Research in Astron. Astrophys. **2009** Vol. **9** No. **11**, 1277–1284
http://www.raa-journal.org *http://www.iop.org/journals/raa*

Research in Astronomy and Astrophysics

A network flow algorithm to position tiles for LAMOST *

Guang-Wei Li and Gang Zhao

National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China;
gzhao@bao.ac.cn

Received 2009 March 23; accepted 2009 June 5

Abstract We introduce the network flow algorithm used by the Sloan Digital Sky Survey (SDSS) into the sky survey of the Large sky Area Multi-Object fiber Spectroscopic Telescope (LAMOST) to position tiles. Because fibers in LAMOST's focal plane are distributed uniformly, we cannot use SDSS' method directly. To solve this problem, firstly we divide the sky into many small blocks, and we also assume that all the targets that are in the same block have the same position, which is the center of the block. Secondly, we give a value to limit the number of the targets that the LAMOST focal plane can collect in one square degree so that it cannot collect too many targets in one small block. Thirdly, because the network flow algorithm used in this paper is a bipartite network, we do not use the general solution algorithm that was used by SDSS. Instead, we give our new faster solution method for this special network. Compared with the Convergent Mean Shift Algorithm, the network flow algorithm can decrease observation times with improved mean imaging quality. This algorithm also has a very fast running speed. It can distribute millions of targets in a few minutes using a common personal computer.

Key words: methods: data analysis — methods: statistical

1 INTRODUCTION

Astronomers can discover rich information from spectra of all kinds of objects. Using the traditional method, one can only obtain one spectrum at a time. So, it is an exhausting and time consuming endeavor to get many spectra. In order to solve this problem, the Two Degree Field system facility on the Anglo-Australian Telescope (AAT+2dF) and SDSS were developed. In general, however, a larger aperture has a smaller field of view (FOV). It is difficult for a modern telescope to simultaneously have a large FOV and a large aperture. However, Chinese scientists have solved this problem by designing the telescope LAMOST, which has both a large aperture and a large FOV.

The FOV (we call it the "tile" later in this article) of LAMOST is 5° and the effective aperture is 4 m. There are as many as 4000 optical fibers in its focal plane. These virtues make spectroscopic surveys faster and easier.

The Observation Control System (OCS) Research Group of the University of Science and Technology of China gave a series of methods to position tiles for LAMOST by Survey Strategy System (SSS) reports, including the Static Method (SSS-0 2002), Dynamic Method (SSS-1 2004) and Mean Shift Method (SSS-2.10 2006). Recently, Li & Zhao (2009) gave a Convergent Mean Shift Method.

The Static Method first selects some observation positions and then, at each position, keeps observing targets until no target is left. The Dynamic Method also first selects some observation positions and

* Supported by the National Natural Science Foundation of China.

then, among these positions, finds a position where the tile can cover the maximum number of targets to begin the observation. After the observation, the Dynamic Method deletes the targets that have been observed from the input catalog, and then finds the next position where the tile can cover the maximum number of targets to begin observation. It repeats the above steps until no target is left in the input catalog. The Mean Shift Method is a greedy method, and at each calculation, it tries to find the densest sky area. However, this method has a problem: it does not necessarily converge. It spends a lot of time finding the densest sky area. In order to solve this problem, Li & Zhao (2009) presented a Convergent Mean Shift Method. This method gives an objective function to describe the total imaging quality of the targets that are covered by the LAMOST focal plane. The method tries to find the position where the tile has the best imaging quality to begin the observation. This method is very good, but it is also a greedy method and is not necessarily globally optimal.

The network method used by Lupton et al. (1998) has good performance in SDSS, but as for LAMOST, there is still one thing that should be considered, which is that the distribution of LAMOST fibers on the focal plane is almost uniform, and each fiber can just move in a small circle with a $2.6'$ diameter. However, on SDSS' focal plane, the fibers can be adjusted according to the target positions and we do not need to consider if these targets are distributed uniformly. Thus, this method should not be used directly for LAMOST. In the rest of this paper, we will solve this problem.

