OPTIMUM DESIGN OF BUMPERS FOR IMPACT APPLICATION USING KKT, SLP, CSD AND PLBA ALGORITHMS AGAINST IMPACT

Reza HEDAYATI[1], Meysam JAHANBAKHSHI[2]

Elastic energy absorbers which consist of ring-liked plate and springs can be a good choice for increasing the impact duration during an accident. In the current paper, an energy absorber system is optimized using four optimizing methods Karush-Kuhn-Tucker (KKT), Sequential Linear Programming (SLP), Constrained Steepest-Descent (CSD), and Pshenichny-Lim-Belegundu-Arora (PLBA). Time solution, convergency, programming length and accuracy of the results were considered in order to find the best solution algorithm. Results showed the superiority of PLBA and modified CSD over the other algorithms.

Keywords: Sequential Linear Programming (SLP); Constrained Steepest-Descent (CSD); Pshenichny-Lim-Belegundu-Arora (PLBA)

1. Introduction

The energy absorber system in this study consists of 4 rings, four springs and a bumper. A schematic view of this system is shown in Figure 1. Two elastic rings riveted to the bumper are connected to the two corresponding rings riveted to the car body by means of springs. When a car impacts an external obstacle, the distance between the bumper and the car body decreases which leads to compaction of the rings and the springs. The energy absorbed by the rings and springs increases the impact duration which decreases the peak force applied to the car and passengers.

Dynamic mechanical optimization problems involve integration of material and mechanical parameters as to determine the response of the system to external inputs [1]. Then, using the response variables, cost and constraint functions for the problem are formulated. These constraints are implicit functions of the design variables. In mechanical and structural design problems, various state variables, such as the displacements, dimensions and velocities, are treated as independent variables. The kinetic and internal energies become equality

[1] PhD, Young Researchers and Elite Club, Najaf Abad Branch, Islamic Azad University, Najaf Abad, Isfahan, Iran, Email: rezahedayati@gmail.com
[2] PhD, Young Researchers and Elite Club, Najaf Abad Branch, Islamic Azad University, Najaf Abad, Isfahan, Iran.

constraints in the formulations. Several possible ways to parameterize these variables have been investigated by Wang and Arora [2-4].

In the current paper, an energy absorber system is optimized using four optimizing methods namely KKT, Sequential Linear Programming (SLP), Constrained Steepest-Descent (CSD), and Pshenichny-Lim-Belegundu-Arora (PLBA). The reviews of the applications of the optimization algorithms can be found under their related subsection in the following.

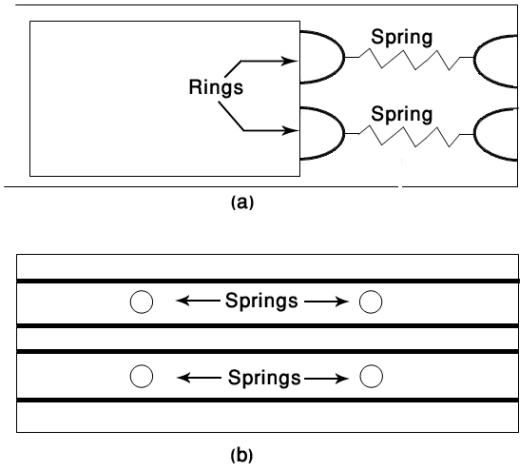

Fig. 1. A schematic view of energy absorber system in the bumper (a) Side view (b) Front view

2. Constraints and Objective Function of the problem

The objective function of the problem is the price of the energy absorber. For calculating the total price, the relationship below can be used [5]:

$$F = 9.81\rho\pi rtL \times C_1 + C_2 \tag{1}$$

in which C_1 is the price of steel per unit volume and C_2 is the price of the four springs. Using a steel type of SAE 960X which has a yield stress of 413 MPa and costs 65\$/lb [5], and considering the price of springs as $0.1\sqrt{k}$, the objective function can be stated as:

$$f(x) = 9.81(7800) \times 100\pi rtL + 0.4\sqrt{k} \tag{2}$$

In order to avoid spring flatting, the inequality below must be governed:

$$\overline{\varphi}_1 = 2(r_0 - \sqrt{2r_0}\,\sin(5^\circ) - \delta \geq 0 \tag{3}$$

