Quasi-Newton methods for large-scale linear programming

immediate

February 12, 2024

Abstract: In general, the application of general algorithms does not apply to solving large-scale linear programming. Currently, there are several researches in the development of special algorithms for certain groups of problems. Researchers present a variety of algorithms with special features for each one, by using different methods. In this thesis, a no-precise Newton algorithm for optimizing with equality constraints on large scales presents by Beard et al., is studied. Often the linear searching methods for non-convex optimization need Jacobean analysing of constraints or elementary-dual matrix. In this approach, those directions are accepted that sufficiently reduce topical approximation of a precise penalty function. This method is free of analysing any matrices. In the Mangasarian Newton method with an external penalty function, a no-limited non-linear optimization problem (a second-order segmented convex function) is formed that is equivalent to the main linear programming problem. Also shows that the minimizing problem of this function is the same optimization condition with the minimal norm of the dual problem. The result of optimizing this function is the result of dual and thereby, primal problems. In this method, the normal result (base) of the linear problem is obtained.

Keywords: Lagrangian method, Optimal Solution, Simplex method, Newton's method, Reduced Hessian, Mangasarian method.

1 Introduction

Today, the optimization concept is accepted as a fundamental principle in analysing many of complex decisions or allocation problems. By applying optimization concept, it would be possible to study on a complex decision problem includes determining values for many related variables by focusing on unique objective that is the direction of qualitative and quantitative measurement and evaluation of plan. This unique objective will be minimum or maximum considering the problem constraints and conditions. It would be possible to make a framework for analysing and study through optimization if can separate an aspect of problem or determine an objective (goal). This objective can be the benefit or loose in business, speed of distance in physics and etc. Therefore, can conclude that the proper formulation of problem is the first step or in other word, the most important step in optimization process. After formulation of model, an optimization algorithm can be used to find its answer. Unfortunately, there is no comprehensive optimization algorithm but there are many algorithms that each are proper for specific problems. Determining the proper algorithm is an important choice, because it determines the speed of solving the problem or either there is no answer. After applying an optimization algorithm for a model, we should be able to determine the success of the considered algorithm in finding the answer. In many conditions (states), in order to check that the current answer set is the same problem answer, there are conditions as optimum conditions. There are many studies on a large scale field also many researchers discover specific methods to analysis those problems. A group of those studies are related to solving linear problems in large scale. This thesis aims to study research in this field and develop a semi-Newton based algorithm. There are many algorithms to solve linear programming. By advancing computation technology, researchers are always seeking more efficient algorithms to solve linear programming. The Simplex method by Dantzig.B George in 1974 is introduced to solve the linear programming problems [7]. This method of solving linear problems is a valuable algorithm. Since the introduction of Simplex method, many modifications and editing have been applied to increase the efficiency of this method. After introducing of this method, the time complexity was the subject. Khachiyan 1979 presented an algorithm for solving the linear problem through oval method [2]. The results of Khachiyan are also used in hybrid optimization. For many years, there was not any correct inference of Khachiyan results.The Oval method was good enough in theory, but its computational results were not able to compete with Simplex algorithm. In 1974 Karmakar introduced a new group of algorithms named inner point. Its computational results showed that it is better than the Simplex method [1]. Since the introduction this method, there have been many modifications and adjustments to this method. Each iteration of inner-point is computationally costly but getting an optimum answer is satisfactory, in other hand, Simplex method has more iteration to reach optimum answer. These two methods are different in getting answers. The inner-point method moves from inside to outside of feasible region to get optimum answer on the border of feasible region. In Simplex method in each iteration, moves from each point on multidimensional feasible region to optimum answer. Theoretically,the Simplex method can solve any type of linear problem, but for large-scale problems,the above algorithm has weak performance and cannot solve this set of problems in reasonable time [4]. Many studies have been done in the field of solving linear programming where all developed algorithms are numerical algorithms. In large-scale, the performance of normal algorithms for solving linear problems are very different. Increasing variables and constraints leads to lack of memory for optimization algorithms and thereby, increasing computational time. Sometimes, the computation of the initial start point would be very difficult. Therefore, in recent decades, researchers have tried to introduce numerical optimization algorithms for solving linear programming problems.