In Section 2, we introduce network flow and Lagrangian relaxation. Then in Section 3, we discuss how to introduce the network flow into the LAMOST sky survey and also give our algorithm. The examples will be given in Section 4. In the last section, we will give our discussions and remarks.

2 INTRODUCTION TO NETWORK FLOW

We give figures to illustrate how the network flow method works (Blanton et al. 2003).

2.1 Terms of Network Flow

Figure 1 is a schematic figure for tiles to cover the targets (solid dots). There are a total of 6 targets, and 2 tiles. Figure 2 is the network flow structure for Figure 1. From Figure 2, we can give some basic terms for network flow theory. If someone wants to know more about network flow theory, *Network Flows: Theory, Algorithms, and Applications* (Ahuja et al. 2005) is a very valuable reference.

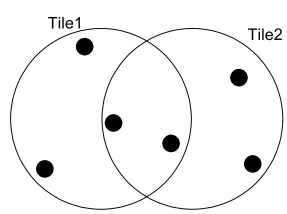

Fig. 1 Two big circles are Tile 1 and Tile 2. There are six targets which are denoted by solid dots.

A *node* is a solid dot in Figure 2. There are basically two kinds of nodes: each node on the left denotes one target; each node on the right denotes one tile. There are still two other special nodes: *source* and *sink*. The source is the point where the flow begins and the sink is the point where the flow ends.

An *arc* is a line between two nodes and denotes that the two nodes have some relationship. The arc between a tile node and a target node denotes that the target can be covered by the tile. We can think of it as a pipe where water or something else can flow.

The *capacity* of an arc is the maximum number of targets that the arc can carry. Usually, it is a positive integer. Here, the capacity of an arc between the source and a target node is 1. The capacity of

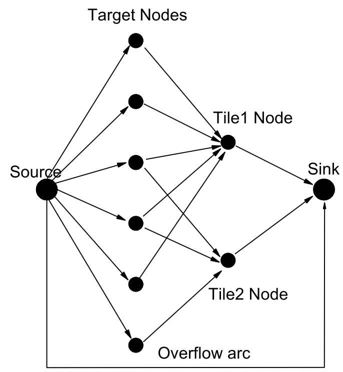

Fig. 2 This network flow comes from Fig. 1. There are four kinds of nodes: the source, target nodes, tile nodes and the sink. The arcs between them denote some relation. Capacity and cost of each arc are not given here.

an arc between a target node and a tile node is also 1. The capacity of an arc between a tile node and the sink is the maximum number of targets that LAMOST can accommodate.

The *cost* of an arc is the charge that we should pay for one target flowing along the arc. Each arc has a fixed cost. In this paper, we compute the cost of each arc by its distance: the further from a tile center the target is, the higher the cost of the arc between the target node and the tile node is.

2.2 Lagrangian Relaxation

In order to use network flow in a sky survey, we must introduce *Lagrangian relaxation* which can translate the problem of tile placement into the problem of minimum cost maximum flow. We now describe the Lagrangian relaxation used in a sky survey as follows.

Firstly, each target in the input catalog can be assigned to a tile with a penalty. The further the target is from the tile center, the higher the cost that the target should pay is. A target which lies outside of a tile can also be assigned to this tile, but a large penalty is needed. We use a penalty function to describe the cost of the penalty. The penalty function encourages nearer targets to be assigned to the tile. The arc between the source and the sink is an overflow arc and only when targets cannot be assigned to any tile do they flow on this arc. From the above, we can see that for a series of tile centers, the maximum assignment of targets to fibers can be translated into the problem of network flow that minimizes total cost. After we have constructed the flow, the next step is to solve it. There are many methods to solve the problem of minimum cost maximum flow, but we will propose our own method in this paper.

If we have solved the problem of minimum cost maximum flow, all targets can be split into different groups by tiles. The first group of targets is assigned to Tile 1, the second group of targets is assigned to Tile 2, ... , and the last group of targets has no tile to which it can be assigned. Each tile has only one group of targets, and except for the last group of targets, one group of targets can only be assigned to one tile. Subsequently, we can move tiles individually to minimize their penalties. If we want to find better tile placements, we can keep constructing flows, solving flows, and moving tiles until we get satisfactory tile positions.