For the car considered, it is usually seen that after an accident, the frontal part of the car is crushed about 1 meter. Ignoring the energy absorbed by the crushing of the frontal parts of the car, which then leads to a safer design of the springs, the impact force can be calculated as [6]:

$$F_d = 1/2\, mv^2 \Rightarrow F \times 1 = 1/2 \times 1200 \times 225 = 375000J \tag{4}$$

The force applied to all the springs should not exceed 375 kN. Therefore, by considering a safety factor of 1.33, the force applied to each of the rings is calculated to be 125 kN. The relevant constraint can be stated as:

$$\overline{\varphi}_2 = F_{\max} - F \geq 0 \Rightarrow \overline{\varphi}_2 = 125000 - \frac{1.15\sigma_y L t^2}{2(r_0 - \delta/2)} \geq 0 \tag{5}$$

Cold forming of absorber is possible if the mean radius of the ring r_0 is greater than three times of the thickness. Regarding a minimum thickness of 0.18034 cm for the rings (the reason will be explained furthur), the third constraint can be written as:

$$\overline{\varphi}_3 = r_0 - r_{\min} \geq 0 \Rightarrow \overline{\varphi}_3 = r_0 - 0.54102 \times 10^{-2} \geq 0 \tag{6}$$

By considering a height of 35 cm for the bumper, the maximum radius for each ring is approximately calculated as 0.15 m. This can be written as:

$$\overline{\varphi}_4 = r_{\max} - r_0 \geq 0 \Rightarrow \overline{\varphi}_4 = 0.15 - r_0 \geq 0 \tag{7}$$

The minimum length of each ring is taken as four times of its thickness. This will make its size suitable for embedding it between the bumper outer surface and the springs. By considering $L_{\min} = 1.442\ cm$, the fifth constraint will be:

$$\overline{\varphi}_5 = L - L_{\min} \geq 0 \Rightarrow \overline{\varphi}_5 = L - 1.442 \times 10^{-2} \geq 0 \tag{8}$$

Since the bumper consists of four rings, and each spring can have a maximum length of 0.7 m, the total length of the springs is 2.8 m, which can be stated as:

$$\overline{\varphi}_6 = L_{\max} - L \geq 0 \Rightarrow \overline{\varphi}_6 = 2.8 - L \geq 0 \tag{9}$$

As stated above, cold forming of absorber is possible if the mean radius of the ring is greater than three times of its thickness. Therefore, the seventh constraint can be written as:

$$\overline{\varphi}_7 = r_0 - 3t \geq 0 \tag{10}$$

Rings with thicknesses higher than 0.071 inch (0.18034 cm) can be found much easier than the ones with lower thicknesses. Therefore the eighth constraint can be taken as:

$$\overline{\varphi}_8 = t - 0.18034 \times 10^{-2} \geq 0 \tag{11}$$

Regarding the connection between r_0 and t, an approximate value of t_{max} can be found by $r_{max}=3t_{max}$. Therefore:

$$\overline{\varphi}_9 = 0.05 - t \geq 0 \tag{12}$$

The available spring constants for connective springs (between 1 and 10^8) can be considered as two other constraints:

$$\overline{\varphi}_{10} = k - k_{\min} \geq 0 \Rightarrow \overline{\varphi}_{10} = k - 1 \geq 0 \tag{13}$$

$$\overline{\varphi}_{11} = k_{\max} - k \geq 0 \Rightarrow \overline{\varphi}_{11} = 10^8 - k \geq 0 \tag{14}$$

