A Graph Model Proposal for Convex Non Linear Separable Problems with Linear Constraints

Jaime Cerda, Alberto Avalos

Abstract—The solution of non linear problems are based on very well grounded mathematical theories. This document proposes a Newton step graph-based model for convex non linear separable problems (NLP) with linear constraints. The Newton step is well suited for this kind of problems, but when the problem size grows the NLSP model will grow in a non linear manner. Furthermore, the constraints handling becomes the main problem as we have to select the right constraints in the different solution steps. When this happens, the sparse matrix representation is the path to follow, but very little has been made in order to fully explode the sparsity structure. Indeed, the Hessian matrix for the NLSP model has a very particular structure which can be exploited by using the graph underlying the problem, this is the approach taken in this document. To this end a graph is built derived from the components involved in the Newton step. which describes the solution for the NLSP problem. Based on this graph, the gradient can be evaluated directly based on the graph topology, as it will be shown, the information needed for such evaluation is embedded within the graph. Furthermore, the explicit graph model derived from the mathematical model allows us to think about it in terms of its structure which will be used in further works.

Index Terms—Non Linear Programming, Linear Constraints, Graph-based Systems, KKT Conditions.

NOMENCLATURE

N	Number of decision variables.
L	Number of equality constraints.
M	Number of inequality constraints.
z_i	Decision variable <i>i</i> .
$\begin{bmatrix} z_i \end{bmatrix}$	Upper limit value of variable z_i .
$ z_i $	Lower limit value of variable z_i .
$\frac{1}{z_i}$	Slack variable for z_i uppper bound.
z_i	Slack variable for z_i lower bound.
$\overline{\Delta}z_i$	Variable x_i increment.
f(z)	Objective function
$g_l(z)$	Equality constraint <i>l</i> .
$h_m(z)$	Inequality constraint m.
a_{li}	<i>i</i> th coefficient in equality constraint <i>l</i> .
b_{mi}	ith coefficient in inequality constraint m .
r_l	RHS of equality constraint <i>l</i> .
s_m	RHS of inequality constraint m.
λ_l	Dual variable for equality constraint <i>l</i> .
μ_m	Dual variable for inequality constraint m .
$\overline{ ho}_i$	Dual variable for z_i uppper bound.
$\underline{\rho}_i$	Dual variable for z_i lower bound.
$\bar{\wp}(S)$	Power set of set S.
Γ_{ϵ}	Set of variables connected to z_i

Manuscript received July 23, 2013; revised August 6, 2013. This work was supported by the Scientific Research Coordination, Universidad Michoacana de San Nicolas de Hidalgo

J. Cerda (jcerda@umich.mx) and J. Avalos (javalos@umich.mx) are with the Electrical Engineering School, Universidad Michoacana de San Nicolas de Hidalgo, Morelia, Michoacan, Mexico

I. INTRODUCTION

Non linear separable problems (NLSP) are NLPs whose objective function can be decomposed as a sum of functions with only one variable each. The solution of NLSPs are based on very well grounded mathematical theories [1]. This document proposes a Newton step graph-based model for convex non linear separable problems (NLP) with linear constraints. The Newton step is well suited for this kind of problems, but when the problem size grows the NLSP model will grow in a non linear manner. Furthermore, the constraints handling task becomes the main problem as we have to select the right constraints in the different solution steps. When this happens, the sparse matrix representation is the path to follow, but very little has been made in order to fully exploit the sparsity structure. Indeed, the Hessian matrix for the NLSP model has a very particular structure which can be exploited by using the graph underlying the problem, this is the approach taken in this document. To this end a graph is built derived from the components involved in the Newton step. which describes the solution for the NLSP problem. Based on this graph, the gradient can be evaluated directly based on the graph topology, as it will be shown, the information needed for such evaluation is embedded within the graph. Furthermore, the explicit graph model derived from the mathematical model allows us to think about it in terms of its structure which will be used to derive more efficient models as well as decentralization tasks. Finally, even that this model is for convex models it has to be remarked that when used with non convex models this will find local minimizers.