3 THE APPLICATION OF NETWORK FLOW FOR LAMOST

3.1 Difference Between SDSS and LAMOST

The method described above cannot be used directly in LAMOST because there is an important difference between SDSS and LAMOST (Xing et al. 1997; Hu et al. 2000). The fibers in SDSS are plugged

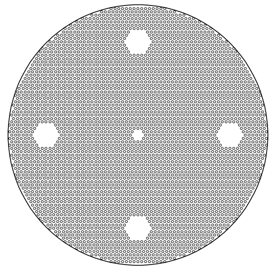

Fig. 3 There are five big holes in LAMOST's focal plane for special use.

in holes, and holes can be drilled anywhere with a distance of more than $55''$. However, each fiber in LAMOST's focal plane is fixed in the positioning unit and the movement range of each fiber is in a small circle with a $2.6'$ diameter. Moreover, the positioning units are uniformly distributed on the focal plane, which means that the fibers are also almost uniformly distributed on the focal plane.

Besides this, the network flow method with Lagrangian relaxation preferentially tries to assign targets near a tile center to that tile, and this tendency may cause too many targets near the tile center to be assigned. For example, we assume that there are 10 000 targets that a tile covers, but more than 5000 targets are in the area around that tile center, and we also suppose this area is 1 square degree. Then in this area, LAMOST's focal plane can collect at most 209 targets (the area of LAMOST's focal plane is 19.635 square degrees, and the focal plane can hold at most 4105 fibers if there were not five large holes in the focal plane).

Thus, the target number that LAMOST can observe in one square degree is $\frac{4105}{19.635} \doteq 209$), but the network flow method with Lagrangian relaxation still assigns 4105 targets to the tile, leaving targets outside of this area unassigned. Hence, it is obviously wrong. Therefore, the network flow method with Lagrangian relaxation of SDSS cannot be directly used in LAMOST.

3.2 Divide the Sky Coverage into Blocks

The problem of the network flow method with Lagrangian relaxation is that there are too many targets around the tile center being assigned to the tile, and these targets cannot be observed during one exposure. Therefore, we expect targets that are assigned to a tile to be distributed uniformly in the tile. For this reason, we can divide the sky into many small blocks, and the maximum number of targets that a tile can collect in each block is proportional to the block's area. Also, the proportion can be set arbitrarily, not just to 209. This proportion has a meaning: it is the maximum number of targets per square degree we want to assign to each tile.

How can we divide the sky? Maybe we can divide it arbitrarily, but it is not necessarily useful to do so. Fortunately, Hardin et al. (2001) gave a catalog of coverings of points on a sphere. These covering points are almost uniformly distributed on the sphere. We can divide the sky into many blocks, and each block center is a covering point.

3.3 Initial Placement of Tiles

At the beginning of this algorithm, we need initial tiles. How should we place initial tiles? Maybe we can place more tiles in the area with more targets and fewer tiles in the area with fewer targets and then we may get the optimal tile positions more quickly. How can we accomplish this? Fortunately Li & Zhao (2009) gave us the wanted placement of tiles. That study used a greedy algorithm known as the Convergent Mean Shift Algorithm to generate tile positions, and the area that has more targets will be placed with more tiles.

3.4 Constructing the Network Flow

Node: There are four kinds of nodes: the source, block nodes, tile nodes and the sink.

Arc: There are four kinds of arcs: arcs between the source and block nodes, arcs between block nodes and tile nodes, arcs between tile nodes and the sink, and the arc between the source and the sink. The arc between the source and the sink is an overflow arc which is for the targets that cannot be assigned to any tile on which to flow.

Capacity: The capacity of each arc has its special meaning: the capacity of the arc between the source and a block node is the number of targets in the block. The capacity of the arc between a block node and a tile node is proportional to the block area. For example, if we set the proportion to be 200, and the block area is 0.56 square degrees, then this arc can carry $200 \times 0.56 = 112$ targets, so its capacity is 112. The capacity of the arc between a tile node and the sink is the product of the proportion and tile area which is 19.635 square degrees. The capacity of the arc between the source and the sink is infinity, because we do not know how many targets cannot be assigned to tiles.