The displacement of the of the absorbing rings, δ, must always be positive, therefore a minimum value of 0.001 was taken for it. Regarding the space limit in the frontal part of the car, δ can be chosen to have a maximum value of 1. Then:

$$\overline{\varphi}_{12} = \delta \geq 0.001 \Rightarrow \overline{\varphi}_{12} = -\delta + 0.001 \leq 0 \tag{15}$$

$$\overline{\varphi}_{13} = \delta \leq 1 \Rightarrow \overline{\varphi}_{13} = \delta - 1 \leq 0 \tag{16}$$

The energy capacity of the bumper is enough if the potential energy gained after elastic deformation of the springs is higher than the initial kinetic energy, i.e. $P.E. - K.E. \geq 0$, where $K.E. = 1/2\, mv^2$ and:

$$P.E. = \frac{1.5}{12}\sigma_y L t^2 \ln\left(\frac{r_0}{r_0 - \delta/2}\right) + \frac{1}{2}k\delta^2 \tag{17}$$

Since the car mass is 1200 kg and its initial velocity is 25 m/s, the above constraint can be rewritten as:

$$\overline{\varphi}_{14} = -375000 + \frac{1.5}{12}\sigma_y L t^2 \ln\left(\frac{r_0}{r_0 - \delta/2}\right) + \frac{1}{2}k\delta^2 \geq 0 \tag{18}$$

The above equation can get infinite if the amount into logarithm gets negative. So the fifteenth constraint can be written as:

$$\frac{r_0}{r_0 - \delta/2} > 0 \Rightarrow r_0 > \frac{\delta}{2} \tag{19}$$

Therefore:

$$\overline{\varphi}_{15} = r_0 - \frac{\delta}{2} > 0 \tag{20}$$

In numerical calculations, it is needed to normalize the constraints. It is due to the fact that the constraints are different in magnitude and unit. As a result when a constraint is violated, it is difficult to judge about how intense the violation has been. Therefore, in the current study we first normalize the constraints and then use them in different algorithms. Before normalizing the constraints, the notation of the variables are changed as: $r_0 \to x_1$, $L \to x_2$, $t \to x_3$, $\delta \to x_4$, $k \to x_5$. Then:

$$\varphi_1 = 1 - \frac{2\left(x_1 - \sqrt{2x_1}\sin(5^\circ)\right)}{x_4} \leq 0 \tag{21}$$

$$\varphi_2 = \frac{1.15 \times 40 \times 10^7 x_2 x_3^2}{250000\left(x_1 - x_4/2\right)} - 1 \leq 0 \tag{22}$$

$$\varphi_3 = 1 - 184.83 x_1 \leq 0 \tag{23}$$

$$\varphi_4 = 6.67 x_1 - 1 \leq 0 \tag{24}$$

$$\varphi_5 = 1 - 69.348x_2 \le 0 \tag{25}$$

$$\varphi_6 = x_2/2.8 - 1 \le 0 \tag{26}$$

$$\varphi_7 = 1 - x_1/3x_3 \le 0 \tag{27}$$

$$\varphi_8 = 1 - 555.56x_3 \le 0 \tag{28}$$

$$\varphi_9 = 20x_3 - 1 \le 0 \tag{29}$$

$$\varphi_{10} = 1 - x_5 \le 0 \tag{30}$$

$$\varphi_{11} = x_5/10^8 - 1 \le 0 \tag{31}$$

$$\varphi_{12} = -1000x_4 + 1 \le 0 \tag{32}$$

$$\varphi_{13} = x_4 - 1 \le 0 \tag{33}$$

$$\varphi_{14} = 1 - \left[\frac{x_5 x_2 x_3^2}{3000000}\ln\left(\frac{x_1}{x_1 - x_4/2}\right) + \frac{x_6 x_4^2}{187500}\right] \le 0 \tag{34}$$

$$\varphi_{15} = 1 - 2x_1/x_4 < 0 \tag{35}$$

Using the new notation for the variables, the objective function (i.e. Eq. (2)) becomes

