
TOOLS
TECHNICAL NOTE
CSILOADOPTIMIZER

- [Overview](#)
- [Description](#)
- [Operation](#)
- [Step-By-Step Procedure](#)
- [SAP2000 Example](#)
- [CSiBridge Example](#)

Overview

CSiLoadOptimizer is a tool available to SAP2000 v15 and later and CSiBridge v15 and later for determining an optimal set of loads, including cable tensioning, to achieve specified goals in a structural model. The loads to be optimized can be applied in any static load case, which can be of type linear, nonlinear, or staged-construction. The goals are specified as the values to be attained for response quantities such as joint displacements, generalized displacements, joint reactions, member forces or moments, and/or bridge superstructure forces and moments (CSiBridge only). The optimization operation consists of determining the scale factors of variable loads in the static load case linear to best meet these goals.

Description

[\(back to top\)](#)

Three classes of problems can be solved, depending on the relationship between the number of variable loads, N_V , and the number of goals, N_G :

- $N_V > N_G$, Optimization problem.
Multiple solutions may exist, and an optimization algorithm is used. Each variable load is assigned a cost, and the cost is minimized using the sum of the squares as the cost (objective) function. For optimization problems, limits may also be specified. Limits are the same as goals, but are specified

as inequalities, whereas goals are specified as equalities. The number of limits is denoted N_L .

- $N_V = N_G$, Determinate problem.
There is only one solution.
- $N_V < N_G$, Best-fit problem.

There may be no solution that exactly achieves the goals, so a least-squares fit is sought. Each goal is assigned a benefit, and the solution is sought that minimizes the error, where the error for each goal is weighted by its benefit.

Any load that can be applied in a static load case can be used for load optimization, including gravity, acceleration, force, strain, temperature, displacement, prestress, pressure, and so on. Target-force loads can be used but are not recommended, as they may compete with the optimization.

Goals (and limits) include force or deformation measures at individual locations. Currently available are:

- Joint displacement, any component
- Joint reaction, any component
- Frame force/moment, any component, at End I, End J, Max, or Min
- Cable tension, at End I, End J, Max, or Min
- Tendon tension, at End I, End J, Max, or Min
- Link force/moment, any component, at End I, End J, Max, or Min
- Link deformation, any component
- Generalized displacement
- Bridge force/moment, at any superstructure section cut (CSiBridge only)

Where Max and Min options are available, the extreme value taken over all the output stations of the object is used.

It is important to choose variable loads and goals that are relevant and independent. For example, a symmetrical load pattern cannot achieve an unsymmetrical goal, and using such a variable load reduces the effective number of variable loads, N_V . Similarly, constrained displacements are neither

independent of each other, nor are the tensions in two cable elements in series, so using these pairs reduces the effective number of goals, N_V . These are properties of the model and cannot be determined automatically by the CSiLoadOptimizer algorithm. However, they may cause a reasonable solution not to be found.

Operation

[\(back to top\)](#)

To use CSiLoadOptimizer, define a static load case as a template for the loads to be applied. The load case can be linear, nonlinear, or nonlinear staged construction. Each of the N_V variable load patterns should be applied in the template load case with scale factors of the user's choosing. In a staged construction load case, applying the same load pattern to a different group, or in a different stage if the structure has changed, can act as an independent variable. For linear load cases, and nonlinear cases that behave linearly, the scale factors are arbitrary, and the same solution will be found regardless of the values chosen. For nonlinear load cases with nonlinear behavior, iteration is often required to find the optimal solution. For this reason, choosing scale factors that are close to the optimal solution improves the rate and chance of convergence. It is possible to apply other load patterns in the template load case that are not variable but are applied as specified.

After defining the template load case, use the Tools menu to start CSiLoadOptimizer. Specify the template load case, select the load patterns to be optimized (the N_V variables), and assign them a cost. Then specify the goals (and limits) and assign them a target value, error tolerance, and benefit. For optimization problems ($N_V > N_G$), targets may be specified as equalities (goals) or inequalities (limits). For determinate and best-fit problems, only goals will be considered.

Then run the optimization. A copy of the template load case will be created, and it will be perturbed for each variable load pattern, and a separate analysis will be performed. The incremental changes in the goals are measured for each incremental change in the variable load patterns, and an influence matrix is formed. This forms the basis for the optimization, determinate, or best-fit problem that is solved for the optimum load pattern scale factors. These are applied to the copied case, which is solved, and another measure of the goals is taken. For linear load cases, or nonlinear load cases that behave

linearly, this is the final answer. For nonlinear behavior, this is a linear approximation of the solution, and the error in the goals is checked to determine whether additional iterations are necessary. Iteration is continued until convergence is obtained or the iteration limit is reached. In any case, CSiLoadOptimizer runs multiple intermediate load cases. For large nonlinear models with many variables this may take a significant amount of time. Canceling the analysis cancels the optimization.

The output of this process is the modified load case in the SAP2000 or CSiBridge model. A log file and summary report file are also produced.

You can save your settings in a data file that can be read back in the future. Because this data file is not part of the model, changes made to the model may not be compatible with the saved data. If inconsistencies are found between the saved data and the model, the data file cannot be read back in.

