Noname manuscript No.
(will be inserted by the editor)

Ultra-Rapid Direct Satellite-Selection Algorithm for Multi-GNSS

Cheng Chi · **Xingqun Zhan** · **Tong Wu** · **Xin Zhang**

Received: date / Accepted: date

Abstract Global navigation satellite systems (GNSS) provide many more satellites than ever before. However for applications extremely sensitive to power consumption, not all satellites can be incorporated into the measurement vector, either because of the sheer computation overload or for purpose of power saving. These applications include but are not limited to unmanned aerial system (UAS), flying cars, and asset tracking. Thus satellite selection methodology should be used to obtain subset satellites with good geometry. Recently, a downdate method proposed in Receiver Autonomous Integrity Monitoring (RAIM) can be used for reference in satellite-selection, although RAIM and GNSS positioning are quite different. In this paper, a DOP based ultra-rapid satellite-selection methodology, the direct satellite-selection (DS) method, is proposed according the downdate method. Furthermore, to compensate for the shortcomings of the DS method, a constrained direct satellite-selection (CDS) method is then proposed by adding error monitoring and restrictive conditions. The two algorithms are examined for precision performance and computational performance. Simulations show the DS method performs about 3 order of magnitude faster than the recursive method, which is the existing fastest DOP based algorithm, with 0.25 increase in DOP on average relevantly when excluding the number of satellites from 42 to 8. And the CDS method performs about 2 order of magnitude faster than the recursive method with only 0.15 increase in DOP even when excluding satellites from 42 to 6. Consequently, both the two methods have much lower computation time than all the existing DOP based algorithms, with very little reduction in precision. Comparatively, the DS method has lower computational load and the CDS method has higher precision. Thereby, the algorithms proposed in this paper successfully address the satellite-selection problem in two scenarios; the CDS method fits in fast satellite-selection and high precision situation; the DS method can be employed in some extremely speed demanding circumstances.

Keywords Direct satellite-selection · Ultra-rapid · Multi-constellation · GDOP · Recursive method

Xingqun Zhan
E-mail: xqzhan@sjtu.edu.cn

School of Aeronautics and Astronautics, Shanghai Jiao Tong University No.800, Dongchuan Road, 200240, Shanghai, China

1 Introduction

The development of multiple constellations provides a higher positioning performance by increasing the visible probability of satellite. It is common for a GNSS receiver observing a huge number of satellites in some ideal areas. Generally, more observed satellites lead to higher positioning accuracy. But satellite tracking and position calculations are time and resource consuming for GNSS receivers. As the number of observed satellites increasing, more tracking channels are needed, which means better processor and higher cost. Meanwhile, GNSS positioning precision depends more on satellite geometry, rather than number of satellites. Thus it is impossible for a civilian grade GNSS receiver to take all-in-view satellites into account. Satellite selection method should be taken to address this issue.

There are many methods for selecting a set of satellites to use for GNSS positioning solution. The basic principle is to minimize the dilution of precision (DOP). Some authors employ satellite measurement error into account, such as using protection level (VPL and HPL) as the judgment condition. While, no matter which principle is used, the ideas are similar and universal. Hence, we just apply DOP as the judgment condition in this paper.

Most of the researchers who use DOP as the judgment condition often use the geometric dilution of precision (GDOP). A simple method is the brute force, which enumerates all the permutations to find the optimal one, namely brute force. Obviously this method is the most time-consuming. There are several methods to optimize the GDOP based satellite selection. Some authors take efforts to optimize the global search process using heuristic algorithms. One such method is to reduce global search time by using genetic algorithms (GAs) [1]. Similarly, [2] uses a chaotic particle swarm optimization (CPSO) method to reduce global search. These methods can be more optimal than other optimization algorithms, but they are both more time-consuming than others. As to more efficient methods, inverse lemma is used to simplify the GDOP calculation [3, 4]. It uses recursive thinking to generate each n-subset from its immediate predecessor by deleting a single satellite for one time. These methods can be induced as the recursive method and is much time reducing.

The following shows the computational load of existing DOP based algorithms. Firstly is the brute force, which is the most time-consuming. Supposing that the observed satellite number is n, and the desired number is k, the brute force method should calculate the GDOP, which contains a huge matrix calculation, for C_n^k times. If set n as 30 (It is a general situation in GPS-GLONASS-BDS system), and set k as 10, the calculation complex will be 30045015 times. It is hard for a civilian grade GNSS receiver operating this algorithm in a short time. The CPSO method given by [2] reduced the calculation time to about 37.5% of the brute force method, but it is still time-consuming. The recursive method proposed by [4] operated a time-saving performance. It reduced the calculation times to $\sum_{i=k}^{n} i$. For $n = 30$ and $k = 10$, the calculation time is 410, which is much smaller than the above method. But the computational time will grow quadratically when the number of observation increases.

Besides, a number of authors employ alternative performance measures beyond DOP [5]. A kind of them is the volume of the polyhedron method [6, 7], which calculate the volume of the polyhedron formed by satellites and the user, or employ elevation angle and azimuth angle to divide satellites into several block and allocate with some logic [8, 9]. While, the polyhedron based algorithm can reach sub-optimal but time-consuming, and the latter performs faster than the recursive method but failed in precision.