$$f(x) = 9.81(7800) \times 100\pi . x_1 x_2 x_3 + 0.4\sqrt{x_5} \tag{36}$$

3. Optimization Algorithms

3.1. KKT

By the method of weight coefficients, as one of the oldest method of multi-criteria optimization, solutions are obtained by solving the scalar task [7]. In mathematics, the Karush–Kuhn–Tucker (KKT) conditions (also known as the Kuhn–Tucker conditions) are first order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. Allowing inequality constraints, the KKT approach to nonlinear programming generalizes the method of Lagrange multipliers, which allows only equality constraints. The system of equations corresponding to the KKT conditions is usually not solved directly, except in the few special cases where a closed-form solution can be derived analytically. In general, many optimization algorithms can be interpreted as methods for numerically solving the KKT system of equations [8, 9]. If x^* is a regular local minimum point for *f(x)* which has the equality and inequality constraints of [10]:

$$h_i(x) = 0 \quad i=1 \text{ to } p \tag{37}$$

$$g_i(x) \le 0 \quad i=1 \text{ to } p$$

The Lagrangian function is defined as:

$$L(x,v,u,s) = f(x) + \sum_{i=1}^{p} v_i h_i(x) + \sum_{i=1}^{m} u_i (g_i(x) + s_i^2) \tag{38}$$

There are Lagrangian constants v^* (a vector of size p) and u^* (a vector of size m) for which the Lagrangian is stable with respect to x_i, v_i, u_i *and* s_i. It means:

$$\begin{aligned} \frac{\partial L}{\partial x_j} &= \frac{\partial f}{\partial x_j} + \sum_{i=1}^{p} v_i^* \frac{\partial h_i}{\partial x_j} + \sum_{i=1}^{m} u_i^* \frac{\partial g_i}{\partial x_j} = 0 \quad \textit{j=1 to n} \\ h_i(x^*) &= 0 \quad \textit{i=1 to p} \\ g_i(x^*) + s_i^2 &= 0 \quad \textit{i=1 to m} \\ u_i^* s_i &= 0 \quad \textit{i=1 to m} \\ u_i^* &\geq 0 \quad \textit{i=1 to m} \end{aligned} \tag{39}$$

in which the derivatives are calculated in point x^*. Solving the equations (39) simultaneously, the candidate minimum points can be found. For the current problem, the Lagrangian can be stated as:

$$\begin{aligned} L = {} & 7.6518\,10^9\,\pi\,x_1\,x_2\,x_3 + 400\,x_5^{0.5} + u_1 \left(1 \right. \\ & \left. - \frac{2\left(x_1 - 0.0872\sqrt{2}\,\sqrt{x_1}\right)}{x_4} + s_1^2 \right) \\ & + u_2 \left(\frac{4.600\,10^7\,x_2\,x_3^2}{250000x_1 - 125000x_4} - 1 + s_2^2 \right) + u_3 \left(1 - 184.83x_1 \right. \\ & \left. + s_3^2\right) + u_4 \left(6.67x_1 - 1 + s_4^2\right) + u_5 \left(1 - 69.348x_2 + s_5^2\right) \\ & + u_6 \left(0.35714x_2 - 1 + s_6^2\right) + u_7 \left(1 - \frac{1}{3}\,\frac{x_1}{x_3} + s_7^2 \right) + u_8 \left(1 \right. \\ & \left. - 555.56x_3 + s_8^2\right) + u_9 \left(20\,x_3 - 1 + s_9^2\right) + u_{10} \left(1 - x_5 + s_{10}^2\right) \\ & + u_{11} \left(1.\,10^{-8}\,x_5 - 1 + s_{11}^2\right) + u_{12} \left(-1000x_4 + 1 + s_{12}^2\right) \\ & + u_{13} \left(x_4 - 1 + s_{13}^2\right) \end{aligned} \tag{40}$$

3.2. Sequential Linear Programming (SLP) algorithm

The sequential linear programming (SLP) method is one of the easiest optimization techniques used to treat non-linear optimization problems. Successful application of the method depends on the proper selection of the move limits, which are unfortunately uncertain in the SLP algorithm [11]. The SLP algorithm is [10]:

Step 1: Guess an initial design point $x^{(0)}$. Set cycle number to zero ($k=0$). Set two positive small amounts for ε_1 and ε_2.