Step-by-Step Procedure

[\(back to top\)](#)

1. Define the problem in SAP2000 or CSiBridge. Decide what goals and limits are desired.
 - a. They should be realistic and independent.
 - b. Goals can include force, moment, or displacement values to be achieved at one or more locations in the structure.
 - c. Each goal should be independent of others. For example, displacements at points that are connected by constraints cannot be independently specified. Similarly, force values that must be in equilibrium with each other cannot be independently specified.
 - d. Limits are the same types of response quantities as goals, but are specified as inequalities to be satisfied. These should be independent of each other and of the goals.
 - e. If necessary, define model components that measure these goals, such as special link elements or generalized displacements.
 - f. The number of goals and limits is called N_G .
2. Decide what variable load patterns can be applied to the structure to achieve these goals.
 - a. These loads will be applied in a static load case (linear, nonlinear, or staged-construction).
 - b. Any type of load can be applied. The most common types are forces on joints or members, support displacements, or strain loads.
 - c. For cable-tightening, it is recommended to define a single load pattern, and assign a unit strain (say 0.001) to all cables in the model.
 - d. Similar to goals, the variable loads must be independent of each other.
 - e. Using staged construction, it is possible to apply the same load pattern to different objects or groups, or to the same objects and groups at different configurations of the structure. Each such load application can be considered to be a different load so long as their effects are independent.
 - f. Loads must be effective at achieving the goals. For example, symmetric loads cannot achieve anti-symmetric goals.
 - g. The number of independent variable load applications to be used is called N_V .

3. Define a template static load case that applies these load patterns.
 - a. Apply each variable load pattern in this load case.
 - b. A single load pattern can be applied multiple times in a staged-construction load case to different parts of the structure.
 - c. For linear cases, the scale factor for each variable load application is arbitrary. For nonlinear load cases, a scale factor close to the expected optimum value may help with convergence of the solution if this value can be estimated. Otherwise, choose a scale factor for which a nonlinear solution can be achieved in the template load case. In any case, zero is an acceptable scale factor.
 - d. Other loads that are not being used for affecting the goals may also be applied in the same load case. The selection of which loads are fixed and which are variable will be done later.
 - e. CSiLoadOptimizer will copy the template load case and modify the scale factors on the variable loads to achieve the desired goals and limits.
4. Using the Tools menu, start CSiLoadOptimizer.
5. Enter a Name for the new data set you will be creating. This data set includes the loads and goals for the problem, and will apply to a particular template load case.
6. Choose the Load Case Type and Load Case name for the template load case that you created. The Load Assignments grid will be filled from that load case. Modify these as follows:
 - a. Set any load patterns that you do not wish to vary to "Fixed." These load patterns will still be applied as defined, but will not be used as variables in the operation.
 - b. Optionally, set a relative cost for each variable load. This only affects optimization problems. Loads with larger cost will tend to result in smaller scale factors when possible. The cost is multiplied by the scale factor when contributing to the optimization objective function.
 - c. Optionally, set the perturbation factor. This is the incremental scale factor used when creating the influence matrix for solving for the scale factors. This only affects nonlinear problems. If convergence is a problem, changing these factors may help, typically by reducing them.

7. Define the Goals and Limits:
 - a. For each goal, choose the type, and then set the name, location, and component, as appropriate.
 - b. Set the Sense (equality or inequality type) and the Target Value. Note that the Sense defines whether a response quantity is a goal (equality) or limit (inequality). Target values are given in the appropriate units for the given displacement, force, or moment measure. The units in effect are always the database units of the model, as shown in the lower-right corner of the form.
 - c. The Stage shown is when the goal will be measured. Currently, this is always the end of the load case.
 - d. Optionally, set a relative benefit for each goal. This only affects best-fit problems. Goals with larger benefit will tend to result in smaller errors when solving the best-fit problem. The benefit is multiplied by the error in achieving the goal. Since the goals have units, the Benefits have the inverse units. It would be expected that displacement goals would be given significantly larger benefit factors than force goals, since displacements tend to have much smaller magnitudes than forces in typical units for structural systems.
 - e. Set the absolute convergence tolerance for each goal. This only affects nonlinear problems where iteration may be required. These tolerances have the same units as the goals themselves.
8. Check the type of problem to be solved:
 - a. This is computed by comparing the number of variables, N_V , with the number of goals, N_G , and is shown in the upper right corner of the form.
 - b. For Optimization problems, choose the type of objective function to be used. Currently, this is always Sum of Squares.
9. For nonlinear problems, set the iteration parameters:
 - a. Max Iterations: The number of times to attempt to find the optimum solution.
 - b. Accel Factor: Multiplier on the change in the solution during nonlinear iteration. This should always be set to unity for linear problems. For nonlinear problems that converge monotonically but slowly, you may want to set this value larger than unity. For nonlinear problems that oscillate or diverge, set this value less than unity.

- c. ConvTol: This is a relative tolerance used to measure convergence of the goals.
10. After the data has been completely specified, the user may save the data to a file for later use, or run the Load Optimizer (which also saves the file). Either way, the default filename is the name of the model file concatenated with the name of this data set. The user can choose any name. The file extension is .LoadOptData.
11. When the Load Optimizer runs, the following occurs in SAP2000 or CSiBridge:
- a. A copy of the template load case is made and given a name that concatenates the name of the template load case with "Opt" and the name of the data set.
 - b. The new load case is run and the initial values of the goals are determined
 - c. The new load case is repeatedly perturbed and run. For each run, the scale factor of a single variable is changed by the perturbation factor, and the effect upon the constraints is measured.
 - d. From these multiple runs, an influence matrix is generated that relates increments in the variables with increments in the loads.
 - e. Using the influence matrix and the appropriate solver for the type of problem, a solution is obtained for the change in scale variable factors needed to achieve the goals.
 - f. The found scale factors are applied to the copied load case, and a final solution is obtained.
 - g. The values of the goals are measured and compared with the target values.
 - h. For a linear case, these should be correct if the variables and goals are well posed and the structural model is well conditioned.
 - i. For the nonlinear case, there may be errors in satisfying the goals since the scale factors were calculated assuming that superposition is valid. Errors are measured by comparing each goal with its target. Convergence is accepted if:
 - Every goal is either satisfied within its absolute tolerance or within the relative tolerance multiplied by the average magnitude of the current goal and its target.

