# 1    Introduction

Solving constrained non linear optimization problems begins with identification of the Karush-Kuhn-Tucker conditions due to the fact that for a differentiable convex system where constraint qualification holds, the solution of NLP is the solution to the KKT system and vise versa. The KKT conditions enables solving the NLP model as an MCP model by representing the KKT conditions as complimentarily equations. Thus, it is often advantageous to explicitly model the KKT conditions as solve them as a Mixed Complementarity Problem (MCP), as described in the PATH Solver Manual (/link). The example provided is of a Linear Transportation model where the traditional model of fixed demand and price is made more realistic by making the variables endogenous, i.e. the price of commodity is market affects the demand of the commodity. However, at times the complexity of the model causes errors in formulation of KKT conditions, leading to incorrect or infeasible solutions. Finding the source of the error can be a particularly tedious task. Thus, in this article, we propose a systematic method for identification of errors in the explicit KKT conditions for solving an NLP. .

We begin by understanding the definition of complementarity. If a function $F(z) : R^n \mapsto R^n$, lower bounds $l \in R \cup -\infty^n$ and upper bounds $u \in R \cup \infty^n$ has a solution vector $z \in R$ such that for each $i \in 1, ..., n$, one of the following three conditions hold :

$$F_i(z) = 0 \text{ and } l_i \leq z_i \leq u_i \text{ or}$$
$$F_i(z) > 0 \text{ and } z_i = l_i \text{ or}$$
$$F_i(z) < 0 \text{ and } z_i = u_i$$

then function $F$ is complementary to the variable $z$ and its bounds. This is written in compact form as

$$F(z) \perp L \leq z \leq U$$

Where the symbol $\perp$ means "perpendicular to". From point of view of optimization, if $F(z)$ is non zero (non-binding constraint), changes in $z$ may further optimize the objective function until the constraint becomes binding. If $F(z)$ is binding, no changes in $z$ would further enhance the objective function, causing the marginal to be zero.

Thus for a general NLP problem,

$$
\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & g_i(x) \leqslant 0 & i = 1, 2...m \\
& h_k(x) = 0 & k = 1, 2...p \\
& d_l(x) \geqslant = 0 & l = 1, 2...q \\
& L \leq x \leq Z \\
& f(x) : R^n \mapsto R, g(x) : R^n \mapsto R^m \\
& h(x) : R^n \mapsto R^p, d(x) : R^n \mapsto R^p
\end{aligned}
\tag{1}
$$

the Lagrange function and KKT conditions in the complementarity form are written as

$$
\begin{aligned}
& L(x, u, v, w) = f(x) - <u, g(x)> - <v, h(x)> - <w, d(x)> \\
& \bigtriangledown_x L \perp L_x \leq x \leq U_x \\
& -\bigtriangledown_u L \perp u \geq 0 \\
& -\bigtriangledown_v L \perp v free \\
& -\bigtriangledown_w L \perp w \leq 0
\end{aligned}
\tag{2}
$$

The gradient w.r.t to marginals result in the inequality and equality constraints of the NLP.

Thus an NLP can be solved as an MCP using the KKT conditions. However, formulating the KKT conditions might prove difficult for complex systems, and need verification of the accuracy. In the following sections, we provide the framework for modeling the KKT conditions by using concept of dummy complementarity equations in the KKT formulation. The process includes the following steps:

1. Solve NLP without explicit KKT conditions, and save the results

2. Restart the problem as an MCP with dummy KKT conditions using solution from step 1 as initial point. The MCP should start at the solution itself

3. Replace one dummy equation with one KKT condition. Resolve the NLP and save the results

4. Restart the problem as MCP. If MCP iteration count is ¿ 0, there exists a problem with the KKT condition which was introduced.

Follow steps 3 and 4 until all KKT conditions have been successfully incorporated.

# 2    Example : Maximum Revenue- NLP formulation

Consider a simple case of a steel factory trying to maximize the revenue under budget constraints, with man-hours(h) and raw materials steel (s) as the decision variables. The revenue is a function of decision variables given by

$$R(h, s) = 200h^{(2/3)}s^{(1/3)}$$

where, budget = $ 20,000 cost of manpower = $ 20 /hr cost of raw material = $ 170 / tonn

In it's standard form, the model can be written as :

$$\min \quad R(h,s) = -200h^{2/3}s^{1/3}$$
$$\text{s.t.} \quad 20h + 170s \leq 20000 \tag{3}$$
$$L < h, s < U$$

The GAMS program below gives the optimum values of

```
Maximum Revenue, R -51854.82
Man hours, h 666.67
Tons of raw material, s 39.22
```

```
$ontext
The model below showcases using GAMS to identify / modify the KKT conditions for a given
NLP formulation. Our starting point is a basic minimization NLP model.
$offtext

scalar
    labor 'cost of labor dollar per hour' /20/
    steel 'cost of steel per ton' /170/
    budget 'total budget for production' /20000/
    ;
variables
    h 'man hours in production' /lo 1, up 50000/
    s 'tons of raw material' /lo 1, up 50000/
    R 'Revenue'
    ;
Equations
    con1 'constraint on budget'
    obj 'objective function'
    ;
obj.. R =e= - 200 * h**(2/3) * s**(1/3) ;
con1.. 20*h + 170*s =l= budget;

h.l=10;
s.l=10;

model khan /con1,obj/;
solve khan using NLP minimizing R;
```

The above model can we written in form of an MCP, by explicitly adding the KKT conditions to the model. For the given system, the KKT conditions are given as

$$L = -200h^{2/3}s^{1/3} - con1_m[20h + 170s - 20000]$$
$$\nabla_h L = -200 * (2/3)h^{(-1/3)}s^{(1/3)} - con1\_m * (20)$$
$$\nabla_s L : -200 * (1/3)h^{(2/3)} * (1/3) * s^{(-2/3)} - con1\_m * (170) \tag{4}$$
$$\nabla_{con1_m} L : 20 * h + 170 * s - 20000 = 0$$

The third KKT equation is the inequality budget constraint itself. The above equations are solved as an MCP model as shown below.

Execution of the model with KKT conditions provide the values of decision variables given by

```
Maximum Revenue, R -51854.82
Man hours, h 666.67
Tons of raw material, s 39.22
```