II. A GRAPH-BASED MODEL FOR CONVEX NLSP

In this section a graph topology for the Newton step method is proposed. For this purpose, let us base the discussion with the NLSP described by Eq. 1. This NLSP consists of N variables, L equality constraints, and M inequality constraints.

$$\begin{array}{rcl} \min_{z_i} & \sum_{i=1}^N f_i(z_i) \\ \text{st.} & g_l(\mathbf{z}) & = & 0, \\ & h_m(\mathbf{z}) & \leq & 0, \end{array} \qquad \begin{array}{r} l = 1, 2, ..., L \ (1) \\ & m = 1, 2, ..., M \end{array}$$

where $f_i(z_i)$ are non linear functions, furthermore, we assume they have second order derivatives. On the other hand, $g_l(z)$ are linear equations while $h_m(z)$ linear inequalities denoted as follows:

Proceedings of the World Congress on Engineering and Computer Science 2013 Vol I WCECS 2013, 23-25 October, 2013, San Francisco, USA

$$\sum_{l=1}^{L} a_{li} \mathbf{z}_{i} = r_{l}$$
$$\sum_{m=1}^{M} b_{mi} \mathbf{z}_{i} \leq s_{m}$$

In general, a NLP solver starts by building the Lagrangian given by Eq. 2.

$$\mathcal{L}(\mathbf{z}) = \sum_{i=1}^{N} f_i(z_i) + \sum_{l=1}^{L} \lambda_l g_l(\mathbf{z}) + \sum_{m=1}^{M} \mu_m h_m(x)$$
(2)

This is the base to implement the Newton step, whose formulation is given by Eq. 3 [1]:

$$H(\mathcal{L}(z))\Delta z = -\nabla(\mathcal{L}(z))$$
(3)

Table I, shows the involved elements to compute the Newton step for problem 1. Do notice we have introduced slack variables as well dual variables which are used to control the bounds of the decision variables. $\underline{\rho}_i$ and \underline{z}_i are used for the lower bound while $\overline{\rho}_i$ and \overline{z}_i are used for the upper bound of z_i

This model can be represented with the graph shown in figure 1.



Fig. 1. Proposed topology for the Newton step method

Each constraint is represented by a dual variable and a set of links which represent the linear terms within the constraint. The terms in the constraints are represented by links which join the primal variables with the dual variables. The only difference between equality contraints and inequality constraints is the kind of links used to build the linking structure. In the case of equality constraints, the linking structure will be active along the whole solution process. On the other hand, the linking structure for the inequality contraints will be active only when the constraint is binding. This is represented by the gray color given to the links belonging to inequality constraints as opposed to the black links which belong to equality constraints. In table I, the same criterion has been imposed in the elements of $H(\mathcal{L}(z))$. As it can be noticed, it is full of empty spaces and gray color, the first are long term sparsity patterns and the second ones are temporary sparsity patterns awaiting to be exploited.

angian $(2) \qquad \begin{array}{c} -\nabla \lambda_{i} & \theta \\ -\nabla \lambda_{i} & \theta \\$

Fig. 2. Subgraph for a primal variable (i.e. z_l).

III. SELF CONTAINED GRAPHS

An interesting fact when this kind of graphs is used is that the information to build and compute the gradient is contained in the graph topology. First, the case where the gradient for a primal variable, z_i , is analysed. The discussion will be focused on the subgraph delineated in figure 2. From table I, it is known that $\nabla_{z_i} = \nabla f_i(z_i) - \lambda_l - \mu_m - \underline{\rho}_i + \overline{\rho}_i$.

Figure 3 shows the gradient evaluation process for a primal variable. Let us suppose that every node has the value related to the variable which itself represents. Therefore, the node gradient evaluaton starts by taking into account the gradient information within the node which in this case would be $\nabla f_i(z_i)$, as shown in figure 3(a).

Then it starts to evaluate the portion of the gradient which is a function of the variables contained by the neighbours of z_i , as shown in figures from 3(b) to 3(e).

Now, let us turn the attention to the case where the gradient for a dual variable is to be found, λ_l in this case. The discussion will be focused on the subgraph delineated in figure 4.

Figure 3 shows the gradient evaluation process for a dual variable. From table I it is known that $\nabla_{\lambda_l} = r_l - z_{1l} - .. - z_{nl}$. As before, the gradient evaluation starts by taking into account the information contained within the node itself, in this case r_l , as shown in figure 5(a). Then the evaluation of the links attached to this node and the variables at the other extreme of the link is performed as shown in figures 5(b) and 5(c)

From the previous discussion, as there exist only dual variables and primal variables, the gradient for every variable can be derived straightforwardly from the graph topology. Therefore, the graph can be said to be self-contained as no external information is needed.