- Alternatively, after the first iteration, if the change in the current value of every goal from the previous iteration is within its absolute tolerance.
 - j. If neither of the preceding conditions is met and the maximum number of iterations has not been exceeded, a new solution is obtained by applying the influence matrix to the error and further modifying the new load case. The acceleration factor is used to determine how much of the error is used for this purpose.
12. When the iteration is complete (either converged or the iteration limit reached), two text files are produced:
- a. A summary report with file extension .LoadOptReport
 - b. A complete log with file extension .LoadOptLog
13. After closing CSiLoadOptimizer, you can view the results for the optimized load case in SAP2000 or CSiBridge.
14. If CSiLoadOptimizer is run again, the user can open the saved file containing the data set for any previous optimization of the same model. It is important to note that any changes to the names of the template load case, the load patterns applied in this load case, or the objects used for goals will invalidate the saved data and it will not be recovered. Similarly, any changes to the number or ordering of the load assignments in the template load case will invalidate the data. Changes to the scale factors in the template case are permitted, and these will overwrite the saved values.

SAP2000 Example – Linear load case

[\(back to top\)](#)

For the SAP2000 model shown in Figure 1, we will jack up three of the columns in order to remove the second column (# 4), but limit the axial force in the three columns being jacked to prevent overstress. We will define a separate ground displacement load pattern for each jacked column. These are called Jack1, Jack9, and Jack13 for the three joints to be loaded. The deformed shape for one of the jacking load patterns is shown in Figure 2. The jacking loads will be applied in a linear load case together with dead load. The goal will be to reduce the axial force in frame #4 to zero at its top, and to keep the axial force at the bottom of the three jacked columns to about one third of the total dead load. As shown in Figure 3, the total dead load is 8.1, so we will limit the axial force in each column to -3.0 .

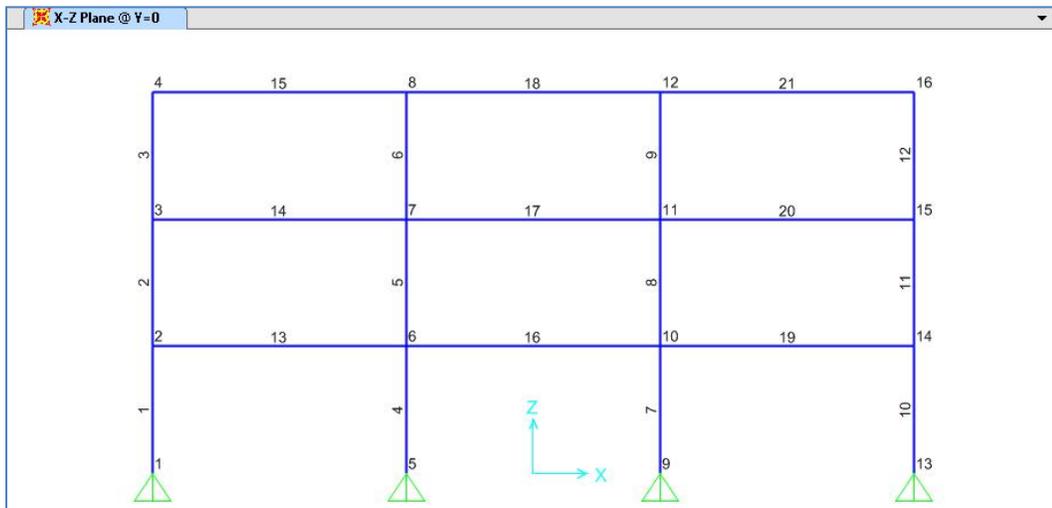


Figure 1 – SAP2000 Model

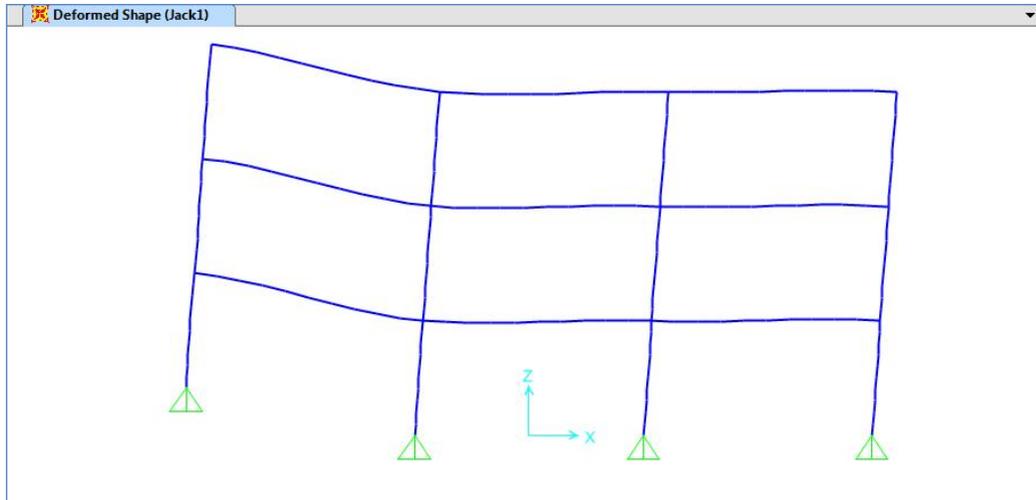


Figure 2 – Deformed shape under load Jack1.