Consequently, existing satellite-selection algorithms either have unsatisfactory time performance, or failed in precision. Currently, [10] provided a downdate method in Autonomous Integrity Monitoring (RAIM) for satellite based augmentation systems (SBAS), with protection level. It can allow us to directly sort the satellite from the all-in-view matrix, rather than calculating each subset matrix. To address the satellite-selection problem in low computational load demanding situation, we apply the downdate method to the geometric condition, namely direct satellite-selection method, using GDOP as the basis for selecting. Then we propose a constrained direct satellite-selection method to increase accuracy and enlarge usage scope. Comparisons with other algorithms are also given in this paper. Besides, this paper will discuss how many satellites should be deleted to get appropriate GDOP value in a global sight.

2 Direct satellite-selection method

Since the downdate method proposed by [10] was proposed to select subset satellites with protection level, it is not suitable for the multi-GNSS DOP based satellite-selection condition. Formulas need to be redefined.

For a single constellation navigation algorithm, the geometry matrix $\boldsymbol{G}$ contains four columns. But in multi-constellation system, the most existing constellations are not synchronized with each other, the geometry matrix is defined as follows [11] (supposing there are L unsynchronized constellations).

$$\boldsymbol{G} = \begin{pmatrix} -e_{rx,1}^{1} & -e_{ry,1}^{1} & -e_{rz,1}^{1} & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ -e_{rx,1}^{m_1} & -e_{ry,1}^{m_1} & -e_{rz,1}^{m_1} & 1 & 0 & 0 & \cdots & 0 \\ -e_{rx,2}^{1} & -e_{ry,2}^{1} & -e_{rz,2}^{1} & 0 & 1 & 0 & \cdots & 0 \\ -e_{rx,2}^{2} & -e_{ry,2}^{2} & -e_{rz,2}^{2} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ -e_{rx,L}^{m_L} & -e_{ry,L}^{m_L} & -e_{rz,L}^{m_L} & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{1}$$

Where the second of the two subscripts is the constellation number, from 1 to L; and the superscript is the satellite number within the k^{th} constellation, from 1 to m_k. And the $GDOP$ is formed by the $\boldsymbol{G}$ matrix.

$$GDOP = \sqrt{\text{trace}\left[\left(\boldsymbol{G}^T \cdot \boldsymbol{G}\right)^{-1}\right]} \tag{2}$$

Readers should notice that the GDOP cannot directly represent the positioning error. The GDOP is just a conversion from satellite measurement error to positioning error. It is more suitable to employ protection level to quantify the positioning error [12]. While, in this paper, we only use GDOP to get the optimal geometry.

The weight coefficient matrix H is given below

$$\boldsymbol{H} = \left(\boldsymbol{G}^T \cdot \boldsymbol{G}\right)^{-1} \tag{3}$$

Rather than computing the inverse matrix, we can use the inverse lemma to get a recursive formula [10]

$$\boldsymbol{H}^{(i)} = \boldsymbol{H} + \frac{\boldsymbol{S_i} \cdot \boldsymbol{S_i}^T}{p_{i,i}} \tag{4}$$

$$\boldsymbol{S} = \left(\boldsymbol{G}^T \cdot \boldsymbol{G}\right)^{-1} \cdot \boldsymbol{G}^T \tag{5}$$

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{G} \cdot \left(\boldsymbol{G}^T \cdot \boldsymbol{G}\right)^{-1} \cdot \boldsymbol{G}^T \tag{6}$$

where $\boldsymbol{H}^{(i)}$ is the weight coefficient matrix with the i^{th} satellite removed, $\boldsymbol{S}_i$ is the i^{th} column of the $/bmS$ matrix, $p_{i,i}$ is the element of line i, column i of the $\boldsymbol{P}$ matrix. Observe the formula (4), we can see that

$$h_{j,j}^{(i)} = h_{j,j} + \frac{s_{j,i}^2}{p_{i,i}} \tag{7}$$

It indicates that when deleting a single satellite, the increase in *GDOP* can be obtained by calculating the term $s_{j,i}^2/p_{i,i}$. From formula (2), when deleting the i^{th} satellite, we can use the following formula to express the increase of $GDOP^2$

$$C_{GDOP^2}^{(i)} = (\sum_{j=1}^{L+3} s_{j,i}^2)/p_{i,i} \tag{8}$$

We can calculate each $C_{GDOP^2}^{(i)}$ of all observed satellite, and select the satellites with k largest $C_{GDOP^2}^{(i)}$ values.

We name this method as the direct satellite-selection (D-S) method. The process is explained as follows.

1) Use formula (1) to compute the $\boldsymbol{G}$ matrix with all-in-view satellites, and get $\boldsymbol{H}$, $\boldsymbol{S}$ and $\boldsymbol{P}$ matrix and $C_{GDOP^2}^{(i)}$ from formula (3), (5), (6) and (8) respectively.
2) Find k maximum $C_{GDOP^2}^{(i)}$ and take the corresponding k satellites as the desired subset satellites.

