

Data Processing System for LP and their uses in Modern Days

Anand Mohan Sinha, Kumar Mukesh, Uma Shankar

Abstract: This paper deals the data processing system for LP and their uses in modern days.

Key words and phrases: Linear Programming in data flows & linear approximation.

AMS Subject Classification 2010: 90C05

I. INTRODUCTION

The general discussion of the data flow algorithmic models, and the linear programming problem with the varying by data flow criterion function co-efficients are presented. The general problem is widely known in different names – data streams, incremental and online algorithms, etc. The more studied algorithmic models include mathematical statistics and clustering, histograms and wavelets, sorting, set cover, and others. Linear programming model is an addition to this list. Large theoretical knowledge exists in this as the simplex algorithm and as interior point methods but the flow analysis requires another interpretation of optimal plans and plan transition with variate co-efficients. An approximate model is devised which predicts the boundary stability point for the current optimal plan. This is valuable preparatory information of applications, moreover when a parallel computational facility is supposed. Data flow is a concept, traditionally appearing in the sensor based monitoring systems. Advanced global networks brought a number of novel applied disciplines intensively dealing with data flows. The network monitoring itself and optimal management of telecommunication systems, search engines with consequent data analysis, the network measuring instruments and network monitoring for security, etc. are the novel examples of data flow models. These deal with continuous data flows and unusual, non-finite and non-stored data set. In this case, the queries (the data analysis requests) are long-term and continuous processes in contrast to usual on-time queries. The traditional databases and data processing algorithms are poorly adjusted for the hard and continuous queries in data flows. This generates the necessity of new studies for serving continuous, multilayered, depending on time and subjected to indefinite behavior of data flows. Concerning the mentioned problem area, systems and algorithms are devised for different needs: real time systems, automation control systems, modeling processes, etc., but they are episodes in point of view of the formulated general problem.

Traditional trade offs of such systems include one-pass and multi-pass algorithms, deterministic and randomized algorithms, and exact and approximate algorithms. Off –line algorithms solve a problem with full knowledge of the complete problem data. Online algorithms construct partial solutions with partial knowledge of the problem data, and update their solutions every time some new information is provided. In other words, they must handle a sequence of closely related and interleaved sub-problems, satisfying each sub-problem without knowledge of the future sub-problems. Standard examples of online problems include scheduling the motion of elevators, finding routes in networks and allocating cache memory. The usual way of measuring the quality of an online algorithm is to compare it to the optimal solution of the corresponding off-line problem where all information is available at the beginning. An online algorithm that always delivers results that are only a constant factor away from the corresponding optimum off-line solution is called a competitive algorithm. The “incremental update” algorithmic model of data analysis modifies the solution of a problem that has been changed, rather than re-solving the entire problem. For example, partial change of conditions of a time-table problem must be force to only partial reconstruction of the table. It is obvious that it is possible to construct a theoretical problem, where any particular change brings to the full reconstruction of the problem. It is also clear that there are numerous problems, which are not so critical to the local transformations. It is an option to try to solve the given data flow problem by the mentioned incremental algorithms, moreover, in the specific conditions it is the only possible way for solving the problem including the data flows analysis. The chapter below discusses an important applied model for the flow environments. We consider the linear programming mathematical problem, parameters of which are formed by data flows. In a moment it is assumed that the optimal plan is found and the coordinates of the target function are variable by the flow. In this case, there is an emerging question: which mechanisms are well suited to follow the co-efficients variations in creating the configuration of the next resulting optimal plan. It is clear that the small changes of co-efficients lead to simple changes of the current optimal plan, probably not requiring the total analysis of the problem by the complete flow information.

II. LINEAR PROGRAMMING IN DATA FLOWS

Let's consider the linear programming problem in its canonical form:

$$\begin{aligned} \min \quad & c'x \\ \text{Ax} = & b, \quad x \geq 0, \end{aligned}$$

Manuscript Received on September 2014.

Anand Mohan Sinha, Department of Mathematics, J. P. University, Chapra, Bihar, India.

Kumar Mukesh, Department of Mathematics, R.P. Sharma Institute of Technology, Patna, Bihar, India.

Uma Shankar, Department of Mathematics, K. K. College of Engineering & Management, Biharsharif, Bihar, India.

Where $c \in R^n, b \in R^m$, A is an $m \times n$ full rank real matrix, and $m < n$. Without a loss of generality we may also suppose that $b_i \geq 0, i = 1, m$. Particular examples of linear programming problem are given through the definition of co-efficient values: a_{ij}, c_j, b_i for $i = 1, m; j = 1, n$.