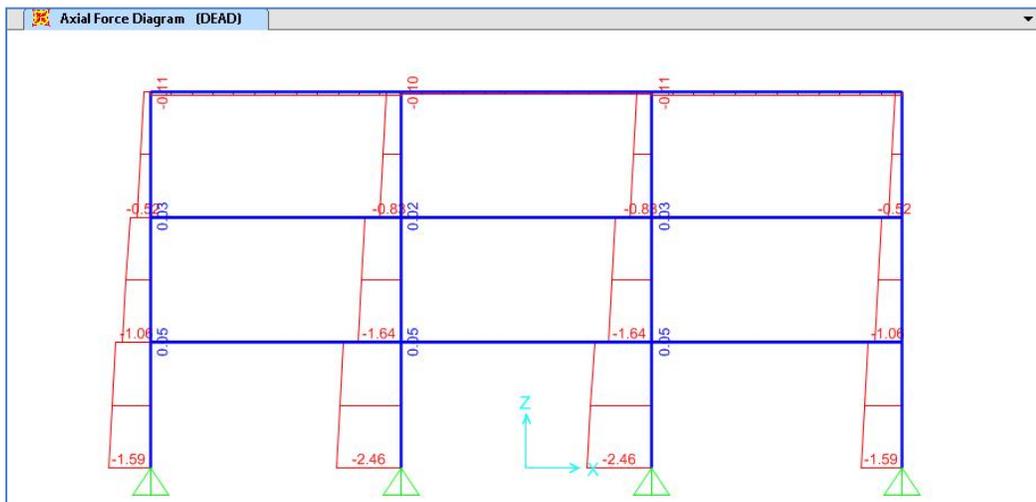


Figure 3 – Axial force under dead load

This problem has three variables, one goal, and three limits, resulting in an optimization problem with $N_V > N_G$. We will define a template linear-static load case, "DEAD+Jack," which applies dead load and the three jacking load patterns. The scale factors for the jacking load patterns have arbitrarily been set to zero. CSiLoadOptimizer is started and the template load case "DEAD+Jack" is chosen, resulting in the following form being displayed:

CSILoadOptimizer v1.0.0

Name: LoadOpt Max. Iterations: 10 Current Prob. Type: Undefined
 Load Case Type: Linear Static Accel. Factor: 1
 Load Case: DEAD+Jack Rel. Conv. Tol.: 0.001 Obj. Function Type: Sum of Squares

Load Assignments (Number of Variables: 4) Hide Fixed

Load Type	Load Name	Scale Factor	Variable	Relative Cost	Perturbation Factor
Load	DEAD	1	Variable	1	0.001
Load	Jack1	0	Variable	1	0.001
Load	Jack9	0	Variable	1	0.001
Load	Jack13	0	Variable	1	0.001

Goals and Limits (Number of Goals: 0, Number of Limits: 0) Delete Goal

Type	Name	Location	Component	Sense	Target Value	Relative Benefit	Absolute Tolerance
**							

Resources Save and Run Save Open Current Units: kip, in, F

We will make the following changes:

- Change the "DEAD" load case to be Fixed, since it cannot be changed.
- Add goal: Axial force P atEnd J (top) of Frame 4 and set the value = 0.
- Add limits: Axial forces P atEndI (bottom) of Frames 1, 7, and 10 and set the value ≥ -3 .

The resulting form, ready to run:

CSiLoadOptimizer v1.0.0

Name: LoadOpt Max. Iterations: 10 Current Prob. Type: Optimization (Minimum Cost)
 Load Case Type: Linear Static Accel. Factor: 1 Limits and Goals will be utilized
 Load Case: DEAD+Jack Rel. Conv. Tol.: 0.001 Obj. Function Type: Sum of Squares

Load Assignments (Number of Variables: 3) Hide Fixed

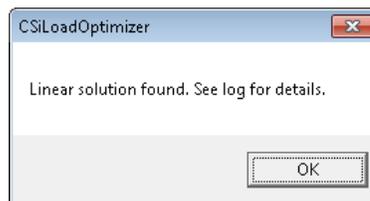
Load Type	Load Name	Scale Factor	Variable	Relative Cost	Perturbation Factor
Load	DEAD	1	Fixed	1	0.001
Load	Jack1	0	Variable	1	0.001
Load	Jack9	0	Variable	1	0.001
Load	Jack13	0	Variable	1	0.001

Goals and Limits (Number of Goals: 1, Number of Limits: 3) Delete Goal

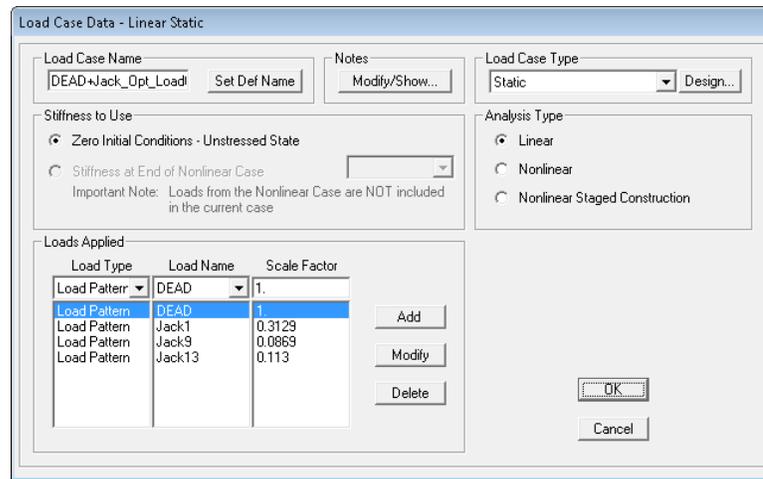
Type	Name	Location	Component	Sense	Target Value	Relative Benefit	Absolute Tolerance
Frame Force	4	EndI	P	=	0	1	0.000001
Frame Force	1	EndI	P	>=	-3	1	0.000001
Frame Force	7	EndI	P	>=	-3	1	0.000001
Frame Force	10	EndI	P	>=	-3	1	0.000001

Resources Save and Run Save Open Current Units: kip, in, F

The Save and Run button is clicked, and a filename is chosen to save these CSiLoadOptimizer settings. When complete, the following message will display:



CSiLoadOptimizer can now be closed, and the results displayed. The optimized load case created from "DEAD+Jack" by CSiLoadOptimizer is called "DEAD+Jack_Opt_LoadOpt," and has the following definition:



The corresponding deformed shape and axial forces are shown in Figures 4 and 5, respectively.