We use a group of data collected in Shanghai to examine this algorithm. The data contains 3 constellations, including G-PS, GLONASS and BDS, and the observed satellite number is 27. We use both the DS method and brute force to operate the data. Figure 1 shows the curve of *GDOP* changing with selected satellite number k.

According to figure 1, the DS method is well agreed with the brute force method if k has a large value. But when k is getting smaller, the error between the DS method and the brute force method is gradually increasing, and the error dramatically increases when k has down to a small value. It is impermissible to use the subset satellites as working satellites for positioning when the k falls to small. In short, the experimental result indicates that the DS method does not fit in long step judgment. This method should be corrected by some restrictive conditions.

Generally, the DS method is a preferred algorithm, because it has a low calculation burden and high precision in a short step. If users are demanding for a fast satellite selection methodology and do not need to exclude too many satellites, this method is the most appropriate.

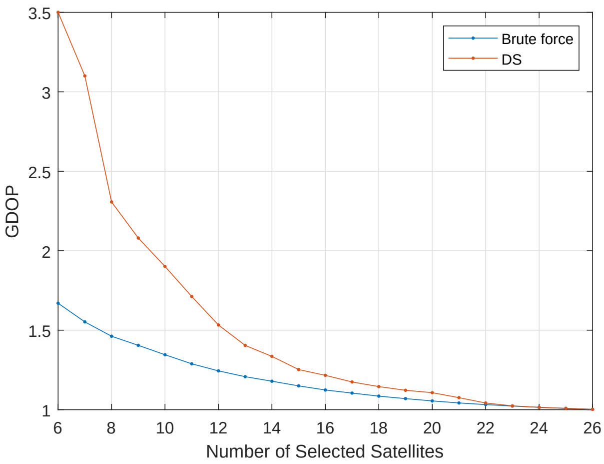


Fig. 1 Precision performance of the direct satellite-selection method, compared with the brute force method

3 Constrained direct satellite-selection method

The weakness of the DS method is to be expected. The relationship between *GDOP* and $C_{GDOP^2}^{(i)}$ is

$$GDOP^{(i)} = \sqrt{GDOP^2 + C_{GDOP^2}^{(i)}} \tag{9}$$

where $GDOP^{(i)}$ is the *GDOP* value with the i^{th} satellite removed.

The standard of judgment $C_{GDOP^2}^{(i)}$ can represent the increase of $GDOP^2$ unbiasedly only when $k = n-1$. To ensure the precision of formula (9), matrices must be refreshed according to (4), (5) and (6) after deleted one single satellite. Actually, it is similar to the recursive method if we refresh the matrixes in each step. And it will add a lot of algorithm complexity.

We can examine the error caused by each step. For the l^{th} step, the *GDOP* error is

$$E_{GDOP_l} = GDOP_l^{(i)} - GDOP_l \tag{10}$$

where the $GDOP_l^{(i)}$ is the biased *GDOP* value calculated by equation (9).$GDOP_l$ is the *GDOP* value with l selected satellites deleted, it is given by

$$GDOP_l = \sqrt{\text{trace}\left[\left(G_l{}^T \cdot G_l\right)^{-1}\right]} \tag{11}$$

The absolute value of E_{GDOP_l} will continuously increase as steps go on, and can get a large value. For the data used before, E_{GDOP_l} rises to -0.545 when $k = 10$, and the value is about 31% of its $GDOP_l$. It obviously performs an enormous influence to the DS step for tracking the optimal geometry. Thus the error E_{GDOP_l} should be limited in an appropriate range. Consider that the gradient of *GDOP* descent rises with the increase of l, it is not suitable to set a constant threshold to judge E_{GDOP_l}. Notice that large E_{GDOP_l} will influence the minimal $C^{(i)}_{GDOP^2}$ judgment, we employ $C^{(i)}_{GDOP^2}$ as the threshold. The constraint condition is given as follows:

$$E_{GDOP_l} > \min_{1 \le i \le n-l} C^{(i)}_{GDOP^2} \tag{12}$$

The inequality searches $C^{(i)}_{GDOP^2}$ of the remaining satellites for each step. If the constraint condition is satisfied, matrixes should be refreshed according to (4), (5) and (6) to reset E_{GDOP_l} and enter into a new epoch, this process continues until $l = k$. By the way, operators can also set the threshold as the above elements divided by d for higher precision:

$$E_{GDOP_l}{}^2 > \min_{1 \le i \le n-l} C^{(i)}_{GDOP^2}/d(d \ge 1) \tag{13}$$

The value of d is related to the trade-off between precision and computational efficiency. In the following sections, our examinations will mainly focus on inequality (13), since it already has a good performance.