Cost: The cost of the arc between the source and a block node is 0. The cost of the arc between a block node and a tile node is a penalty value, and the further the covering point in the block is from the tile, the bigger the penalty value is. The cost of the arc between a tile node and the sink is 0, because if targets have been assigned to the tile, they do not need to be reassigned. The cost of the arc between the source and the sink is a very large number which is the maximum value of the penalty function, because we hope that if a target does not want to be assigned to any tile, it will pay the maximum cost, which will force as many targets as possible to be assigned to tiles.

3.5 A Quick Computational Method for Network Flow

After we have constructed the network flow, the next step is to solve it. The solution method (Goldberg 1997) of this problem in SDSS is very general, but this network flow has a very special structure which means that we can find a special method to solve this problem. This network is a *bipartite network*, which means that all nodes in the network can be partitioned into two parts such that all arcs are between the nodes that are in different parts. We can partition the nodes in the network flow into two parts: the first part includes the source and tile nodes; the second part includes the sink and block nodes. Then we can see that each arc in the network has the following character: one end is in the first part and one end is in the second part. Thus, our network is a bipartite network.

For this network, we propose a method named the *bipartite cost scaling algorithm* to solve it. This algorithm is from the book *Network Flows: Theory, Algorithm, and Applications* (Ahuja et al. 2005), page 364, with modifications to the procedure improve-approximation(ε, x, π) and the procedure push/relabel(i):

The general cost scaling algorithm is:

Procedure cost scaling algorithm
begin
  $\pi := 0; \varepsilon := C$;
  let x be any feasible flow;
  while $\varepsilon \geq \frac{1}{n}$ **do**
  begin
    improve-approximation(ε, x, π);
    $\varepsilon := \varepsilon/2$;
  end;
  x is an optimal flow for the minimum flow problem cost;
end;

The modifications of the procedure improve-approximation(ε, x, π) and the procedure push/relabel(i) are as follows:

Procedure improve-approximation(ε, x, π)
begin

compute all node balances;
while the network contains an active node **do**
begin
select an active node i;
push/relabel(i);
end;
end;
Procedure push/relabel(i)
begin
if the residual network contains an admissible arc (i, j) **then**;
if the residual network contains an admissible arc (j, k) **then**
push $\delta :=$ min$\{e(i), r_{ij}, r_{jk}\}$ units of flow over path $i - j - k$;
else $\pi(j) = \frac{\varepsilon}{2}+$ min$\{c^{\pi}_{jk} : r_{jk} > 0\}$;
else $\pi(i) = \frac{\varepsilon}{2}+$ min$\{c^{\pi}_{ij} : r_{ij} > 0\}$;
end;

This algorithm is faster than the general cost scaling algorithm in theory and its complexity is $O(n_1^2 m \log(nC))$, where n_1 is the number of nodes in the first part, m is the number of arcs, n is the number of nodes, and the positive integer C is the maximum arc cost in the network.

3.6 Moving Tiles

After the network has been solved, the targets in the catalog have been divided into $n_0 + 1$ groups, where n_0 is the number of tiles. Every tile covers its groups of targets and all the targets that cannot be assigned to any tile also form one group.

Now, if we assume that $f(r)$ is the penalty function, and $G(r) = \frac{f'(r)}{r}$, then we can use the iterative formulation $\boldsymbol{y}_{i+1} = \frac{\sum_{j=1}^{n} G(||\boldsymbol{x}_j - \boldsymbol{y}_i||)\omega_j(\boldsymbol{x}_j - \boldsymbol{y}_i)}{\sum_{j=1}^{n} G(||\boldsymbol{x}_j - \boldsymbol{y}_i||)\omega_j}$ mentioned in SSS2.10 (SSS2.10 2006) to find a new position for a tile. In this formulation, $\boldsymbol{x}_j$ is the position of a target that can be covered by the tile, and $\boldsymbol{y}_i$ is the iterative position of the tile.