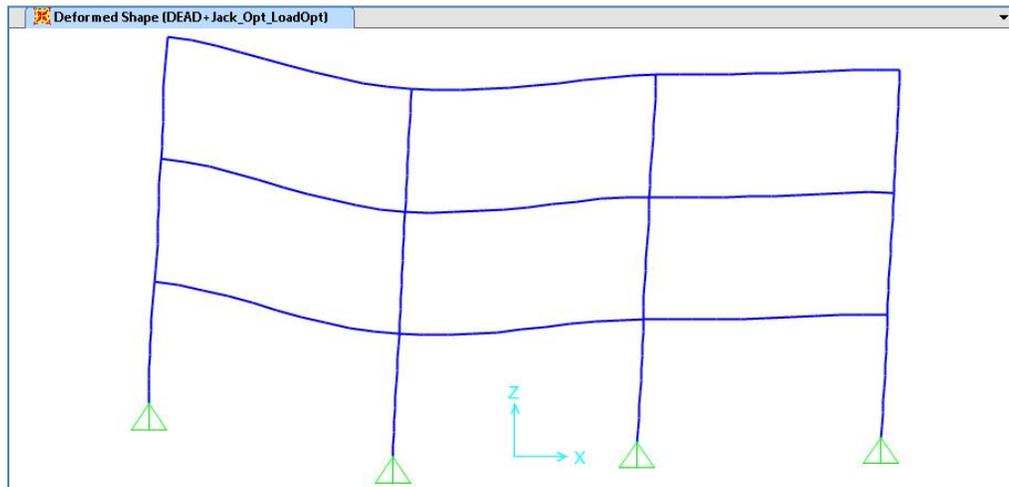


Figure 4 – Optimized deformed shape

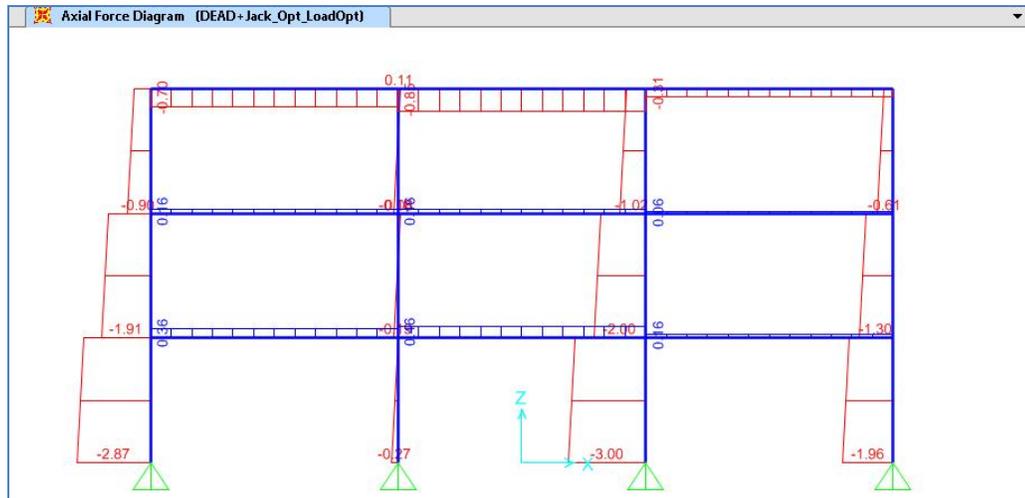


Figure 5 – Optimized axial forces

CSiBridge Example – Nonlinear Cable Tightening

[\(back to top\)](#)

For the CSiBridge model shown in Figure 6, we will apply strain load to the cables of a cable-stayed bridge to achieve a specified negative moment in the deck under dead load. The bridge deck is modeled as a bridge object with a concrete box superstructure section. The cables and tower have been added, and the cables attached to the deck with stiff links. A staged-construction load case is defined that first builds the bridge segmentally. In a subsequent load case, strain load will be applied individually to symmetrical sets of cables with load factors to be determined to achieve negative moment in the deck.

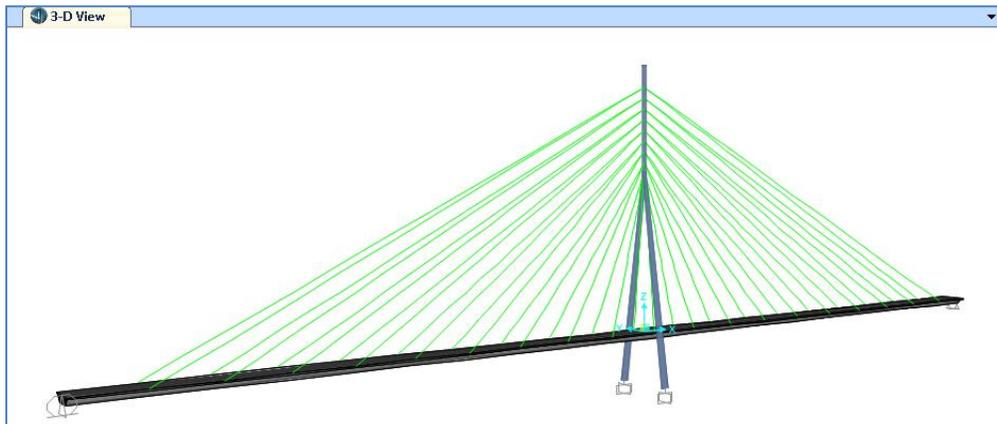


Figure 6 – Cable-Stayed Bridge model in CSiBridge

After staged-construction of the structure under dead load, the resulting moment diagram is shown in Figure 7, and the cable tensions are shown in Figure 8.