In order to examine the error value E_{GDOP_l}, we need to obtain the $GDOP_l$ in each step. Instead of computing the inverse matrix, which is much time-consuming, inversion lemma can be used to optimize the calculation [4]. As a deformation of formula (4), the recursive formula is as follows:

$$\boldsymbol{H}^{(i)} = \boldsymbol{H} + \frac{\boldsymbol{H} \cdot \boldsymbol{g}_l \cdot \boldsymbol{g}_l^T \cdot \boldsymbol{H}}{1 - \boldsymbol{g}_l^T \cdot \boldsymbol{H} \cdot \boldsymbol{g}_l} \tag{14}$$

where $\boldsymbol{g_l}$ is the row vector deleted from the $\boldsymbol{G}$ matrix, and $GDOP_{l-1}$ is the $GDOP$ value in the last step. This formula can recursively compute the new $GDOP$ when one single satellite is deleted by the DS algorithm. We name this method as constrained direct satellite-selection (CDS). The core structure of the CDS method is represented in the scheme in figure 2, and the methodology is explained step by step in the following:

1) Use formula (1) to compute the $\boldsymbol{G}$ matrix with all-in-view satellites, and get $GDOP$, $\boldsymbol{H}$, $\boldsymbol{S}$ and $\boldsymbol{P}$ matrix and $C^{(i)}_{GDOP^2}$ from formula (2), (3), (5), (6) and (8) respectively.
2) Find minimum $C^{(i)}_{GDOP^2}$ from all remaining satellites and exclude the corresponding satellite in this step.
3) Compute the new $GDOP$ of this step by formula (14) and (2).
4) Use formula (9) to calculate the biased $GDOP^{(i)}_l$. Clear that this term is not the true $GDOP$. It is just a roughly estimated value computed by corresponding elements of the $\boldsymbol{S}$ and $\boldsymbol{P}$ matrix.
5) Compute the GDOP error E_{GDOP_l} of the l^{th} step by formula (10).
6) Judge the constraint condition given by (12), or (13).

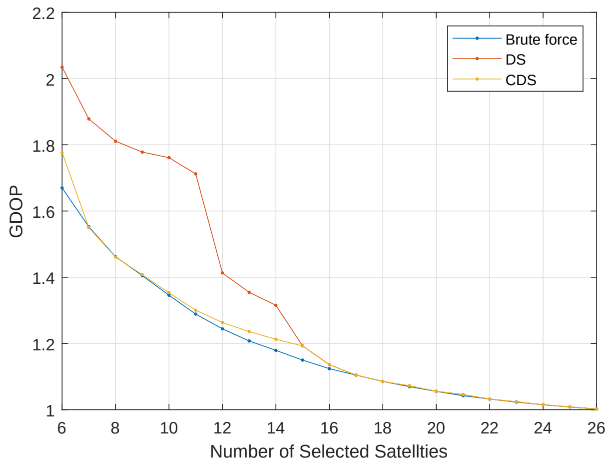


Fig. 3 Performance comparison between the DS method and the CDS method

a. If the constraint condition is satisfied, refresh the elements included in step 1 with the remaining satellites of last selection step.
b. If the constraint condition is not satisfied, it indicates that the error is not large enough to enormously influence the selection process. So the process can be continued to step 7

7) Examine whether the process reaches the desired satellite number. If not, jump to step 2 to delete more satellites. Or return the final subset satellites.

Using the same data as before, we can get a performance comparison between the DS method and the CDS method. The result is shown in figure 3. It indicates that the CDS method addressed the problem of the DS method and keep closer to the brute force. While the result discussed before is just based on a single group of data. The detailed test will be given in the later section.

Inevitably, the CDS method invested in more computing time. If the algorithm refreshed elements for ξ times, the big matrix computation (which is the main part of computing time) contains two parts. On the one hand, the recursive formula (66 matrix for 3 constellations) is calculated for $n-k+\xi$ times. On the other hand, the $\boldsymbol{H}$, $\boldsymbol{S}$ and $\boldsymbol{P}$ matrix ($n-l$ dimensions) is calculated for ξ times. Normally, the refreshing times ξ is a small number, which is just 4 when selecting 6 satellites out of 27 observations (Using the data above). Though its computation is larger than the DS method, it is much smaller than recursive method. The time performance will be examined in detail later.

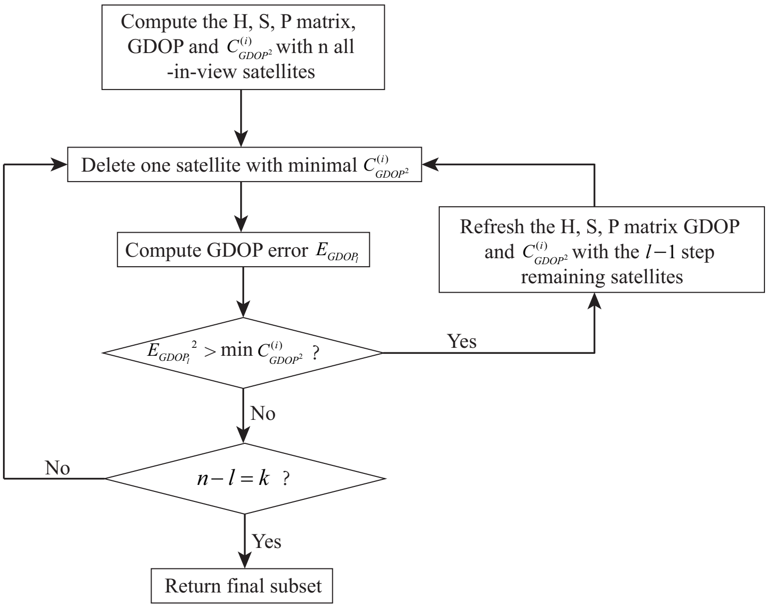


Fig. 2 Schematic overview of the constrained direct satellite-selection method

4 Experiments and discussion

In this section, an assessment of the proposed methodology will be given by comparing the results with the other methods, including the brute force and the recursive method.