Step 2: Calculate the constraints and objective function values in the current design point $x^{(k)}$,i.e.:

$$f_k = f(x^{(k)}) \tag{41}$$

$$e_j = -h_j(x^{(k)}) \qquad j=1 \text{ to } m \tag{42}$$

$$b_j = -g_j(x^{(k)}) \qquad j=1 \text{ to } p \tag{43}$$

Step 3: Calculate the gradients of the constraints and objective functions, i.e.:

$$c_i = \frac{\partial f}{\partial x_i} \qquad i=1 \text{ to } n \tag{44}$$

$$n_{ij} = \frac{\partial h_j}{\partial x_i} \qquad i=1 \text{ to } n; \qquad j=1 \text{ to } p \tag{45}$$

$$a_{ij} = \frac{\partial g_j}{\partial x_i} \qquad i=1 \text{ to } n; \qquad j=1 \text{ to } p \tag{46}$$

Step 4: Choose an appropriate move limit $\Delta_{il}^{(k)}$ and $\Delta_{iu}^{(k)}$ as a fraction of current design.

Step 5: Define the relevant LP sub-problem by:

$$\overline{f} = c^T d \tag{47}$$

$$N^T d = e \tag{48}$$

$$A^T d \le b \tag{49}$$

Step 6: If needed, convert the LP sub-problem into standard Simplex form and calculate $d^{(k)}$ solving it.

Step 7: Check the convergence. If the following inequalities are satisfied, stop the solution, otherwise continue to Step 8.

$$\left\|d^{(k)}\right\| \le \varepsilon_2, \qquad \left\|h_i\right\| \le \varepsilon_1, \qquad g_i \le \varepsilon_1 \tag{50}$$

Step 8: Update the design point by:

$$x^{(k+1)} = x^{(k)} + d^{(k)} \tag{51}$$

and set $k=k+1$ and go to step 2. It must be kept in mind that in this algorithm, convergence is highly dependent on move limits. In fact, move limits can be so limiting that no solution is found for the LP sub-problem. For example, in the current project a move limit of $u=0.26=26\%$ was found to be appropriate. A small change in the appropriate amount of u decreased the convergence speed very quickly. It must also be mentioned that the convergence is also dependent on the initial point. The appropriate initial point must be chosen by try and error (as it was for the u). For example in the current paper, if the initial point has δ greater

than *0.01*, *L* greater than 0.03, and r_0 greater than 0.03, the solution will be divergent.

3.3. Constrained Steepest-Descent (CSD) algorithm

CSD is an effective method, in which model-based design decisions can be applied to multidisciplinary, conceptual design. Of particular concern is the ability to implement this framework without having to extensively modify or adapt the software used for the multidisciplinary system optimization. The CSD algorithm is [10]:

Step 1: Set $k=0$. Guess an initial design point $x^{(0)}$. Choose an appropriate initial value for penalty parameter R_0, a constant γ between 0 and 1, and two positive small amounts for ε_1 and ε_2. $R_0=1$ and $\gamma=0.25$ are suitable choices.

Step 2: Calculate objective function, constraint functions and their gradients at $x^{(k)}$. Calculate the highest constraint violation by:

$$V_k = \max\left\{0;|h_1|,|h_2|,...,|h_p|;g_1,g_2,...,g_m\right\} \tag{52}$$

Step 3: Using the objective function, constraint functions and their gradients, define the (Quadratic Programming) QP sub-problem:

$$\overline{f} = c^T d + 0.5 d^T d \tag{53}$$

$$N^T d = e \tag{54}$$

$$A^T d \le b \tag{55}$$

Solve the QP problem for searching direction $d^{(k)}$ and Lagrange coefficients $v^{(k)}$ and $u^{(k)}$.

