - — The dimension is considered no more than 79, because exact solutions are required for the table, they are calculated using the method of branch and bound (the variant of it that was described in previous publications, firstly in [2]), because of this, restrictions occur.
  - — We have already noted that instead of the value of the solution, we use the sum of the contour lengths. The negative values are associated with this, i.e., instead of deterioration of the values, their improvement occurs [7].

[7] At the same time, preliminary results on very large dimensions do give deterioration: the number of contours will increase at a not very high rate. However, since we have not built exact solutions for large dimensions, we really can only talk about preliminary results so far.

- The last columns are marked with the values of the standard deviation for the pseudo-geometric version, 0.04 and 0.08. In them, we place the calculated ratio of the "adjusted to the answer" solution of the pseudo-geometric version (i.e., the solution corresponding to the order of points of the geometric version) to the geometric solution. It is clear that in real algorithms for solving the pseudo-geometric version of the traveling salesman problem, we cannot get such a value; however, we can get it by knowing the generation algorithm, and these values are interesting for describing the solution algorithms.
  At the same time, in the last two columns, the result is given as a percentage: for example, the value 4.17 corresponds to an increase of 1.0417 times.

We think, that the obtained results of computational experiments in general approximately correspond to the expected values.

5. CONCLUSION

Thus, the motivation for the implementation of the algorithms of this paper was as follows.

- The main part of the motivation (global theme) was that we intend to complete a long-started cycle of work related not to the geometric version (for which the algorithms described here were previously developed), but to the pseudo-geometric one.
- Secondly, in a very large number of works ([9, 10] etc.), similar variants of the general description of the "onion husk" algorithms were given — but we did not find a sufficiently detailed description of specific algorithms.
- Thirdly, in the work of [6], we began to implement the idea of applying such algorithms to the source data given in the form of a pseudogeometric version. In the mentioned work, a comparison of all the algorithms of interest to us was given — however, each of them was implemented without any attempts to improve the quality of implementation.

As possible extensions of the topic under consideration here, we note the following. Possible knowledge of the variance value (or sigma) will allow us to describe much more successful algorithms for solving the general version of the traveling salesman problem, i.e. if it is known that a particular case of the problem was obtained as a pseudo-geometric traveling salesman problem, then it is desirable to learn how to build a probability density for the possible variance for which this problem turned out. We will also continue to apply the geometric approach to the pseudogeometric version, as well as compare the results of this approach with other algorithms for solving the traveling salesman problem.

Acknowledgement. The work of the first author was partially supported by a grant from the scientific program of Chinese universities "Higher Education Stability Support Program" (section "Shenzhen 2022 — Science, Technology and Innovation Commission of Shenzhen Municipality") — 深圳市2022年高等院校定支持划助目.

References

1. G. Gutin and A. Punnen, eds., *The Traveling Salesman Problem and Its Variations*, Dordrecht Netherlands: Kluwer Academic Publishers, 2002.
2. B. Melnikov and E. Melnikova, "On the Classical Version of the Branch and Bound Method," *Computer tools in education*, no. 1, pp. 21–44, 2021 (in Russian); doi:10.32603/2071-2340-2021-1-21-45
3. B. Melnikov and E. Melnikova, "About the classical version of the branch and bound method," *Computer tools in education*, no. 2, pp. 41–58, 2022 (in Russian); doi:10.32603/2071-2340-2022-2-41-58
4. B. Melnikov, "On the object-oriented implementation of the branch and bound method for the traveling salesman problem. Part I," *Modern information technologies and IT education*, vol. 18, no. 2, pp. 287–299, 2022 (in Russian); doi:10.25559/SITITO.18.202202.287-299

5. B. Melnikov, “On the object-oriented implementation of the branch and bound method for the traveling salesman problem. Part II,” *Modern information technologies and IT education*, vol. 18, no. 3, pp. 644–654, 2022 (in Russian); doi:10.25559/SITITO.18.202203.644-654
6. S. Makarkin and B. Melnikov, “Geometric methods for solving the pseudo-geometric version of the traveling salesman problem,” *Stochastic optimization in computer science*, vol. 9, no. 2, pp. 54–72, 2013 (in Russian).
7. G. Bulynin, B. F. Melnikov, V. Y. Meshchanin, and Y. Y. Terentyeva, “Optimization problem, arising in the development of high-dimensional communication networks, and some heuristic methods for solving them,” *Informatization and communication*, no. 1, pp. 34–40, 2020 (in Russian); doi:10.34219/2078-8320-2020-11-1-34-40
8. M. Lagutin, *Visual mathematical statistics*, Moscow: BINOM — Laboratory of Knowledge, 2009 (in Russian).
9. D. Johnson and L. McGeoch, “Experimental analysis of heuristics for the STSP,” in *G. Gutin and A.Punnen, eds. The traveling salesman problem and its variations*, Dordrecht, Netherlands: Kluwer Academic Publishers, pp. 369–443, 2002.
10. T. Ralphs, M. Saltzman, and M. Wiecek, “An improved algorithm for solving biobjective integer programs,” *Annals of Operations Research*, vol. 147, no. 1, pp. 43–70, 2006; doi:10.1007/s10479-006-0058-z