Firstly, to obtain the performance of the proposed methods and the others when setting different selected number of satellites, we observed the satellites, including GPS, GLONASS and BDS, at Shanghai for 24 hours, with few signal occlusion. We operated four satellite selection algorithms every 10s, setting the desired number to 16, 12 and 10 separately, and compared them in each measurement time. Results are shown in figure 4.

According to figure 4(a), when set $k = 16$ the DS method and the CDS method both have good performance compared with the recursive method. The DS method seldom has protruding values. While, according to figure 4(b), there are more different values between the two proposed methods. According to figure 4(c), there are a lot of dramatic increases when using the DS method. Meanwhile, the CDS method is far more close to the recursive method, and there is no sharply increasing point during the whole testing time, though it performed worse than $k = 16$.

To examine the relationship between the desired number of satellites and the performance of two proposed methods, we compared them to the recursive method separately in each desired number. The result is shown in figure 5. It is clear that the performance of the two is similar when the number of selected satellites are above 14. The performance of the DS method get worse when the number of selected satellite falls down below 14, and it sharply increases when the number is below 8. Though the performance of

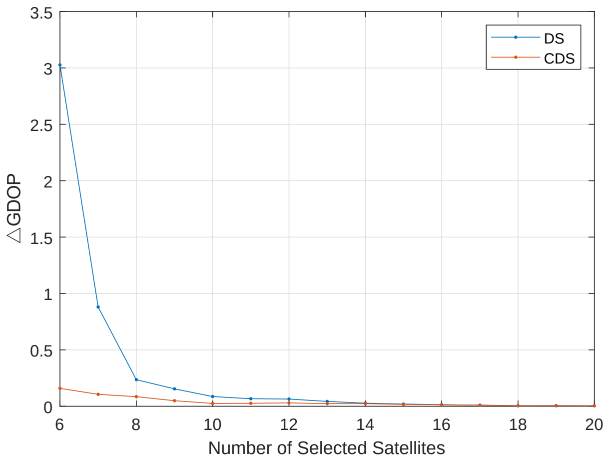
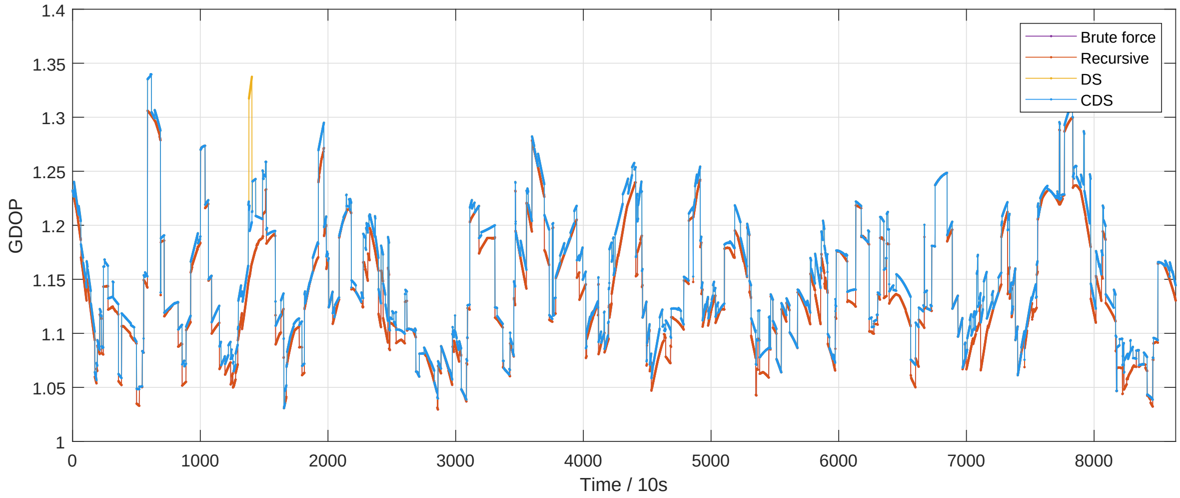