After all the tiles have been moved, we construct the network flow and solve it again. If the total cost of the network flow decreases greatly, then we repeat these steps; else stop, and we finally get the tile positions.

3.7 Assign Targets to Fibers

After we have calculated the tile positions, the next thing we should do is to assign targets to fibers. We use the method named *minimum cost assignment*. This method is based on the cost of the network flow. It assigns targets in each block to tiles by cost, that is, the nearer the block is to a tile, the more preferentially the targets in the block are assigned to it. We can sort all the arcs in the network flow by cost, and because each arc also tells us the tile and the block that it connects, we can assign the targets in the block to the corresponding tile in arc cost order.

3.8 Outer Loop

Sections 3.3 – 3.7 describe the process of finding a better set of tile centers for a set of given tile centers by using the network flow method. We call this process the *inner loop*. Besides this, our program also has an *outer loop*.

In the outer loop, we would give the number of targets we want to observe, and then give many more tiles that can collect all these targets, and then decrease the tile number gradually until the inner loop cannot find tile positions where tiles can collect as many targets as we want.

4 EXAMPLES

We use USNO A2.0 (Monet et al. 1998) as our input catalog. The treatment of the catalog is the same as in the paper by Li & Zhao (2009). In order to compare results with the Convergent Mean Shift Method, we also use the same sky area as in their study. The penalty function we used is

$$P(r) = \begin{cases} 16 \times \frac{\mathrm{r}^2}{6.25} & r < 2.5, \\ 646.153846 \times \frac{\mathrm{r}^2}{6.25} - 630.153846 & 2.5 \le r < 4, \\ 1024 & r \ge 4. \end{cases} \quad (1)$$

4.1 Example for the Focal Plane with Edge Effect

The function we use to measure imaging quality is

$$f(r) = \begin{cases} 1 - (\mathrm{r}/2.5)^3 & r < 2.5, \\ 0 & r \ge 2.5. \end{cases} \quad (2)$$

We do four trials. The first observes about 85% of the total stars, the second about 90%, the third about 95%, and the last about 99%. Table 1 shows the results. The top part of Table 1 shows the results generated by the Convergent Mean Shift Method given by Li & Zhao (2009). The bottom part of Table 1 shows the results generated by the network flow method whose initial tiles are from the first part of Table 1.

The meanings of item titles are as follows: POS is the proportion of observed stars to total stars; NT is the number of targets; NCS is the number of all collected stars; MOQ is the mean observation quality and the larger this value is, the better the observation quality is; RFU is the ratio of fiber utilization, which is $\frac{NCS}{NT \times 3950}$. In the bottom part, TPSD is the maximum number of targets that can be assigned to a tile per square degree in the network flow.

From the two parts of Table 1, we can see that the network flow method can improve mean observation quality by using fewer tiles. From the bottom part of Table 1, we can also see that in order to get the lowest number of tiles, the more targets that are observed, the less proportion is needed in the network flow.

Table 1 Comparison of Convergent Mean Shift Method and Network Flow Method for the Case with Edge Effect

POS	NT	NCS	MOQ	RFU
85.03%	1 379	4 959 158	0.6334	91.04%
90.00%	1 518	5 249 154	0.6374	87.54%
95.02%	1 710	5 541 588	0.6416	82.04%
99.00%	2 048	5 774 124	0.6463	71.38%

POS	NT	NCS	MOQ	RFU	FPSD
85.05%	1 328	4 960 119	0.6370	94.56%	209
90.02%	1 447	5 249 991	0.6468	91.85%	200
95.01%	1 624	5 540 960	0.6686	86.38%	188
99.01%	1 984	5 774 384	0.7234	73.68%	182

Table 2 Comparison of Convergent Mean Shift Method and Network Flow Method for the Case without Edge Effect

POS	NT	NCS	MOQ	RFU
85.01%	1 318	4 957 705	0.6197	95.23%
90.00%	1 436	5 249 230	0.6205	92.54%
95.01%	1 601	5 541 282	0.6215	87.62%