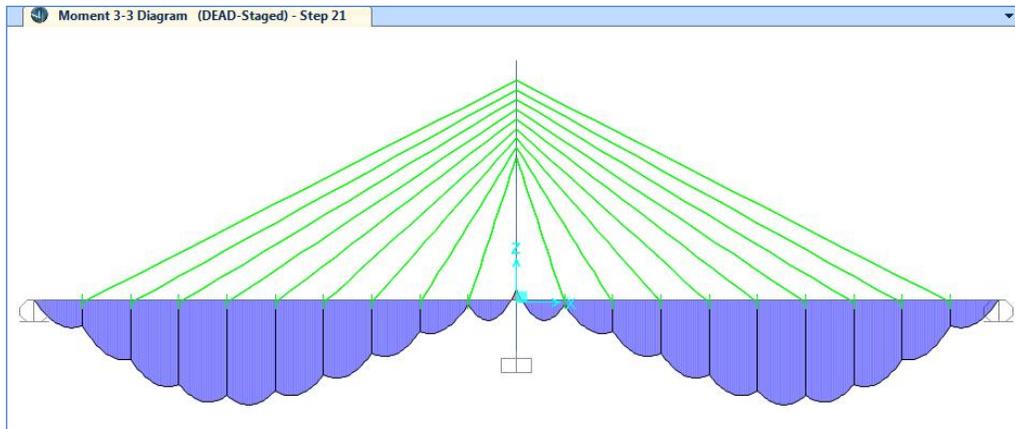


Figure 7 – Superstructure moment diagram under dead load

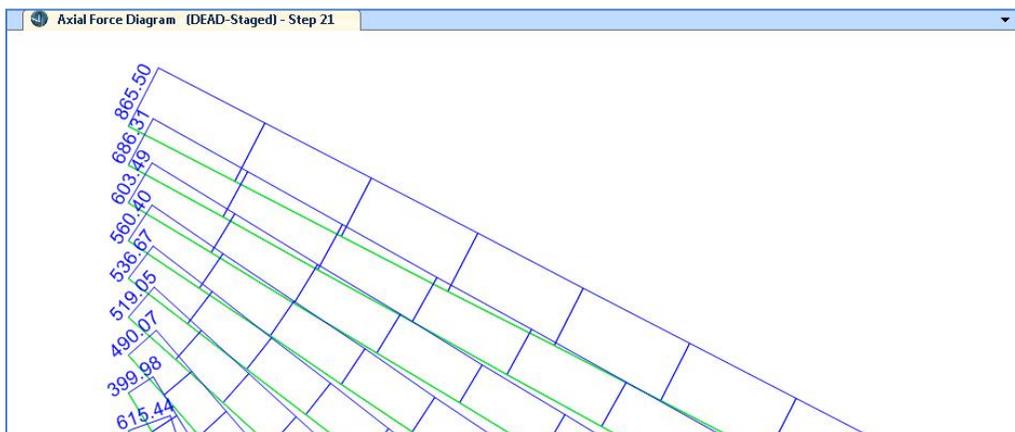


Figure 8 – Cable axial forces under dead load

We will define a single load pattern, "Tighten," that applies a uniform strain of 0.001 to all cables. Using a staged-construction load case, it is possible to apply this single load pattern separately to individual sets of cables. Using symmetry, there is one load application to the vertical pair of cables at the tower, and nine load applications to symmetrical sets of four cables on either side of the tower, for a total of ten load applications.

Because cables are highly nonlinear, it is necessary to apply the tightening operation to the structure after the construction sequence under dead load. This is done in the template load case, "Tighten," defined in the following

form. A unit scale factor is arbitrarily chosen for each load application. These scale factors will be optimized later:

Load Case Data - Nonlinear Static Staged Construction

Load Case Name: Notes:

Load Case Type:

Initial Conditions:

Zero Initial Conditions - Start from Unstressed State

Continue from State at End of Nonlinear Case

Important Note: Loads from this previous case are included in the current case

Analysis Type:

Linear

Nonlinear

Nonlinear Staged Construction

Geometric Nonlinearity Parameters:

None

P-Delta

P-Delta plus Large Displacements

Stage Definition:

Stage No.	Duration (Days)	Provide Output	Output Label	User Comments	
1	0.	No			<input type="button" value="Add"/>
1	0.	No			<input type="button" value="Add Copy"/>
					<input type="button" value="Modify"/>
					<input type="button" value="Insert"/>
					<input type="button" value="Delete"/>

Expand Stage Definition

Data For Stage 1 (0. days):

Operation	Object Type	Object Name	Age At Add	Type	Name	Scale Factor
Load Objects	Group	GROUP0		Load Patte	CableTensi	1.
Load Objects	Group	GROUP1		Load Pattern	Cable Tension	1.
Load Objects	Group	GROUP2		Load Pattern	Cable Tension	1.
Load Objects	Group	GROUP3		Load Pattern	Cable Tension	1.