Fig. 5 Mean error of 24 hours relative to the recursive method

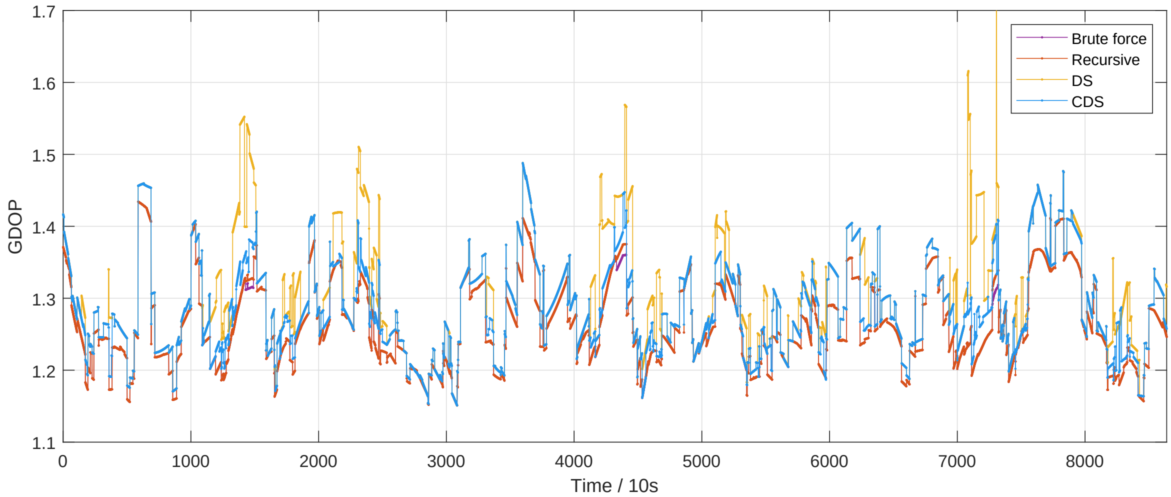
CDS method also get worse when the number of satellites decreasing, it is still much better than the DS method.

In order to verify the universality of the algorithms, we simulated satellites situation by MATLAB in the global range, also included the three constellations. We examined the DS method and CDS method globally in 24 hours, and compared them to the recursive method. Both desired number of satellites k are set as 10. Results are shown in figure 6. We can see that the $\Delta GDOP$ of CDS method is far smaller than that of DS method. Even the maximum value of CDS method approximately equal to the minimum value of the DS method.

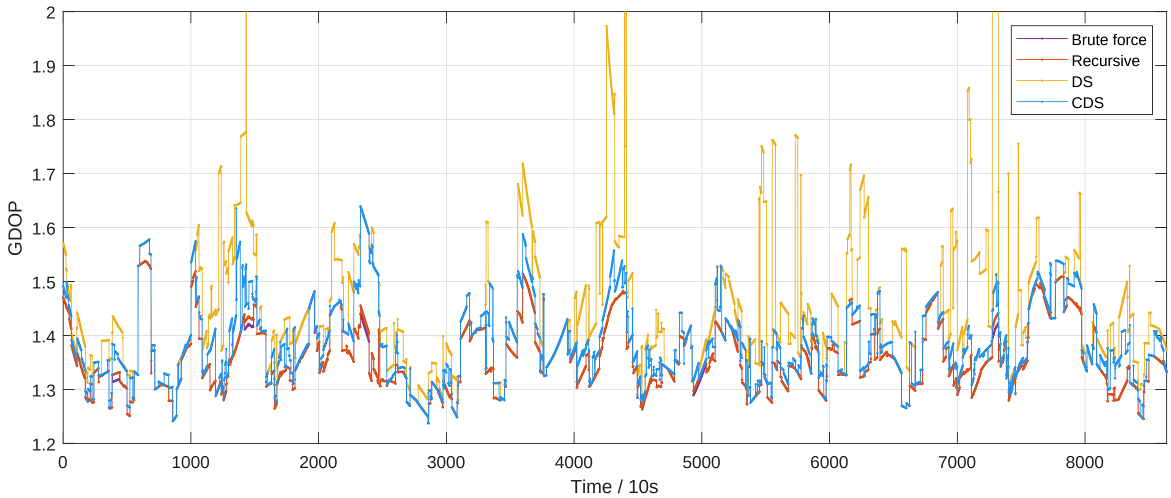
We can conclude that the two proposed method performs similar precision with each other when setting k as a big



(a) $k = 16$

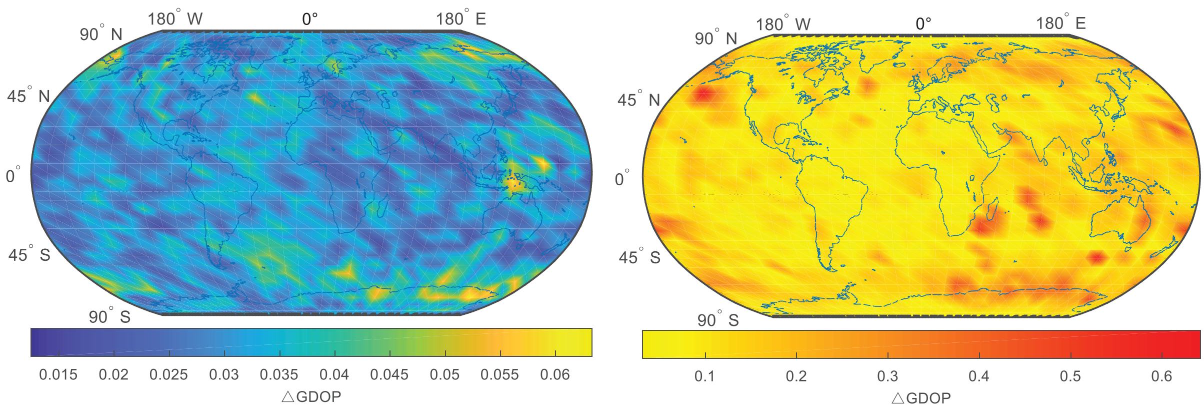


(b) $k = 12$



(c) $k = 10$

Fig. 4 Comparison of *GDOP* using the brute force, recursive method, proposed DS method and CDS method for 24 hours. The recursive method and the brute force are almost overlapped, so only compare the proposed method with the recursive method is sufficient.