Step 4: Check the convergence criterion $\left\|d^{(k)}\right\| \le \varepsilon_2$ and the highest constraint violation V_k. If the optimization criterion is satisfied stop the solution, otherwise continue.

Step 5: Calculate r_k summation:

$$r_k = \sum_{i=1}^{p}\left|v_i^{(k)}\right| + \sum_{i=1}^{m} u_i^{(k)} \tag{56}$$

Set $R=max\ \{R_k, r_k\}$.

Step 6: Set $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, where α_k is a suitable step size.

Step 7: save the current penalty parameter $R_{k+1}=R$. Update the counter ($k=k+1$) and go to step 2.

It must be noted that in the CSD method there is no need to define move limits. In the SLP method, choosing an unsuitable amount of move limit decreases the convergence rapidly. Another advantage of this method is that, unlike the SLP method, this convergence is not dependent on the initial point.

3.4. Pshenichny-Lim-Belegundu-Arora (PLBA) algorithm

The recursive quadratic programming algorithm, PLBA algorithm, uses the derivatives of the objective function and the constraint function with respect to design variables to search for an optimum direction [12]. The PLBA algorithm is [13, 14]:

Step 1: Set k=0. Guess an initial design point $x^{(0)}$. Choose an appropriate initial value for penalty parameter R_0, a constant γ between 0 and 1, and two positive small amounts for ε_1 and ε_2. $R_0 = 1$ and $\gamma = 0.25$ are suitable choices. Choose a unit matrix for initial Hessian matrix (i.e. $H^{(0)}=I$)

Step 2: Calculate objective function, constraint functions and their gradients at $x^{(k)}$.Calculate the highest constraint violation by (46).

If $k=0$ go to step 3, otherwise if $k>0$, update the Hessian matrix using equations below:

$$s^{(k)} = \alpha_k d^{(k)} \tag{57}$$

$$z^{(k)} = H^{(k)} s^{(k)} \tag{58}$$

$$y^{(k)} = \nabla L(x^{(k+1)}, u^{(k)}, v^{(k)}) - \nabla L(x^{(k)}, u^{(k)}, v^{(k)}) \tag{59}$$

The difference between the gradients of the Lagrangian function between two points is:

$$\xi_1 = s^{(k)}.y^{(k)} \tag{60}$$

$$\xi_2 = s^{(k)}.z^{(k)} \tag{61}$$

$$if \begin{cases} \xi_1 \geq 0.2\xi_2 \rightarrow \theta = 1 \\ \xi_1 < 0.2\xi_2 \rightarrow \theta = \dfrac{0.8\xi_2}{(\xi_2 - \xi_1)} \end{cases} \tag{62}$$

$$w^{(k)} = \theta y^{(k)} + (1-\theta) z^{(k)} \tag{63}$$

$$\xi_3 = s^{(k)}.w^{(k)} \tag{64}$$

$$D^{(k)} = \frac{1}{\xi_3} w^{(k)} w^{(k)^T} \tag{65}$$

$$E^{(k)} = \frac{1}{\xi_2} z^{(k)} z^{(k)^T} \tag{66}$$

And finally the Hessian matrix is updated by:

$$H^{(k+1)} = H^{(k)} + D^{(k)} - E^{(k)} \tag{67}$$

Step 3: Using the objective function, constraint functions and their gradients, define the (Quadratic Programming) QP sub-problem:

$$\overline{f} = c^T d + 0.5 d^T H d \tag{68}$$

$$N^T d = e \tag{69}$$

$$A^T d \leq b \tag{70}$$

Solve the QP problem for searching direction $d^{(k)}$ and Lagrange coefficients $v^{(k)}$ and $u^{(k)}$.