This article is organized as follows. In the next section,in Armijo rule an approach for its linear search where $\bar{\lambda}$ frequently halving until reach the minimum integer $t \geq 1$ that $\theta\left[\frac{\bar{\lambda}}{2^t}\right] \leq \hat{\theta}\left[\frac{\bar{\lambda}}{2^t}\right]$. Therefore $\left[\frac{\bar{\lambda}}{2^t}\right]$ selects as step size. In section 3, is focus on for solving optimization problems, an algorithm is used that generates a sequence of points to converge towards a solution. The optimal solution is reached by this algorithm.Comprehensiveness, reliability, novelty, and sensitivity to parameters and data provide us with an efficient algorithm for addressing problems. In part 4 we explain about Newton's method on solve liner programming, the linear search process starts from an initial point and moves along it in the created direction towards the maximum or minimum of the objective function. At the new point, a new direction is set and the process is repeated. In general, after finding a new point, the process continues until reaching the stopping point. in Section 5, an external penalty function is described. Furthermore, penalty function-based methods transform a constrained problem into an unconstrained or a sequence of unconstrained problems. Constraints are incorporated into the objective function with a penalty parameter, such that violations of constraints result in a penalty. In section 6, we discuss the extended Newton method by Mangasarian where the reduction direction in Newton algorithm has high efficiency and the amount of function decline is considerable. Those set of imprecise Newton methods where approximate solution of Newton equations with some arbitrary methods are considered is derided in section 7. Moreover,in this section we elaborated on the concept of the descent condition, which aims to determine when the obtained decrease in a local approximation is sufficiently large for a given step. An imprecise Newton method based on model reductions is shown in section 8. Static tests 1 and 2 are the ones that should be fulfilled in the condition of the reduction model, and the stages present in this tests are the ones that are directly valid in the condition of the reduction model are presented in sections 9 and 10. In part 11, explained about Hessian modification strategy and when we use it that, w_k should be modified if and only if computed step is not holds in static steps 1 and 2. In section 12, an approximate algorithm INS using an unbounded matrix for solving constrained optimization problems has been presented in which method, approached delves into optimizing a local function based on stability tests for reduction. Additionally, the correction of the Hessian matrix in iterative calculation step has been proposed and demonstrated. Moreover, a global heuristic algorithm has been applied to first-order optimal points, and its effectiveness has been examined and tested on various problems. INS Algorithm is derived in section 13. Finally, Implementing imprecise Newton method for non-convex optimization with equality constraints and tables are presented in sections 14 and 15.

2 The Armijo rule

According to Armijo rule due to costly computation of function it is not possible to use a precise linear search. Therefore, the optimization methods normally use a sufficient precise linear search that guarantees the reduction in objective function. Assume that we want to minimize the differentiable function $f : R^n \to R^n$

in point $\bar{x}$ and d is a reduction direction. Define the linear search function $\theta : R \to R$ for $\lambda \succ 0$ as follows:

$$\theta(\lambda) = f(\bar{x} + \lambda d)$$

Then assume first order approximation θ in $\lambda = 0$

$$\theta(0) \neq \lambda\theta'(0)$$

$$\theta(\lambda) = \theta(0) + \lambda \bar{\varepsilon}\, \theta'(0)$$

where $0 \prec \bar{\varepsilon} \prec 1$. Step size $\bar{\lambda}$ is acceptable if

$$\theta(\bar{\lambda}) \leq \hat{\theta}(\bar{\lambda}) \tag{1}$$

Then, $\bar{\lambda}$ selects as step size and if $\bar{\lambda}$ does not satisfy(hold) the Armijo rule then would be an approach for its linear search where $\bar{\lambda}$ frequently halving until reach the minimum integer $t \geq 1$ that $\theta\left[\frac{\bar{\lambda}}{2^t}\right] \leq \hat{\theta}\left[\frac{\bar{\lambda}}{2^t}\right]$. Therefore $\left[\frac{\bar{\lambda}}{2^t}\right]$ selects as step size.

3 Algorithms

Algorithm means a solution of a problem as an iterative process that should have following three conditions:

1. be step by step.
2. each step be free of ambiguity.
3. has termination condition.

Mathematically, it is possible to show an ambiguity-free command by mapping. So, an algorithm is a mapping as:

$$A : X \to 2^x$$

$$x_k \to x_{k+1} \in A(x_k)$$

4 Newton method to solve linear programming

The developed Newton method by Mangasarian is presented and is convergent for the set of problems where the number of constraints is more than variables [6]. This method can find the answer with minimum norm for linear programming problem. This method uses of external penalty function of elementary problem with minimum norm for secondary problem of linear programming Assume that the primal problem is:

$$min\ c^T x \quad s.t \quad Ax \leq b \tag{2}$$

$$x \in R^n$$

where $A \in R^{m+n}, b \in R^m$ and $c \in R^n$. Consider that decision variables of this linear programming model are free-sign and secondary model of above programming model is:

$$\max_{u \in R^m} -b^T u = \min_{u \in R^m} b^T u \tag{3}$$

$$s.t \quad A^T u + c = 0, \quad u \geq 0$$

Here the parameter modelling of an external penalty function with a fixed penalty parameter ε as a minimum problem is

$$\min f(x)$$

$$x \in R^n$$

where f is a penalty function as:

$$f(x) = \min \varepsilon c^T x + \frac{1}{2}||(Ax - b)||^2 \tag{4}$$

$$x \in R^n$$

When the penalty parameter ε tends to zero, then the answer of equation (4) tends to the answer of linear programming [6]. If assume $\inf f_p$ as the largest limit of p if $\inf f_p$ is a real number, then the primal linear problem (also dual problem) has non-empty solution set. Now assume $L = \inf\{b^T v : A^T v + c = 0, v \geq 0\}$ as optimum answer of dual problem. We can get optimum answer of minimum norm for secondary problem as:

$$\min \frac{1}{2} v^T v \tag{5}$$

$$s.t \quad b^T u = L, \quad A^T v + c = 0, \quad v \geq 0$$

Normal and standard method to find optimum answer of minimum norm for convex programming is based on Tikhonov simplification.This simplification creates v_k sequence where v_k is the answer of following programming [5]

$$-\min(b^T v + \frac{\varepsilon}{2} v^T v) \tag{6}$$

$$s.t \quad A^T v + c = 0, \quad v \geq 0$$

Optimum answer of minimum norm for secondary problem for each ε with $\varepsilon \in [0, \bar{\varepsilon}]$ is from the set of dual answers as objective function of this problem is convex then the answer of this problem $\bar{v}$ is unique [3, 5]. The necessity and sufficiency conditions Karush-Kuhn-Tucker for model(4) states that there is $y \in R^n$ if:

$$\varepsilon v \geq 0, \quad \varepsilon v - b - Ay \geq 0, \tag{7}$$

$$\varepsilon v^T(\varepsilon v - b - Ay) = 0, \quad A^T v + c = 0$$

In other words:

$$\varepsilon v = (Ay - b)_+, \quad A^T v + c = 0 \tag{8}$$

According to (8) we can conclude that:

$$v = \frac{1}{\varepsilon}(Ay - b)_+,$$

$$A^T(Ay - b)_+ + \varepsilon c = 0.$$

By defining $f(y)$ in form of (4) optimization conditions for equation (5) for minimum norm of dual problem is :

$$v = \frac{1}{\varepsilon}(Ay - b)_+, \tag{9}$$

$$\nabla f(y) = A^T(Ay - b)_+ + \varepsilon c = 0.$$

Or it can be said in a more complete method

$$v = \frac{1}{\varepsilon}(Ay - b)_+ \tag{10}$$

$$y \in \arg f(y) = \arg\min \varepsilon c^T y + \frac{1}{2}||(Ay - b)_+||^2 \tag{11}$$

$$y \in R^n$$

Precisely is the necessity and sufficiency condition for positivity of minimum answer of external penalty function $f(y)$ for primal linear programming (2) for each penalty parameter ε.

5 External penalty function.

Unique answer for minimum norm of dual problem in linear programming (2) obtains from

$$v = \frac{1}{\varepsilon}(Ay - b)_+, \tag{12}$$

Where the following optimization problem is the answer of primal basic problem:

$$\min f(y) = \varepsilon c^T y + \frac{1}{2}||(Ay - b)_+||^2 \tag{13}$$

$$y \in R^n$$

Consider that function gradient is as follows:

$$\nabla f(y) = A^T (Ay - b)_+ + \varepsilon c \tag{14}$$

We can define generalized Hessian with many features of basic Hessian. Following Hessian matrix is positive sub semi-definite

$$\partial^2 f(y) = A^T diag(Ay - b)_* A$$

Where $diag(Ay - b)_*$ refers to a diagonal matrix $m \times m$ with diagonal arrays $A_i y - b_i$. step function $(.)_*$ is defined in introduction. Matrix $\partial^2 f(y)$ is used to produce Newton directions. Generally, use Newton method for determining optimum value of differentiable and continues functions. Newton reduction direction of second order expansion is obtained from Taylor series:

$$f(x_k + d) = f_k + d^T \nabla f_k + \frac{1}{2} d^T \nabla^2 f_k d = m_k(d) \tag{15}$$

If assume that $\nabla^2 f_k$ is positive-definite, by selecting vector d that minimizes the $m_k(d)$), Newton direction obtains. By putting zero the derivative of function $m_k(d)$, Newton direction in kth iteration (d_k^n) would be defined as follow:

$$(d_k^n) = -\nabla^2 f_k^{-1} \nabla f \tag{16}$$

When the difference between function $f(x_k+d)$ and its second order model $m_k(d)$ is not big, Newton direction is most trustworthy. Comparing above results and Taylor theorem shows that only difference between these is in third sentence where matrix $\nabla^2 f_k$ is replaced with $\nabla^2 f(x + t_d)$.

We can use Newton direction in time linear search where $\nabla^2 f_k$ is positive-definite. We will have:

$$\nabla f_k^T d_k^N = -{d_k^N}^T \nabla^2 f_k d_k^N \leq -\sigma_k ||d_k^N||^2 \tag{17}$$

$-\sigma_k$ is in fact Armijo step size in kth iteration. In order to use Newton direction to solve non-constraint optimization problem (11) we should apply the same function.