IV. KARUSH-KHUN-TUCKER CONDITIONS

The Langrange multipliers method define optimality conditions for equality constraints. However, many problems are defined in terms of inequality constraints defined by 4.

$$\min_{z} f(z)
st. g_{l}(z) = 0, \qquad l = 1, 2, ..., L \quad (4)
h_{m}(z) \leq 0, \qquad m = 1, 2, ..., M$$

Proceedings of the World Congress on Engineering and Computer Science 2013 Vol I WCECS 2013, 23-25 October, 2013, San Francisco, USA



THE NEWTON STEP INGREDIENTS

To face this problem, Karush-Khun-Tucker conditions (KKT) generalize the Langrange multipliers method defining a minimum set of conditions which guarantee the optimality conditions for non lineal programming problems with inequality constraints. Furthermore, KKT conditions provide sufficient optimality conditions for convex programming problems, as the one we are dealing with. If z^* is the optimal solution for a non lineal problem with N decision variables, L equality constraints and M inequality constraints, these conditions are [1]

$$\nabla f(\mathbf{z}^*) + \sum_{l=1}^{L} \lambda_l \nabla g_l(\mathbf{z}^*) + \sum_{m=1}^{M} \mu_m \nabla h_m(x^*) = 0 \quad (5)$$

$$g_l(\mathbf{z}^*) = 0, \qquad l = 1, 2, .., L$$
 (6)

$$h_m(\mathbf{z}^*) \leq 0, \quad m = 1, 2, .., M$$
 (7)

$$\mu_m h_m(\mathbf{z}^*) = 0, \quad m = 1, 2, .., M$$
 (8)

$$\mu_m \geq 0, \quad m = 1, 2, .., M$$
 (9)

Where Eq. 5 represents equilibrium between the gradients of the objective function and the active constraints. Equations 6 and 7 represent the feasibility of the solution in the optimum point z^* . Equation 8 represent the complementarity conditions (i.e. $\mu_m = 0$ or $h_m(z^*) = 0$). Finally, Eq. 9 represents dual feasibility.

Condition 9, establishes the Lagrange multipliers non negativity property. If the Lagrange multiplier was a negative, then z^* is within the feasible region, not at the boundary, and is even feasible to improve the objective function. Therefore, we can conclude that its corresponding constraint is not active anymore. This condition is exploited to discriminate the active from the non active parts of the graph.

V. GRAPH ANALYSIS

A node and the links which are attached to it represent an equation. In this section the analysis for a node and the equation it represents is done. To this end let us extract the equation corresponding to z_i from the system of linear equations which describes the Newton step. This is given by Eq. 10.

$$\frac{\partial^{2} \mathcal{L}(z)}{\partial^{2} z_{i}} \Delta z_{i} + \sum_{\forall j \in \Gamma_{i}} \frac{\partial^{2} \mathcal{L}(z)}{\partial z_{i} \partial z_{j}} \Delta z_{j} = -\nabla_{z_{i}} \mathcal{L}(z)$$
(10)

solving for Δz_i leads to

$$\Delta z_{i} = \frac{-\nabla_{z_{i}}\mathcal{L}(z) - \sum_{\substack{\forall j \in \Gamma_{i} \\ \overline{\partial z_{i}} \partial z_{j}}} \frac{\partial^{2}\mathcal{L}(z)}{\partial z_{i} \partial z_{j}} \Delta z_{j}}{\frac{\partial^{2}\mathcal{L}(z)}{\partial z_{i}}}$$
(11)

This can be rewritten as

$$\Delta z_{i} = \frac{-\nabla_{z_{i}} \mathcal{L}(z)}{\frac{\partial^{2} \mathcal{L}(z)}{\partial^{2} z_{i}}} - \sum_{\forall j \in \Gamma_{i}} \frac{\frac{\partial^{2} \mathcal{L}(z)}{\partial z_{i} \partial z_{j}}}{\frac{\partial^{2} \mathcal{L}(z)}{\partial^{2} z_{i}}} \Delta z_{j}$$
(12)

This expression can be thought as the improvement in the solution for the component in the orthogonal axis z_i . It can be split into two parts. The first part, described by expression 13, is a component which involves the gradient and, therefore, it is needed within any solution approach for the graph.

$$\frac{-\nabla_{z_i}\mathcal{L}(z)}{\frac{\partial^2 \mathcal{L}(z)}{\partial^2 z_i}} \tag{13}$$

This is the contribution based on $-\nabla_{z_i} \mathcal{L}(z)$ just like in the steepest descent methods. However the length of the step will be reinforced with the second order information provided by $1/\frac{\partial^2 \mathcal{L}(z)}{\partial^2 z_i}$. This will be the case for the primal variables, however for the dual variables there will not be second order information, and therefore the gradient step size will have to be controlled by some other means.