Let's imagine a specific situation arising from application that the mentioned co-efficients are changing in time. Such problems appear, for example, in data flow analysis.

Let us consider a data flow $B(t, n)$, which is finite but a very large-sized sequence of values b_1, b_2, \dots, b_n , where $b_t, t = 1, n$ are certain structures. The data flows processing algorithms may use comparatively small storages than the input size. A limitation window is given in certain cases for processing separate data fragments. The time-dependent values of parameters, forming the applied problem model are formed as a result of algorithmic analysis. Unlike other natural and similar definitions, the variation of parameters is unpredictable here, as it has not probabilistic distribution and is not described by one or another property, instead, it is considered that the variation takes place very slowly, because of the accumulation concept. In its turn, the applied problem demands to have ready answers to the certain questions in each time stamp. There are two strategies: (1) solving a problem for each stage by all actual information which is practically impossible because of the large data sizes; and (2) structuring hereditary systems when the new data analysis is relatively easily integrated with the results of the previous data analysis. We are going to consider the linear programming model in the mentioned conditions. The arbitrary variation of the co-efficients is not allowed, instead, slow variations are considered so that the variation is fully monitored and it changes the solutions very slowly. Of course, it is possible to formalize this fully. At the same time, it is possible to consider partial behavior of parameters variations, providing simple scenes of the algorithmic developments. Let's suppose that the co-efficients c_j of

linear criterion function $z = \sum_{j=1}^n c_j x_j$ of the linear

programming problem are varying by flow $B(t, n)$.

Assume that t_0 is the moment where the complete analysis exists, i.e. we know about the existence of optimization at that moment and the optimal vertex and plan, if the latter exists. This vertex obeys the property of stability of optimality for certain variations of coefficients c_j . The stability area is described by a set of simple inequalities and it is clear that it is an issue to consider the border of this area. The theoretical analysis of optimality of vertex set elements of the area of base restrictions is well known as the simplex method. The simplex method looks for sequence chains of vertex transitions, which converge to an optimal plan. Complementary, in our case, we study all the possible ways of optimality transitions driven by the changes of coefficients c_j . The novelty is that we device the concept

of equivalency of vertices groups of the feasible polyhedron vertices set and prove that the transition from one optimal vertex to another takes place through these groups. So the continuous change of target function coefficients generates the continuous change of optimality vertices. From practical point of view – a path prediction process is possible to apply to the situation with varying coefficients. Prediction of intersection of the trajectory extrapolation of coefficient changes to the boundary of stability area of the current optimal plan helps to determine the vertex equivalency cluster and so – the further possible transitions and by these – the most likely arriving optimums when coefficients keep the track of their current modifications. Going through the transitions some vertices might neighbour with comparatively large equivalency groups of vertices and then the total number of those vertices can become large. Theoretically, in terms of flows, having the current optimization vertex, it is necessary to prepare neighbouring equivalent vertices by calculating them by the current and predicted coefficients c_j . The weakness of the direct

application of the given approach is in drastic increase in the number of calculations for the vertex sets involving the predictions and equivalencies. The considered below natural approach gives primary significance to the vertices which correspond to the linear approximations of the given variations. Let's denote the optimal vertex at the moment t_0

by \hat{x}^{t_0} and let \hat{c}^{t_0} is the corresponding vector of coefficients. Let's follow the transition of \hat{c}^{t_0} to the \hat{c}^t . It is clear that this transition is more or less arbitrary as it is controlled by the flow. It is important if during the transition

the vector \hat{c}^l of coefficients approaches to the border of stability of current optimal plan \hat{x}^{t_0} , - or not. To see this

we have to monitor the changes of \hat{c}^l . Alternatively, it is possible to approximate the transition, activating the possible future "optimal plans". For example, spline approximations or a more simple approximation by the main values and standard deviations might be applied. The most simple is the linear approximation model, which we consider below. As an extrapolation, it leads to the intersection with the stability boundary (shell) of the vertex \hat{x}^{t_0} at the most probability point. In case of sufficient computational resources, it is also possible to consider some neighbourhood of that point, and it is important that in contrast to the above mentioned theoretical model, this applied approach gives an opportunity to work with the limited quantity of the possible candidate vertices. Depending on the considered problem, an algorithmic system is able to choose a corresponding extrapolation scheme, which deals with different numbers of neighbouring vertices. The approximation of \hat{c}^l by the flow averages and dispersions requires their calculation, which is a simple flow problem.