d d is the adjusted Newton direction:

$$d = -(\delta I + \partial^2 f(y))^{-1} \nabla f(y) \tag{18}$$

Where δ is a small positive number. This adjust ability is due to making possible to use this matrix when Hessian matrix is singular. By using this direction and the size of different steps, we can proof the global convergence. The critical practical computational point about this direction is global convergence for a class of linear programming problems where the number of constraints is much more than the number of its variables. Its start point is arbitrary and is free of computing the step size. Precise answer of minimum norm $\bar{v}$ for dual linear programming is computed by primal problem of external penalty for $\varepsilon \in [0, \bar{\varepsilon}]$ and then by using (10) unique answer of minimum norm $\bar{v}$. We can use the precise answer of dual to obtain the answer of primal linear programming problem. So we must solve equally, the set of linear programming equations related to positive arrays $\bar{v}$.

$$A_j z = b_j, j \in S, S = \{j | \bar{v}_j \succ 0\} \tag{19}$$

It should be considered that this linear equations system always has answer based on the result of auxiliary conditions of equation (5)

$$A_j z - b_j = 0, \bar{v_j} \succ 0\} \tag{20}$$

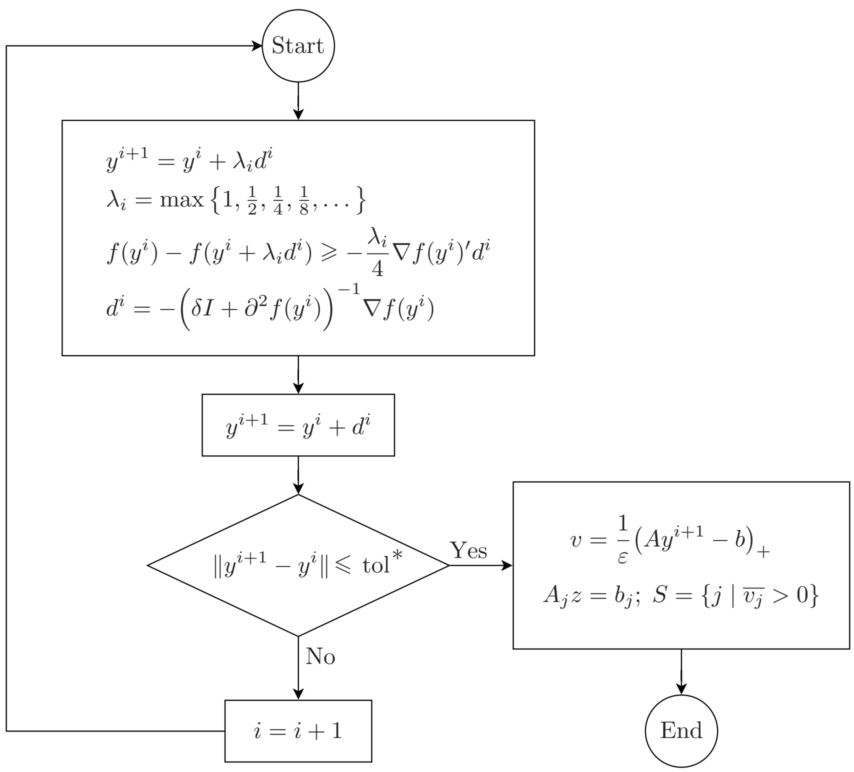
In fact, leads to obtain the answer of primal linear programming problem by considering following assumption:

The columns of matrix A_k are linear-independent. So this presumption that gradually points to uniqueness of primal optimum answer is sufficient but not necessary to produce primal answer.

6 Newton algorithm to solve linear programming problems

The presented algorithm includes solving the problem (11) for approximate answer y, computing precise answer minimum norm $\bar{v}$ for dual linear programming from (18) and finally, computing answer z for primal programming problem is from (2). Equation (5) is used to guarantee the global convergence, Armijo step size. Also, we should consider the linear independence assumption for precise answer of minimum norm of linear programming problem.

The steps of Newton algorithm to solving linear programming problem is:



* (tol is considered to 10^{-4}, 10^{-1} and 10^{-12})

Figure 1: Newton Algorithm.

In this paper we discuss the extended Newton method by Mangasarian where the reduction direction in Newton algorithm has high efficiency and the amount of function decline is considerable (11) Another advantage of Newton algorithm is stabling the step size of reduction direction (18)that prevents computing the value of objective function (13) and minimizes the number of approximations for objective function gradient. The minimum reduction direction of Newton leads to minimizing the time of solving problem and the number of iterations to obtain optimum answer. But the Newton algorithm is sensitive against start point and the convergence speed of this method is efficient when start point is sufficiently close to the optimum point.

7 An imprecise Newton method for non-convex optimizing with equality constraint

As stated before, Newton is a classic algorithm for solving a non-linear system of equation that in large scale, computing the exact answer of Newton system (if existed) by using the Gauss removal method, is costly and so using the repetitive solving method to approximately solve the Newton system become reasonable. Therefore, imprecise Newton methods presented by Styhawke and Co-workers where in each iteration, needs approximate answers of Newton equation. In this chapter, we describe imprecise Newton methods and shows that how use them to solve non-linear system of equations; in under study approach in this chapter, those directions are accepted that make sufficient reductions in topical approximation of a precise penalty function. This approach uses repetitive solution methods of generalized minimum residuals GMERS for solving initial-dual systems. Although, we can use some other repetitive methods [8].