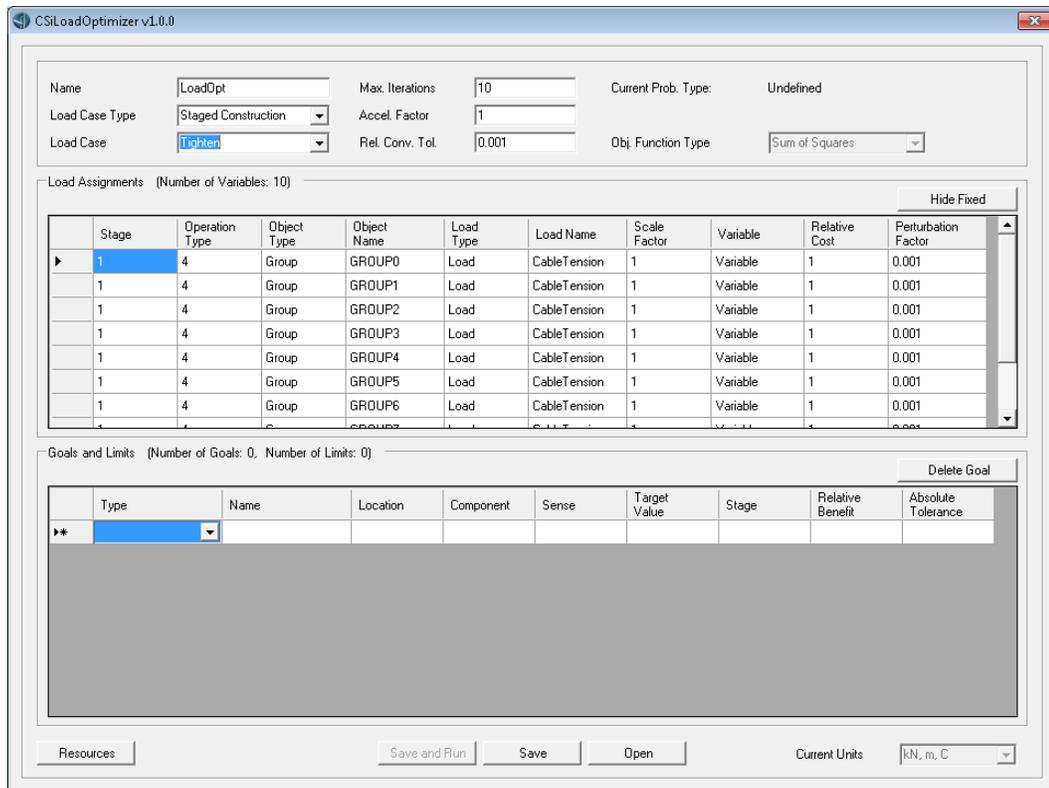
Expand Stage Data Stage: << < 1 > >> of 1

Other Parameters:

Results Saved:

Nonlinear Parameters:

CSiLoadOptimizer is started and the template load case "Tighten" is chosen, resulting in the following form being displayed:



At each bridge superstructure section cut where the cables attach, we will require the major moment to be equal to -1000 on the outer (away from the tower) side of the attachment. This assures that the moment between the cable attachments will be positive at the center of most deck segments. This problem has no limits, only goals. We could seek 19 goals, one at each cable-pair attachment location. This would create a best-fit problem, since the number of goals exceeds the number of loads. Instead, we will use symmetry and seek ten goals: the moment at the tower, plus the moments at the nine cable attachments on one side of the tower. This creates a determinate problem. Because the loads are symmetrical, both formulations would produce the same solution.

Because cable behavior is highly nonlinear, we can expect that iteration will be required to find the optimal solution. Accordingly:

- The absolute tolerances were changed from the defaults for each goal and set to realistic values (10) to speed convergence

- The number of iterations is left at the default value of 10
- The acceleration factor is set to a value less than unity to reduce oscillation of the iterative solution
- The relative convergence was not changed

The values are problem-dependent, and some experimentation may be needed to determine the best values.

The resulting form, ready to run:

CSiLoadOptimizer v1.0.0

Name: LoadOpt Max. Iterations: 10 Current Prob. Type: Determinate
 Load Case Type: Staged Construction Accel. Factor: 0.618 Only Goals will be utilized
 Load Case: Tighten Rel. Conv. Tol.: 0.001 Obj. Function Type: Sum of Squares

Load Assignments (Number of Variables: 10)

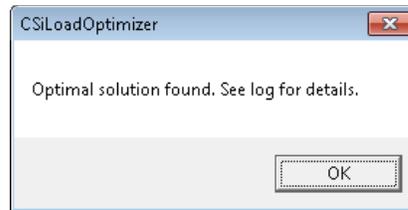
Stage	Operation Type	Object Type	Object Name	Load Type	Load Name	Scale Factor	Variable	Relative Cost	Perturbation Factor
1	4	Group	GROUP0	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP1	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP2	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP3	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP4	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP5	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP6	Load	CableTension	1	Variable	1	0.001
1	4	Group	GROUP7	Load	CableTension	1	Variable	1	0.001

Goals and Limits (Number of Goals: 10, Number of Limits: 0)

Type	Name	Location	Component	Sense	Target Value	Stage	Relative Benefit	Absolute Tolerance
Bridge Force	BOBJ1	10.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	20.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	30.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	40.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	50.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	60.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	70.000 Before	M3	=	-1000	1	1	10
Bridge Force	BOBJ1	80.000 Before	M3	=	-1000	1	1	10

Resources Save and Run Save Open Current Units: kN, m, C

The Save and Run button is clicked, and a filename is chosen to save these CSiLoadOptimizer settings. When complete, the following message box is displayed:



Before closing CSiLoadOptimizer, we can select the optimized load case "Tighten_Opt_LoadOpt" that was created to see the optimized scale factors that were found:

CSiLoadOptimizer v1.0.0

Name: LoadOpt Max. Iterations: 10 Current Prob. Type: Determinate
 Load Case Type: Staged Construction Accel. Factor: 0.618 Only Goals will be utilized
 Load Case: Tighten_Opt_LoadOpt Rel. Conv. Tol.: 0.001 Obj. Function Type: Sum of Squares

Load Assignments (Number of Variables: 10) Hide Fixed

	Stage	Operation Type	Object Type	Object Name	Load Type	Load Name	Scale Factor	Variable	Relative Cost	Perturbation Factor
▶	1	4	Group	GROUP0	Load	CableTension	-59.0764863	Variable	1	0.001
	1	4	Group	GROUP1	Load	CableTension	-52.9861007...	Variable	1	0.001
	1	4	Group	GROUP2	Load	CableTension	-39.3243545...	Variable	1	0.001
	1	4	Group	GROUP3	Load	CableTension	-28.2122397...	Variable	1	0.001
	1	4	Group	GROUP4	Load	CableTension	-20.2574103...	Variable	1	0.001
	1	4	Group	GROUP5	Load	CableTension	-14.5373756...	Variable	1	0.001
	1	4	Group	GROUP6	Load	CableTension	-10.2509603...	Variable	1	0.001
	1	4	Group	GROUP7	Load	CableTension	-6.87399329...	Variable	1	0.001

Goals and Limits (Number of Goals: 10, Number of Limits: 0) Delete Goal

	Type	Name	Location	Component	Sense	Target Value	Stage	Relative Benefit	Absolute Tolerance
▶	Bridge Force	BOBJ1	10.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	20.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	30.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	40.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	50.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	60.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	70.000 Before	M3	=	-1000	1	1	10
	Bridge Force	BOBJ1	80.000 Before	M3	=	-1000	1	1	10

Resources Save and Run Save Open Current Units: kN, m, C

Note that the first scale factor is -59.1 , and corresponds to a strain of about -6% applied to the central cables. The strain load decreases towards the ends of the bridge. Checking the CSiLoadOptimizer log file, it is found that nine nonlinear iterations were required to find the solution after the influence matrix was determined.

CSiLoadOptimizer can now be closed and the results displayed. The resulting moment diagram is shown in Figure 9 and the cable tensions shown in Figure 10. Results are as expected.

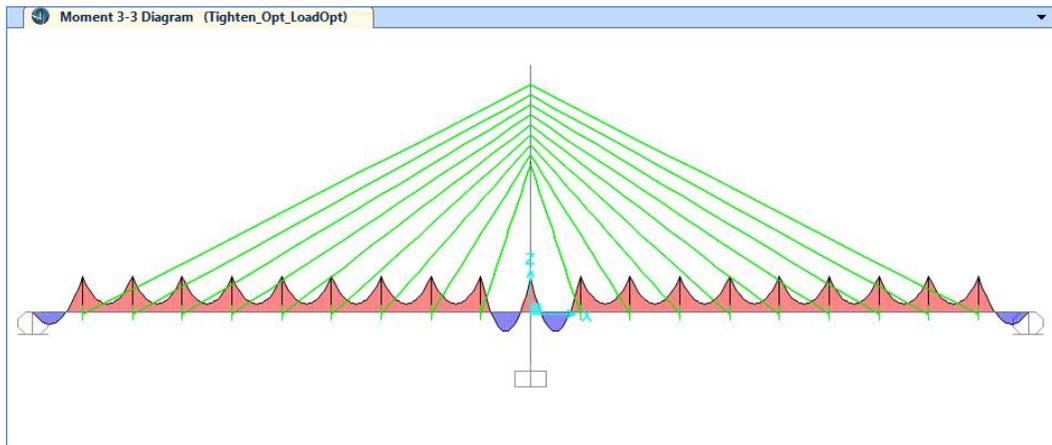


Figure 9 – Superstructure moment diagram after optimizing cable tension

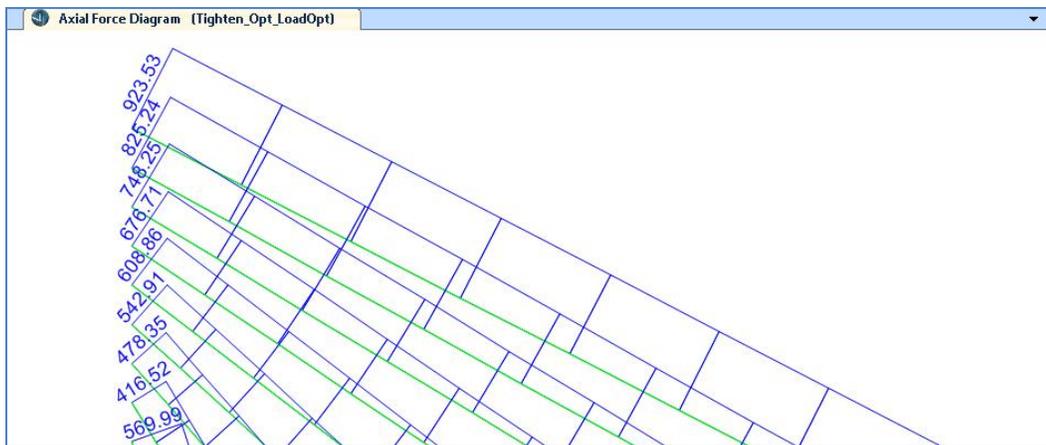


Figure 10 – Optimized cable tensions