Supposing that this question is clarified, let's consider the problem behavior in the case of linear approximations.

III. LINEAR APPROXIMATION

In the case mentioned, variation of the optimization criteria function coefficients is supposed to be changed by an expression $c_j(\lambda) = c_j^{l_0} + \lambda(c_j^l - c_j^{l_0})$, where λ varies in certain limits. The interval $[0, 1]$ for λ is internal and characterizes the variation from c^{l_0} to c^l , and the values $\lambda > 1$ are extrapolating the further behavior of coefficients in a linear model. Let's denote $c_j^\Delta = c_j^l - c_j^{l_0}$. So we are given the linear function

$$z = \sum_{j=1}^n (c_j^{l_0} + \lambda c_j^\Delta) x_j \quad (1.1)$$

And the system of linear requirements, given by

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, 2, \dots, m, \\ x_j &\geq 0, \quad j = 1, 2, \dots, m, \end{aligned} \quad (1.2)$$

It is necessary to accompany the changes of λ , finding out in the interval $1 < \lambda$ the minimal value at which the change of the optimal plan takes place for the first time. Assume that the vector $\tilde{x}^1 = (x_1^l, x_2^l, \dots, x_m^l)$ which satisfies the system (1.2) introduce the corresponding new optimization basis. According to the assumptions, we have optimal solution when $\lambda = 0$. Assume that the solution basis consists of the first m vectors of a_1, \dots, a_n . In accord to the simplex algorithm and its optimization condition, all the "estimations" in this case must obey to the following condition: $j_j - c_j^{l_0} \leq 0, j = 1, 2, \dots, n$.

As $c_j(\lambda) = c_j^{l_0} + \lambda c_j^\Delta$ then that general optimization condition becomes:

$$z_j - c_j(\lambda) = (c_j^{l_0} + \lambda c_j^\Delta) x_j^0 - (c_j^{l_0} + \lambda c_j^\Delta) \leq 0, \quad j = 1, 2, \dots, n,$$

Let's group the expression in the following way:

$$c_j^{l_0} (x_j^0 - 1) + \lambda c_j^\Delta (x_j^0 - 1) \leq 0, \quad j = 1, 2, \dots, n$$

And let introduce the notations: $\alpha_j = c_j^{l_0} (x_j^0 - 1)$ and $\beta_j = c_j^\Delta (x_j^0 - 1)$. The constants α_j and β_j are defined by the initial configuration: optimization criterion function coefficients and the corresponding solution basis, criterion current coefficients with the supposition that

optimization did not change during that period. For $\lambda = 0$ we have the optimization vertex \tilde{x}^0 , and therefore, we get the following limitations: $\alpha_j = c_j^{l_0} (x_j^0 - 1) \leq 0$. The optimization vertex change does not take place when $0 \leq \lambda \leq 1$, so we get also:

$$\alpha + \lambda \beta_j = c_j^{l_0} (x_j^0 - 1) + \lambda c_j^\Delta (x_j^0 - 1) \leq 0$$

In particular, when $\lambda = 1$ we get $\alpha_j + \beta_j \leq 0$. The extreme condition will be written in the following general form: $\alpha_j + \lambda \beta_j \leq 0, j = 1, 2, \dots, n$. Let's find the minimal value of λ at which at least one of this inequalities violates for the first time.

Let's separate negative and positive cases of β_j . The restrictions on λ will accept the following forms:

$$\begin{aligned} \lambda &\geq -\alpha_j / \beta_j \quad \text{for all } \beta_j < 0, \text{ and} \\ \lambda &\leq -\alpha_j / \beta_j \quad \text{for all } \beta_j > 0. \end{aligned}$$

Let's introduce one more notation:

$$\lambda = \{\min(-\alpha_j / \beta_j) \text{ if } \beta_j > 0, \text{ and } +\infty, \text{ when all } \beta_j \leq 0\}$$

The optimal solution for $\lambda = 0$ coincides with optimal solutions for all λ which obeys the condition $0 \leq \lambda \leq \lambda$. It is ensured that λ is the possible transition configuration. If $\lambda = +\infty$ then there is no change of optimal plan. If λ is limited then it is necessary to consider two cases: the first one (a) is the point λ with the possible equivalent optimal plans and possible continuations in this case, and the second one (b): if there is a new optimal plan and if the problem has no solution at $\lambda > \lambda$.