7.1 Introducing imprecise Newton methods

Initially we will give a brief description about the imprecise Newton method. A distinctive feature of Newton method is that based on any initial proper and sufficient guess, would be converged and thereby, a linear system in each time iteration with large variables could be costly. Therefore, those set of imprecise Newton methods where approximate solution of Newton equations with some arbitrary methods are considered [9,10]. those set of imprecise Newton methods where approximate solution of Newton equations with some arbitrary methods are considered is devrived in section 6.

7.2 Reduction condition of model

Assume that $\theta \geq 0$ and (d_k, δ_k) is an imprecise answer for following equation

$$\begin{bmatrix} w_k & A^T(x_k) \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g(x_k) + A(x_k) \\ c(x_k) \end{bmatrix} \tag{21}$$

if and only if $0 \leq \sigma \leq 1$ and for proper π_k we have:

$$\Delta m_k(d_k, \pi_k) \geq \max \left\{ \frac{1}{2} d_k^T w_k d_k, \theta r \right\} + \delta \pi_k \max\{||c_k||, ||r_k|| - ||c_k||\} \tag{22}$$

and r_k is defined according to the following relation. Where r_k is defined in

$$\begin{bmatrix} w_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix} \tag{23}$$

Finally, we present important results that is critical for our analysis in convergence section. If direction derivative of penalty function $\rho(x_k, \pi_k)$ in x_k and in $d_k$direction presents with $D\rho(d_k, \pi_k)$ then we have

$$\begin{aligned} D\rho(d_k, \pi_k) \quad &\leq \quad g_k^T d_k - \pi_k(||c_k|| - ||r_k||) \\ &= -\Delta l_k(d_k) - \pi_k \Delta ||r_k(d_k)|| \\ &= -\Delta m_k(d_k, \pi_k) \end{aligned} \tag{24}$$

So, a step that satisfies (holds) model reduction condition would be the reduction direction for penalty function $\rho(x_k, \pi_k)$ in x_k you can see [11].

8 An imprecise Newton method based on model reductions

In this section, a complete algorithm for non-convex optimization based on sufficient reduction in topical approximation of a merit function is described. First, a test step (d_k, δ_k) is computed that satisfies (holds) dual-primal equation (21). if the condition of reductive model satisfies (holds) for maximum recent value in penalty parameter π, then call it π_{k-1} and consequently accept this step and assume that $\pi_{k-1} \Rightarrow \pi_k$ and continue the iteration, else, due to obtaining a reduction in constraint model (see [12]), consider the increasing of π for fulfill the condition of reductive model. Any way this should be done in both states. In other hand, if $\frac{1}{2}d_k^T w_k d_k \geq \theta r_k$ then assume that the problem if sufficiently convex and therefore,$(d_k, \delta_k)$is acceptable and π must be increased. Also, if $\frac{1}{2}d_k^T w_k d_k \leq \theta r_k$ then we just consider the increasing of π if v_k be a considerable part of d_k (like when normal step in sufficient). Therefore, condition $r \leq \varphi r_k$ states for some $\varphi \succ 0$ where:

$$0 \leq v_k \leq ||v_k||^2 \tag{25}$$

(Where every lower band for square norm is from normal step component). If any above mentioned approaches for trust in properness of reductive model condition for every $\pi_{k-1} \prec \pi_k$ are not efficient, then we will have no other choice unless we have changes in step computations by modifying w_k. When such a modification is applied, we compute a new test step by modifying dual-primal equations and iterate the above steps to acceptable level. In imprecise state, we merge this process with added constraints for residual in (23) Then we introduce static tests where we can introduce each two tests as a generalization of static test approximation reduction of merit matrix (in brief, SMART Tests) [13].

9 Static test 1

Assume $\sigma \leq 0$, $k \prec 1$ and θ is in reduction model condition. Step (d_k, δ_k) is acceptable if reductive model condition satisfies $\pi_{k-1} = \pi_k$ and

$$\begin{bmatrix} \rho_k \\ r_k \end{bmatrix} \leq k \parallel \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} \parallel \tag{26}$$

Steps in static test 1 are the same steps that directly satisfies(holds) reductive model condition. There is no need to change penalty parameter or modify the step computation (we will describe in next sections).