Steps 4-7: Like steps 4-7 for the CSD algorithm (subsection 3.3)

4. Results and Discussion

The optimization problem was solved using the five methods mentioned. Table 1 shows the optimum points and optimum prices resulted from different methods. As it can be seen all the codes show very close results and their results can be considered coincident. The optimum price for KKT and SLP was 10.565 $. For CSD, modified CSD, and PLBA algorithms the optimum price was 10.495 $. The main difference between the codes can be the differences in solution parameters, i.e. solution time, convergence, etc. Table 2 lists the specifications of the solutions done by the methods. All the problems were solved using MATLAB on a PC having 2.26 GHz Core2Due CPU and 3 GB of RAM.

After solving the problem using the CSD algorithm, it was seen that in the optimum point, four design variables are at their lowest boundary. In order to get better results new boundaries were considered for the variables. This is called modified CSD in Tables 1 and 2.

$$\begin{array}{lcl} 0.0054102 \le x_1 \le 0.15 & & 0.01 \le x_1 \le 0.1 \\ 0.01442 \le x_2 \le 2.8 & & 0.01442 \le x_2 \le 0.1 \\ 0.0018034 \le x_3 \le 0.05 & \Rightarrow & 0.0018034 \le x_3 \le 0.05 \\ 0.001 \le x_4 \le 0.1 & & 0.001 \le x_4 \le 0.1 \\ 1 \le x_5 \le 10^8 & & 1 \le x_5 \le 10^2 \end{array} \quad (71)$$

It was seen that the CSD, PLBA, and KKT methods are convergent to a local minimum point starting from any point. On the other hand, the SLP method should not be used as a black box approach for engineering design problems. The selection of move limits is one of trial and error and can be best achieved in an interactive mode. Also, the SLP method may not converge to the precise minimum since no descent function is defined, and line search is not performed along the search direction to compute a step size [10]. While the KKT algorithm is relatively simple and short in length, it also has some shortcomings. The KKT conditions are not applicable at the points that are not regular. In those cases their use may yield candidate minimum points; however, the Lagrange multipliers may not be unique [10].

It was seen that the CSD algorithm converges to a local minimum point starting from an arbitrary point, feasible or infeasible. The starting point can affect performance of the algorithm. For example, at some points the QP subproblem may not have any solution. This need not mean that the original problem is infeasible. The original problem may be highly nonlinear, and the linearized constraints may be inconsistent, giving an infeasible QP subproblem. This

situation can be handled by either temporarily deleting the inconsistent linearized constraints or starting from another point. For more discussion on the implementation of the algorithm, Tseng and Arora [15] may be consulted [13].

It seems that the modified CSD method is the best choice for optimizing the problem introduced in the paper. It is because it needs a short programming and it converges relatively fast. In general, the PLBA algorithm converges quicker than modified CSD but anyway it needs a lengthy programming. Therefore it can be stated that in large problems, the PLBA algorithm is preferred, while in the small problems the modified-CSD is suggested.

Table 1: Optimum points and prices resulted from different methods

Method	Optimum Point $(r_0 \quad L \quad t \quad \delta \quad k)^T$	Optimum Price
KKT	$(0.016292 \quad 0.01442 \quad 0.0018 \quad 0.001 \quad 1)^T$	10.565 $
SLP	$(0.016292 \quad 0.01442 \quad 0.0018 \quad 0.001 \quad 1)^T$	10.565 $
CSD	$(0.0162 \quad 0.0144 \quad 0.0018 \quad 0.001 \quad 1)^T$	10.495 $
Modified-CSD	$(0.0162 \quad 0.0144 \quad 0.0018 \quad 0.001 \quad 1)^T$	10.495 $
PLBA	$(0.0162 \quad 0.0144 \quad 0.0018 \quad 0.001 \quad 1)^T$	10.495 $