[A]

Assume that λ is finite, i.e. $\lambda = -\alpha_k / \beta_k$ for the corresponding value of parameter k . It means that $z_k - c_k(\lambda) = 0$ from which follows that the optimization plan is not single. Actually, let's insert the k -th vector into the basis and according to the simplex method let's exclude one of the vectors from the previous basis. We will get a new optimal plan the criterion value of which will stay unchanged. It follows even more - that, by all null estimations and by all basis modifications we can get many optimization equivalent vertexes and all elements of their linear closure also have the same discussing optimization value.

[B]

In this case, we consider the values $\lambda > \lambda$ and λ is finite. If the coefficients of above mentioned k -th vector all not positive, i.e. $\tau_{ik} \leq 0$,

by optimization basis, then according to the simplex method, the criterion function becomes unlimited. This takes place any time when according to the increasing character of the criterion function we get the vector which is going to be involved into the basis $z_k - c_k(\lambda) > 0$, but it becomes clear that the vector has no positive $\tau_{ik} > 0$ coordinate because of we could exclude it from the basis. In this case, it is impossible to choose such a coefficient $\theta > 0$ that any $xi - \theta\tau_{ik} = 0$ when $i = \{1, \dots, m\}$. Therefore, we get the optimization plan with $m + 1$ positive components; the set of $a_1, a_2, \dots, a_m, a_k$ vectors are linearly depending and this corresponds to the non-angle vertex. Therefore, linear criterion function could not get to its minimal value. This means that hyper-plane which is defined by linear function could not become supporting hyper-plane of permissible polyhedron at any shift in the direction of gradient. If a $\tau_{ik} > 0$ then the vector a_k is included into the basis and another vector a_1 is excluded from it. As the new basis is constructed by the simplex method then it corresponds to a new optimal solution, and at those inequalities

$$\alpha'_j + \lambda\beta'_j \leq 0, j = 1, 2, \dots, n \quad (1.3)$$

are compatible.

Let's show that any $\lambda < \lambda$ does not satisfy the system (1.3) of inequalities. Really, for the vector a_1 , excluded from the basis we will get the following:

$$\alpha'_1 = -a_k / \tau_{1k}; \beta'_1 = -\beta_k \tau_{1k}, \quad (1.4)$$

Where $\tau_{1k} > 0$. Suppose that (3) takes place for any $\lambda < \lambda$ then $\alpha'_j + \lambda\beta'_j \leq 0$, or according to (4) $-\alpha_k - \lambda\beta_k \leq 0$. As $\beta_k > 0$ then, from the latter inequality follows that $\lambda \geq -\alpha_k / \beta_k = \lambda$.

REFERENCES

1. E. D. Andersen and Y. Ye (1999), 'On a homogeneous algorithm for monotone complementarity system', Math. Program. 84, 375-399.
2. H. H. Bauschke, O. Guler, A. S. Lewis and H. S. sendov (2001), 'Hyperbolic polynomials and convex analysis', Canad. J. Math. 53, 470-488.
3. A. Ben-Tal and A. S. Nemirovski (2001), Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications, SIAM, Philadelphia.
4. R. E. 13ixby (2002), 'solving real-world linear programs: A decade and more of progress', Oper. Res. 50, 3-15.
5. S. Boyd and L. Vandenberghe (2004), Convex Optimization, Cambridge University Press.
6. S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan (1994), Linear Matrix Inequalities in System and Control Theory, Vol. 15 of studies in Applied Mathematics, SIAM.
7. R. H. Byrd, J. Nocedal and R. A. Waltz (2006), Knitro: An integrated package for nonlinear optimization. In large-Scale Nonlinear Optimization (G. di Pillo and M. Romma, eds), Springer, New York, pp. 35-59.
8. J. S. Chai and K. C. Toh (2007), 'Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming', Comput. Optim. Appl. 36, 221-247.

9. L. Chen and D. Goldfarb (2006), 'Interior-point 12-penalty methods for nonlinear programming with strong convergence properties', Math. Program. 108, 1-36.
10. J. Faraut and A. Koranyi (1994), Analysis on Symmetric Cones, Clarendon Press, Oxford.
11. L. Faybusovich (1997), 'Linear systems in Jordan algebras and primal-dual interior-point algorithms', J. Comput. Appl. Math. 86, 149-175.