10 Static test 2

Assume $0 \leq \varepsilon \leq 1, \beta \geq 0$ and φ and θ is in reduction model condition. step (d_k, δ_k) is acceptable if:

$$||r_k|| \leq \varepsilon ||c_k||, \tag{27}$$

$$||\rho_k|| \leq \beta ||c_k|| \tag{28}$$

And

$$\frac{1}{2} d_k^T w_k d_k \geq \theta r_k \tag{29}$$

or

$$\varphi v_k \geq r_k$$

Satisfies (holds) for r_k, (ρ_k, r_k) and v_k that are defined by (23),(25) Steps that satisfy(hold) the static test 2 do not necessarily satisfy(hold) the reductive model condition for $\pi_{k-1} = \pi_k$

In order to update penalty parameter, we need following:

$$\pi_k \geq \frac{g_k^T d_k + \max\{\frac{1}{2}d_k^T w_k d_k, \theta_r\}}{(1-r)(||c_k|| - ||r_k||} \cong \pi_k^{trial}$$

Where $0 \prec r \prec 1$. Considering this inequality and through equations (27) shows that:

$$\begin{aligned} \Delta m_k(d_k, \pi_k) \quad \geq \quad & \max\left\{\frac{1}{2}d_k^T w_k d_k, \theta r\right\} + r\pi_k(||c_k|| - ||r_k||) \\ & + \max\left\{\frac{1}{2}d_k^T w_k d_k, \theta r\right\} + r(1-\varepsilon)||c_k|| \end{aligned} \tag{30}$$

Therefore, this step satisfies (holds) reductive model condition for $\sigma = r(1-\varepsilon)$. In fact, to balance static test 1 and static test 2, we assume that $\sigma = r(1-\varepsilon)$ is selected and finally, Hessian modification strategy implements to modify the step computations. A modification technique w_k is increase some or all especial values to close resulted matrix to positive definite. For example, if f includes a regression parameter then this parameter can raise. w_k can replace with positive half-definite semi-Newton approximation or a positive half-definite added to w_k to get final matrix named $\widetilde{w_k} - w_k \succ \mu l$ that holds for some $\mu \succ 0$.

11 Hessian modification strategy

Assume that w_k is current Hessian approximation and θ, σ *and* φ in sufficient reductive model and static test 1 and 2 and (d_k, δ_k) given test step if (d_k, δ_k) holds in following equation:

$$\begin{aligned}\Delta m_k(d_k, \pi_{k-1}) \quad \leq \quad & \max\left\{\frac{1}{2}d_k^T w_k d_k, \theta r\right\} + \sigma\pi_{k-1}\max\{||c_k||, ||r_k|| - ||c_k||\} \\ & + \left\{\frac{1}{2}d_k^T w_k d_k\right\} \leq \theta r_k \ and \ \varphi v_k \leq r_k \end{aligned} \tag{31}$$

Therefore, w_k is modified otherwise, we will keep current w_k. (Also, after definite number of modifications, $w_k \geq 2\theta$) is assumed. This update is possible by viewing a state where primal-dual equation is solved directly (21). Like this state, w_k should be modified if and only if computed step is not holds in static steps 1 and 2. In this state, it is clear that $(\rho_k, r_k) = 0$ and therefore (26) and (27) are held. Anyway, it should be considered that under these conditions of strategy, definite number of modifications during k^{th} iteration are feasible and $w_k \geq 2\theta I$ shows that $||v_k|| \leq y_2(||c_k|| - ||r_k||)$ does not hold. Finally, to compute acceptable step, we have a reversible searching route on merit function $\phi(x, \pi_k)$ where step size α_k in following Armijo condition holds for some $\eta \in (0, 1)$.

$$\phi(x_k + \alpha_k d_k, \pi_k) \leq \phi(x_k, \pi_k) + \eta\alpha_k D(d_k, \pi_k) \tag{32}$$

$$||g_k + A_k^T\lambda_k||_\infty \leq 10^{-6}\max\{||g_0||_\infty, 1\} \tag{33}$$

12 Computing step of INS algorithm

Assume that $w_k = \nabla_x^2.L_k, \quad (d^0, \delta^0) = 0, \quad j = 0$ and loop starts. $\mu = 10^{-4}$

$$while \ \ j \leq n + t$$

Then put,

$$j = j + 1$$

- Run iteration GMARES from equation (21) to compute (d^j, δ^j).
  - from $\frac{||A_k d||^2}{||A_k||^d} \to v$ and $||d||^2 - \frac{||A_k d||^2}{||A_k||^d} \to r$, compute r^j and v^j.
  - if static test 1 and static test 2 are held, stop.
  - If (d^j, δ^j) holds in $||v_k||^2 \leq y_1 \ \ \max\{||c_k||, ||r_k||\}$, $||v_k||^2 \leq y_2\{||c_k|| - ||r_k||\}$.

  and $v_k \leq ||A_k^T(A_k - A_k^{T-1})||(||c_k|| + ||r_k||)$. Then we assume that $j = 0, \ w_k = w_k + \mu I, \ (d^0, \delta^0) = (d^j, \delta^j)$, $\mu = 0$ and loop terminates.

  - We have also a return such that $(d^j, \delta^j) \to (d_k, \delta_k)$, and $(r^j, v^j) \to (r_k, v_k)$.

13 INS algorithm

This algorithm is showed as a flowchart in figure 2.