POS	NT	NCS	MOQ	RFU	FPSD
85.07%	1 318	4 961 600	0.6332	95.30%	208
90.06%	1 440	5 252 333	0.6439	92.34%	208
95.05%	1 605	5 543 353	0.6627	87.43%	208

4.2 Example for the Focal Plane without Edge Effect

Again, we use the Selecting Star Algorithm without Edge Effect (Li & Zhao 2009) to reanalyze this example; the results are listed in the first part of Table 2. The top part of Table 2 shows the results of the network flow whose initial tiles are from the first part of Table 2. Because all stars in a tile have the same observation quality, MOQs have no meaning here, but we also calculate them with Equation (2).

From the results shown in Table 2, we can see that the network flow method does not decrease the number of tiles. For the case with edge effect, however, we can use the Mean Shift Method without edge effect to generalize the initial tiles, and then use the network flow method to improve the mean observation quality. For example, when we survey the sky near Dec 60°, the imaging quality of a target that is near the tile edge is much worse than the imaging quality of a target that is near the tile center. However, if you still care more about observation times than mean imaging quality, then you can use the Mean Shift Method without edge effect to generalize initial tiles, and subsequently use the network flow method to improve mean observation quality.

4.3 Comparison of the Two Examples

From Table 1, we can see that, compared with the Convergent Mean Shift Method, the network flow method not only improves the observation quality but also decreases the observation times for observing the same number of targets for a focal plane with edge effects. As for the focal plane without edge effect case, in order to observe the same number of targets, we cannot decrease the number of tiles. There may be three reasons for this: the first reason is that the fiber utilization ratio is very high; the second reason is that the simplification of the sky area is not good enough, and the blocks should be smaller; the last reason is that some parameters in the program may not be chosen properly.

5 DISCUSSION AND REMARKS

In this paper, we have introduced the network flow method into the LAMOST sky survey strategy. The performance of this method does well in decreasing tile number and improving observation quality. From our experiments, we also found that our network flow algorithm is very fast, and the calculation of the network flow can be finished in a few minutes. The computer we used is a DELL microcomputer with an Intel Pentium(R) 4 2.66 GHz processor and 512 Mbytes of memory. Most of the time is consumed by the assignment of targets to tiles.

However, there are still some problems to be studied. For example, how do we choose the penalty function? How can the sky coverage be divided into smaller blocks? Is it true that the smaller the blocks are, the better performance the program gives?

Acknowledgements This research is supported by the National Natural Science Foundation of China (Grant No. 10821061), and by the National Basic Research Program of China (973 Program) (Grant No. 2007CB815103).

References

Ahuja, R. K., et al. 2005, Network flows: Theory, Algorithms, and Applications (Beijing: Chinese Machine Press)
Blanton, M. R., Lupton, R. H., Maley, F. M., Young, N., Zehavi, I., & Loveday, J. 2003, AJ, 125, 2276
Goldberg, A. V. 1997, J. Algorithms, 22, 1
Hardin, R. H., Sloane, N. J. A., & Smith, W. D. 2001, *http://www.research.att.com/~njas/icosahedral.codes/*
Hu, H. Z., Xing, X. Z., Li, W. M., Zhai, C., Yu, Q. Y., Chen, H. Q, Qiu, L. H., & Ren, L. Z. 2000, Journal of University of Science and Technology of China, 30(6), 707
Li, G. W., & Zhao, G. 2009, RAA (Research Astron. Astrophys.), 9, 803
Lupton, R. H., Maley, F. M., & Young, N. 1998, J. Algorithms, 27, 339
Monet, D., Bird, A., Canzian, B., et al. 1998, *http://vizier.u-strasbg.fr/vizier/VizieR/pmm/usno2.htx*
OCS Research Grop, 2002, SSS-0 System Design Report
OCS Research Grop, 2004, SSS-1 System Design Report
OCS Research Grop, 2006, SSS-2.10 System Design Report
Xing, X. Z., Hu, H. Z., Du, H. S., Zhai, C., & Zhu, B. 1997, Journal of University of Science and Technology of China, 27(4), 492