(a) The GDS method (b) The DS method

Fig. 6 Global error relative to the recursive method ($k = 10$). Note that the error of the GDS method is about one order of magnitude lower than that of the DS method.

number. While, the CDS performs much better precision than the DS method when k is small.

Then we tested the computational load. It was discussed in the previous sections. In the simulation, we will give the computing time through specific tests. Note that all tests in this section are operated on the MATLAB environment, and the hardware setup includes an Intel Core i7-8700 CPU @ 3.20GHz and a 16 G RAM @ 2666MHz.

Following the discussion in the previous sections, we can get the time complexity of the brute force method is $O(C_n^k)$. And the time complexity of the DS method and the CDS method is constant order $O(1)$ and linear order $O(n)$. While the recursive is square order $O(n^2)$. Apparently, the computational load of brute force is several orders of magnitude higher than the recursive method and the two proposed method in almost all cases. It is no need to compare the brute force method with others. Thus we will not invest the brute force method into comparison.

Firstly, we simulated 42 observed satellites and used the recursive method, DS method and CDS method separately to select 6 to 30 satellites. All data are computed 30 times per selection epoch and averaged. Figure 7 shows the performances of the three algorithms. It is clear that the DS method takes the shortest time. It is not sensitive to the number of selected satellites, because it can directly select the subset satellites, without extra computation when the number of s-elected satellites changes. Meanwhile, the DS method takes an order of magnitude higher computation time than the CDS method. While, the recursive takes much more time. In theory, both recursive and the CDS method changes its computing time when changing the desired the number of s-elected satellites. But in this test, the variances over satellite number is so small that it is drowned in computer time error due to unstable frequency. Anyway, it is highly discernable to judge the computing time between the three algorithms.

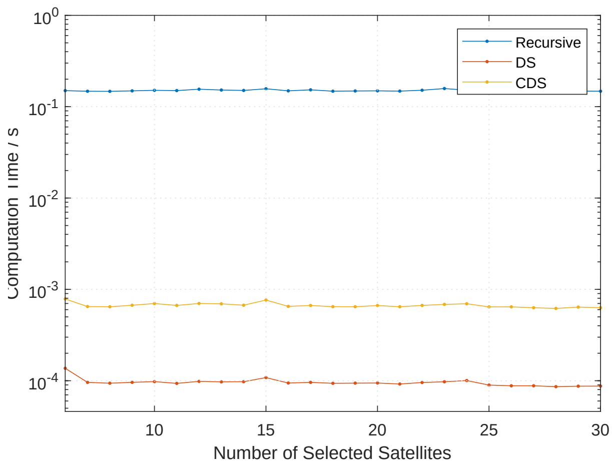


Fig. 7 Average computation time per selection when selecting 6 to 30 satellites out of 42 observed satellites

In figure 8, the computation time versus the number of observed satellites is given. We set the desired satellite number as 10, and generated 18 to 42 observed satellites in each epoch. It is shown in the figure that sequence of the three methods do not change in this case, and both the DS method and the CDS method are also not sensitive to the number of observed satellites. While, as for the recursive method, computation time increase as the observed satellites increasing. It is reasonable, because it must compute bigger matrix for several times when all-in-view number rising.

Consequently, the DS method performed the best in the computation load case, and the CDS method did not cause so much extra computation time compared to DS. However, the recursive method performed a much longer computation time, especially when the number of observed satellites is great. Hence, the best choice is the DS method in terms of efficiency, followed by the CDS method.

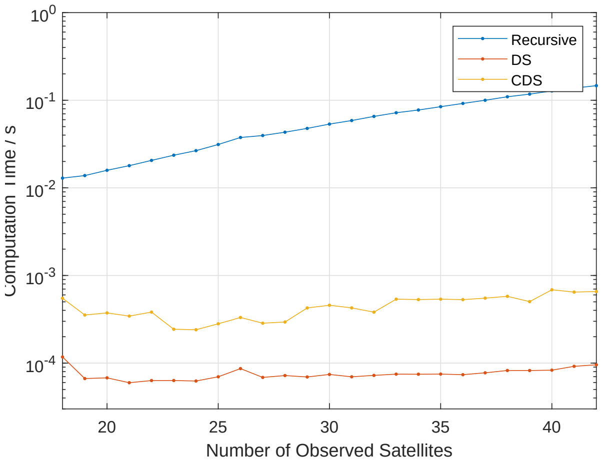


Fig. 8 Average computation time per selection when selecting 10 satellites out of 18 to 42 observed satellites

According to the performance tests shown above, it is safe to say that either the DS method or the CDS method is very fast satellite selection algorithm, with decent performance. To comparison, the DS method has the most efficient computational capacity but has unsatisfactory precision when the number of subset satellites is low. The CDS method has excellent performance covering all satellite quantity circumstances, with a larger computational time than the DS method. We suggest users use the CDS method when operating satellite selection because its high adaptability and fast computational performance, unless very short computational time is demanded.