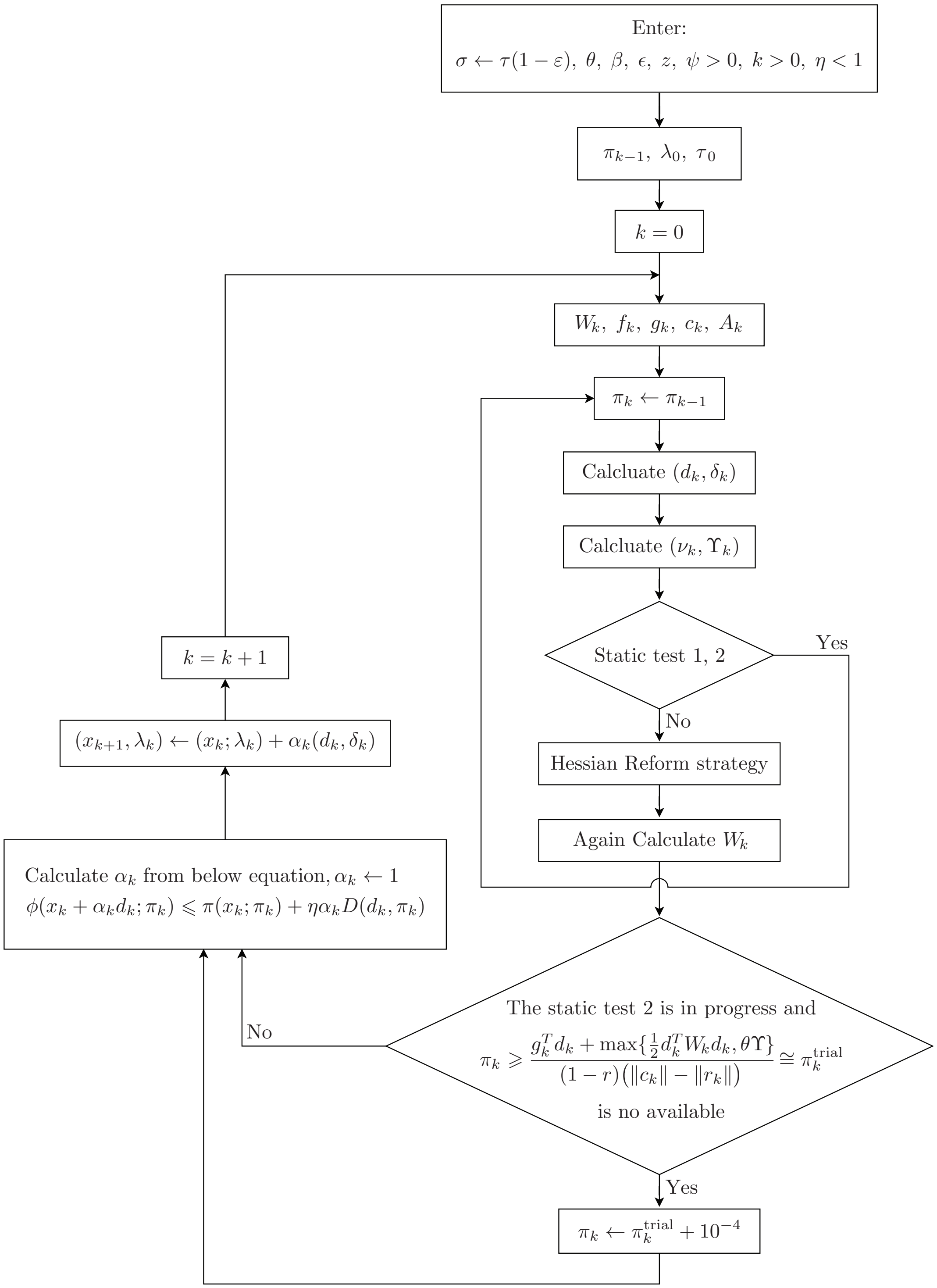


Figure 2: INS Algorithm

14 Implementing imprecise Newton method for non-convex optimization with equality constraints

INS algorithm is presented in MATLAB software. Here we noted some issues about implementing algorithm and time order of running its steps and then describe its performance on different sets of experimental problems. A general guide for selecting parameters $k, \varepsilon, \beta, \tau$ *and* σ is given in [11]. Generally, parameters k, ε, β can have considerable effect on algorithm convergence speed and preferably should be implemented separately for each program. For instance, see[14]. For INS algorithm residuals entrance, do the following selections. First, do w_k as precise Hessian Lagrange for all k and then an approximate value θ,during each iteration recommended as:

$$10^{-8} \max\{||w_k||, 1\} \rightarrow \theta. \tag{34}$$

Our acceptable conditions are independent from f, c. For ψ ,approximate value is considered between 0 and 1. A step is sufficient step if $||u_k||^2$ is equivalent to $||v_k||^2$. Estimated values are applied for these data (values). Table 3 shows a complete list of parameters in our implementation.