Received 22-06-2023, the final version — 17-08-2023.

Boris Melnikov, Doctor of Sciences (Phys.-Math.), Professor in Shenzhen MSU-BIT University, Shenzhen, China; Chief Researcher in Center for Information Technologies and Systems of Executive Authorities, Moscow, Russia, ✉ bormel@mail.ru

Elena Melnikova, Associated Professor, Russian State Social University, Moscow, Russia, ya.e.melnikova@yandex.ru

---

Компьютерные инструменты в образовании, 2023
№ 4: 30–40
УДК: 004.421.2:519.178
http://cte.eltech.ru
doi:10.32603/2071-2340-2023-4-30-40

Версии алгоритма «луковой шелухи» для псевдогеометрической версии задачи коммивояжёра с малой дисперсией

Мельников Б. Ф.[1,2], доктор физ.-мат. наук, профессор, ✉ bormel@mail.ru, orcid.org/0000-0002-6765-6800
Мельникова Е. А.[3], канд. физ.-мат. наук, доцент, ya.e.melnikova@yandex.ru, orcid.org/0000-0003-1997-1846

[1]Совместный университет МГУ – ППИ, район Лунган, Даюньсиньчэн, ул. Гоцзидасюеюань, д. 1, 517182, Провинция Гуандун, Шэнчжэнь, Китай
[2]Центр информационных технологий и систем органов исполнительной власти, Пресненский Вал, д. 19, стр. 1, 123557, Москва, Россия
[3] Российский государственный социальный университет, ул. Вильгельма Пика, д. 4, стр. 1, 129226, Москва, Россия

Аннотация

Мы продолжаем рассматривать псевдогеометрическую задачу коммивояжера. В частности, мы рассматриваем несколько вспомогательных алгоритмов, необходимых для реализации различных версий алгоритма «луковой шелухи». Мы не нашли в литературе точного описания конкретных версий алгоритмов для геометрической версии (впрочем, в этом нет необходимости, поскольку исходные версии этих алгоритмов необходимо реализовать для псевдогеометрической версии), поэтому мы начинаем работу с геометрической версии.

Случайная генерация данных для вычислительных экспериментов соответствовала решаемой проблеме: для каждого из нескольких вариантов размерности задачи были проведены некоторые вычислительные эксперименты со случайно сгенерированными входными данными. Были рассчитаны следующие характеристики полученных результатов: среднее количество результирующих контуров для геометрического варианта; отношение решения с контурами к оптимальному решению; отношение решения псевдогеометрической версии, соответствующее порядку точек геометрической версии, к геометрическому решению.

Полученные результаты вычислительных экспериментов в целом приблизительно соответствуют ожидаемым значениям.

Ключевые слова: *оптимизационные проблемы, задача коммивояжера, эвристические алгоритмы, алгоритм «луковой шелухи», алгоритмы реального времени, Си++.*

Цитирование: Мельников Б. Ф., Мельникова Е. А. Версии алгоритма «луковой шелухи» для псевдогеометрической версии задачи коммивояжёра с малой дисперсией // Компьютерные инструменты в образовании. 2023. № 4. С. 30–40. doi:10.32603/2071-2340-2023-4-30-40

Поступила в редакцию 22.06.2023, окончательный вариант — 17.08.2023.

Мельников Борис Феликсович, доктор физико-математических наук, профессор Совместного университета МГУ-ППИ в Шэньчжэне, Китай; главный научный сотрудник Центра информационных технологий и систем органов исполнительной власти, Москва, ✉ bormel@mail.ru

Мельникова Елена Анатольевна, кандидат физико-математических наук, доцент Российского государственного социального университета, Москва, ya.e.melnikova@yandex.ru