The methods can also be used to automatically optimize GDOP value. When considering on the desired precision, it is more suitable to focus on maximal (or alert) GDOP, rather than the number of desired satellites, since the number of satellites and the value of GDOP are not absolutely corresponding and for fixed satellite numbers, satellites with better geometry can have a smaller GDOP value. Just as what figure 4(a)-4(c) indicate, take the most optimal one, recursive method, as an example, the GDOP changes with time changing when setting the stable number of subset satellites. Hence, in the actual operation, manufacturers can take the discussion above for guidance to decide how the number of subset satellites fits the GDOP value.

As we mentioned before, satellite measurement error is not included in the geometric selection method. Thus users should take care of the multipath, ionosphere error, and others for some satellite. The DS and the CDS method both prefer to select satellites with low elevation angle, which may have huge measurement error, such as multipath. The kind of satellites with low performance should be excluded before operating the satellite selection algorithms.

5 Conclusions

In this paper, two ultra-rapid satellite-selection algorithms were proposed, namely DS and CDS. We have identified the high computational time of the brute force, and demonstrated that even the current most efficient DOP based algorithm, recursive, could cost several orders of magnitude higher time to achieve satellite selection. We have also pointed out the different precision range of the two proposed methods.

Simulations showed that the DS method could save computational time for about three orders of magnitude lower than the recursive method, and the CDS method saved the time for about two order of magnitude lower. Both the two proposed methods showed almost the same precision performance when reducing the number of satellites from 42 to 16. Precision performances decrease when further reducing the number, with about 0.25 increase in DOP for the DS method relative to the recursive and about 0.10 for the GDS method, when reducing subset from 42 to 8. While, the error of the DS method could be large when the number falls below 8, but the CDS method kept high precision, which was about 0.2 error relative to recursive method. Thus, both of the two proposed methods have much lower computational load than existing DOP based satellite-selection algorithms and have very little decrease in precision.

Operators are suggested to use the CDS method for satellite selection for its high adaptability and fast computational performance. In very short computational time desired situation, special embedded system or other similar conditions, the DS method can be employed.

References

1. Mosavi M, Divband M (2010) Calculation of geometric dilution of precision using adaptive filtering technique based on evolutionary algorithms. In: 2010 International Conference on Electrical and Control Engineering, IEEE, DOI 10.1109/icece.2010.1171
2. Wang E, Jia C, Feng S, Tong G, He H, Qu P, Bie Y, Wang C, Jiang Y (2018) A new satellite selection algorithm for a multi-constellation GNSS receiver. In: Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Institute of Navigation, DOI 10.33012/2018.15993
3. Phatak M (2001) Recursive method for optimum GPS satellite selection. IEEE Transactions on Aerospace and Electronic Systems 37(2):751–754, DOI 10.1109/7.937488
4. Liu FMAJRL M (2009) A recursive quasi-optimal fast satellite selection method for gnss receivers. In: Proceedings of the 22nd International Technical Meeting of the Satellite Division of The Institute of Navigation

(ION GNSS 2009), Savannah, GA, September 2009, pp. 2061-2071
5. Swaszek PF, Hartnett RJ, Seals KC (2016) Multi-constellation GNSS: New bounds on DOP and a related satellite selection process. In: Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Institute of Navigation, DOI 10.33012/2016.14748
6. Kihara M, Okada T (1984) A satellite selection method and accuracy for the global positioning system. Navigation 31(1):8–20, DOI 10.1002/j.2161-4296.1984.tb00856.x
7. Blanco-Delgado N, Nunes FD (2010) Satellite selection method for multi-constellation GNSS using convex geometry. IEEE Transactions on Vehicular Technology 59(9):4289–4297, DOI 10.1109/tvt.2010.2072939
8. Zhang M, Zhang J (2009) A fast satellite selection algorithm: Beyond four satellites. IEEE Journal of Selected Topics in Signal Processing 3(5):740–747, DOI 10.1109/jstsp.2009.2028381
9. Li NAWLBS Jie (1999) Gps receiver satellite/antenna selection algorithm for the stanford gravity probe b relativity mission. In: Proceedings of the 1999 National Technical Meeting of The Institute of Navigation, San Diego, CA, January 1999, pp. 541-550
10. Walter T, Blanch J, Kropp V (2016) Satellite selection for multi-constellation SBAS. In: Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Institute of Navigation, DOI 10.33012/2016.14608
11. Swaszek PF, Hartnett RJ, Seals KC, Swaszek RMA (2017) A temporal algorithm for satellite subset selection in multi-constellation GNSS. In: Proceedings of the 2017 International Technical Meeting of The Institute of Navigation, Institute of Navigation, DOI 10.33012/2017.14904
12. Blanch J, Walter T, Enge P, Wallner S, Fernandez FA, Dellago R, Ioannides R, Hernandez IF, Belabbas B, Spletter A, Rippl M (2013) Critical elements for a multi-constellation advanced RAIM. Navigation 60(1):53–69, DOI 10.1002/navi.29