Also, we need the two following equations for stop criteria of external loop of INS algorithm, 2.

$$||c_k|| \leq 10^{-6} \max\{||c_0||_\infty, 1\} \tag{35}$$

This algorithm will terminate if this equality does not hold after 1000 iteration. Following, a simple technique is presented to compute lower and upper bounds tangential component u and normal component for primary step d during iteration k. in fact we have:

$$||d||^2 = ||u||^2 + ||v||^2 \tag{36}$$

When $A_k u = 0$ then we have following inequality

$$||v|| \geq \frac{||A_k u||}{||A_k||} = \frac{||A_k d||}{||A_k||} \tag{37}$$

According to

$$\begin{aligned} min \quad & q_k(d) \cong f(x_k) + g(x_k)^T + \frac{1}{2} d^T w_k d \\ & s.t \qquad r_k(d) \cong c(x_k) + A(x_k)d = 0 \end{aligned} \tag{38}$$

we have

$$||u||^2 \leq ||d||^2 - \frac{||A_k d||^2}{||A_k||^2}. \tag{39}$$

Therefore, considering (37) and (39)we can use the following equation

$$\frac{||A_k d||^2}{||A_k||^2} \rightarrow v. \tag{40}$$

and

$$||d||^2 - \frac{||A_k d||^2}{||A_k||^2} \rightarrow r \tag{41}$$

That is lower bound for $||v||^2$ and upper bound for $||u||^2$. Then by computing step of INS algorithm introduce it as iteration loop in algorithm.

15 Conclusion

More complex linear programming models and the application of mathematical models to the real world have increased the scale of linear programming problems. Therefore, designing algorithms that are able to obtain the answer to a linear programming problem with higher quality and less time is expanding. Many algorithms are recommended in this field that, according to their structure, have challenges in solving some large-scale linear programming problems, while these algorithms theoretically are rich. Computation occupies much memory when applying numerical algorithms on a large scale to solve mathematical programming problems. Also, it holds for non-linear programming. Selecting and designing algorithms and determining the linear searching path depends on implementing algorithms, and determining parameters could result in improvements and upgrades in performance, quality of answer, and time. This paper discusses a Mangasarian Newton method-based algorithm and, in order to upgrade algorithm performance, uses an imprecise Newton method for non-convex optimization with equality constraints free of matrix analysis. Comparing computational results shows that the new recommended algorithm has higher performance than other methods. Computational results for small-scale problems have the same Newton algorithm performance, but for large-scale problems, the imprecise Newton method can solve problems much faster than other methods. The recommended method not only can highly efficiently solve the problems that the Newton algorithm, due to its lack of memory, is not able to solve, but also has higher speed and performance, so it can solve linear programming in large-scale and higher dimensions.

References

[1] Karmarkar.N.K, 'A New polynomial-time Algorithm for Liner Programming, combinatorica, 4: 373-395, 1985.

[2] Khachiyan. L.G.,' A polynomial Algorithm for liner programming', Soviet Math. Dokl, pp.191-194, (1979).

[3] Richard H.Byrd. Frank E. Curtis. Jorge Nocedal An inexact Newton method for nonconvex equality constrained optimization Math program, pp.122,273, 2010.

[4] Eislet.H.A,C-L. Sandblom,'Liner programming and its Applications', springer-verlay Berlin,Heidelberg, pp. 379-394, 2007.

[5] Mangasarian O.L and Meyer.R.R Nonliner perturbation of Liner programs. SLAM jornal on Control and optimazation. vol. 21, pp. 1-31, 2018.

[6] Mangasarian. O.L.A Newton Method for liner programming Computer Science department university of Wisconsin 1210 West Dayton Street Madison , pp. 161-180, 2001.

[7] Dantzing,George, liner programming and extensions. Princeton university press and the RAND Corporation, 1963.

[8] Saad,Y., Schultz, M.H.GMRES: a generalized minimal residual algorithm for solving nonsymmetric liner systems. SIAM J.SCI.Stat. Comput., pp. 856-869, 1986.

[9] Alsattar,H., Zaidan,A., Bahaa,B.(2020).Novel meta-heuristic bala eagle search optimisation algorithm. Artificial intelligence Review, 53, 2237-2264.

[10] Bazaraa. M.S, H.D.Sherali, C.M.Shetty. nonliner programing. Wiley- Interscience [John Wiley Sons], Hoboken, NJ, 3rd, 2016. Theory and algorithms.

[11] Byrd, R.H.,Curist, F.E., Nocedal, J.:An inexact SQP method for equality constrained optimization-SLAM J.Optim.19(1),351-369 (2008).

[12] Birgin, E.G., Martínez, J. M. (2020) Complexity and performance of an augmented Lagrangian algorithm. Optimization Methods and Software. doi: 10.1080/10556788.2020.1746962.

[13] Al-Dabbagh R, Neri F, Idris N, Baba M (2018) Algorithmic design issues in adaptive differential evolution schemes: review and taxonomy. Swarm Evol Comput 43:284–311.

[14] D. Li, D. Zhu, "An affine scaling interior trust-region method combining with line search filter technique for optimization subject to bounds on variables," Numerical Algorithms, vol. 77 no. 4, pp. 1159-1182, DOI: 10.1007/s11075-017-0357-2, 2018.