Table 2: Specifications of optimization codes

Method	Solution Time (Minutes)	Convergence Dependency on Initial Point	Length of Program	Number of Steps Required
KKT	60	No	Short	-
SLP	30	Yes	Medium	2
CSD	480	No	Medium	5
Modified-CSD	60	No	Medium	4
PLBA	45	No	Lengthy	3

5. Conclusions

In this work, an energy absorber system was optimized using four optimizing methods namely Karush-Kuhn-Tucker (KKT), Sequential Linear Programming (SLP), Constrained Steepest-Descent (CSD), and Pshenichny-Lim-Belegundu-Arora (PLBA). Time solution, convergence, programming length and accuracy of the results were considered in order to find the best solution algorithm. It sounds that the modified CSD method is the best choice for optimizing the problem introduced in the paper. It is because it needs a short programming and it converges relatively fast. In general, the PLBA algorithm converges quicker than modified CSD but anyway it needs a lengthy programming. Therefore it can be stated that in large problems, the PLBA algorithm is preferred, while in the small problems the modified-CSD is suggested.

REFERENCES

[1]. *S.S. Rao*, Engineering optimization: Theory and practice. Hoboken, NJ: John Wiley, 2009.

[2]. *Q. Wang, J.S. Arora*, "Alternative formulations for transient dynamic response optimization", AIAA Journal, vol. 43, no. 10, pp. 2188-2195, 2005.

[3]. *Q. Wang, J.S. Arora*, "Alternative formulations for structural optimization: An evaluation using trusses". AIAA Journal, vol. 43, no. 10, pp. 2202-2209, 2005.

[4]. *Q. Wang, J.S. Arora*, "Several alternative formulations for transient dynamic response optimization: An evaluation". International Journal for Numerical Methods for Engineering, vol. 80, pp. 631-650, 2009.

[5]. *J.N. Siddall*, Analytical Decision Making in Engineering Design, Prentice Hall, Englewood Cliffs, 1972.

[6]. *A.K. Chopra*, Dynamics of structures: Theory and applications to earthquake engineering (3rd ed.). Upper Saddle River, NJ: Prentice-Hall, 2007.

[7]. *M. Žižović, N. Damljanović, V. Lazarević, N. Deretić*, New Method For Multicriteria Analysis, U.P.B. Sci. Bull., Series A, Vol. 73, Iss. 2, 2011.

[8]. *S. Boyd, L*. Vandenberghe, Convex Optimization. Cambridge: Cambridge University Press, pp. 244, ISBN 0-521-83378-7, MR 2061575, 2004.

[9]. *D.A. Pierre*, Optimization theory with applications, Courier Dover Publications, 2012.

[10]. *J.S. Arora*, Introduction to Optimum Design, Academic Press, 2011.

[11]. *T.Y. Chen*, "Calculation of the Move Limits For the Sequential Linear Programming Method", International Journal for Numerical Methods in Engineering, vol. 36, pp. 2661-2679, 1993.

[12]. *O.K. Lim, J.S. Lee*, Structural Topology Optimization for the Natural Frequency of a Designated Mode, KSME International Journal, vol. 14, no.3, pp. 306-313, 2000.

[13]. *A.D. Belegundu, J.S. Arora*, "A Recursive Quadratic Programming Algorithm with Active Set Strategy for Optimal Design". International Journal for Numerical Methods in Engineering, vol. 20, no. 5, pp. 803-816, 1985.

[14]. *O.K. Lim, and J.S. Arora*, "An Active Set RQP Algorithm for Optimal Design". Computer Methods in Applied Mechanics and Engineering, vol. 57, pp. 51-65, 1986.

[15]. *C. H. Tseng, J.S. Arora*, "On implementation of computational algorithms for optimal design 1: Preliminary investigation; 2: Extensive numerical investigation". International Journal for Numerical Methods in Engineering, vol. 26, no.6, 13651402, 1988.