The second part, described by expression 14, is composed by all the second order contributions which will be collected by z_i from its neighbours (i.e. Γ_i).

$$-\sum_{\forall j \in \Gamma_i} \frac{\frac{\partial^2 \mathcal{L}(z)}{\partial z_i \partial z_j}}{\frac{\partial^2 \mathcal{L}(z)}{\partial^2 z_i}} \Delta z_j$$
(14)

This part has several of components which will allow us to formulate models which can go from taking into account the second order information from all the neighbors to the other extreme where no second order information from them will be collected at all. The first approach would be the full centralised Newton step and the second one would result in the steepest descent reinforced with the second order information for the same orthogonal axis. Nevertheless, between these two approaches there is a plethora of options which involves a different number of the components of second order information. In fact there are $|\wp(\Gamma_i)|$ choices and the choice at any point will impact the precision of the Newton step and therefore the convergence. Let us define \mathcal{L}_k as the set of links which are taken into account for this process, do notice $|\mathcal{L}_k| = k$. In fact k = 0 denotes the steepest descent reinforced with second order information for the same orthogonal axis whereas $k = |\Gamma_i|$ denotes the full Newton step. Finally, with this in mind expression 10 can be rewritten as expression 15

Proceedings of the World Congress on Engineering and Computer Science 2013 Vol I WCECS 2013, 23-25 October, 2013, San Francisco, USA





Fig. 3. Graph-based gradient evaluation for a primal variable (i.e. z_i).

$$\Delta z_{i} = \frac{-\nabla_{z_{i}}\mathcal{L}(z)}{\frac{\partial^{2}\mathcal{L}(z)}{\partial^{2}z_{i}}} - \sum_{\substack{\forall j \in \Gamma_{i} \\ (i,j) \in \mathcal{L}_{h}}} \frac{\frac{\partial^{2}\mathcal{L}(z)}{\partial z_{i}\partial z_{j}}}{\frac{\partial^{2}\mathcal{L}(z)}{\partial^{2}z_{i}}} \Delta z_{j}$$
(15)

VI. CONCLUSIONS

This document has presented a graph model proposal for convex non linear separable problems with linear constraints. It has been set the ingredients involved in the Newton step,

ISBN: 978-988-19252-3-7 ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



Fig. 4. Subgraph for a dual variable (i.e. λ_l).



Fig. 5. Graph-based gradient evaluation for a dual variable (i.e. λ_l).

which is based on the Lagrangian of the system. In this representation it has been notorious the sparsity patterns of both types long term and temporary. The first are patterns which are present at the beginning of every Newton step iteration, whereas the send ones are patters which can differ between each Newton step iteration. Then, it has been shown how to transit from the matrix model to the graph model, which reflects the sparsity already noticed in the matrix representation. Following, by analyzing the graph interconnections, it has been uncovered the self-containing characteristic whose basic mean is that no information beyond that involved in the graph is needed to compute the gradient. This allow us to compute the gradient directly from the graph, provided the correct information is attached to each node. Finally, it has been presented an analysis of the equation represented by the node and its links. This analysis uncovered different patterns to compute the Newton step with different precision levels ranging from the gradient oriented model reinforced with its proper second order information to the full Newton step model where all the second order information terms are taking into account. Furthermore, in between there is a plethora of models which can be derived which will allow more efficient models as well as decentralized models to face problems such as those in [2], [3], [4].

REFERENCES

- [1] I. Griva, S. G. Nash, and A. Soferr, *Linear and nonlinear optimization*. SIAM Society for Industrial and Applied Mathematics, 2009.
- [2] G. Cohen, "Optimization by decomposition and coordination: A unified approach," *IEEE Transactions on Automatic Control*, vol. 2, no. 2, pp. 222–232, 1978.
- [3] A. Bakirtzis, P. N. Biskas, N. Macheras, and N. Pasialis, "A decentralized implementation of dc optimal power flow on a network of computers," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 25–33, February 2000.
- [4] P. Biskas and A. Bakirtzis, "Decentralised opf of large multiarea power systems," *IEE Power, Generation, Transmission and Distribution*, vol. 153, no. 1, January 2006.