Introducing Single-Criterion Optimization Methods into Mechanics Classes

William K. Szaroletta
Purdue University, West Lafayette, Indiana

Abstract:
Single criterion optimization problems are shown to be readily taught and understood at lower division course levels using algebra/calculus, exhaustive numerical searches, and solver-type tools in standard spreadsheet packages. Genetic algorithms provide another suitable numerical technique that is relatively easily understood by students in high-level form, directly applicable to the single criterion class of optimization problems and very helpful to the multiple criteria class of optimization problems.

This paper describes the methods of lower-division mechanics classroom introduction of optimization methods including algebra/calculus, spreadsheet solver, exhaustive search, and genetic algorithms. A classical solid mechanics problem utilizing the simply supported beam with a central load is used as the baseline in this paper for presenting the optimization methods introduced. Several other more complex problems are described. Multiple criteria optimization problems, which can quickly exceed the capability of typical spreadsheet solver tools, require students to utilize the multiple criteria optimization capability of genetic algorithms software. A possible framework that would support both single criterion and multiple criteria optimization methods, based on using genetic algorithms software, is presented that could allow these powerful methods to be introduced and successfully utilized earlier in the student's college experience.

Introduction:
Two major course learning objectives of a lower-division mechanics course, MET 211, Applied Strength of Materials, are: to understand the differences between analysis and design problems and to be able to properly address both types of problems. Analysis problems typically require a given input design with input design parameters[1,2]. Design problems typically are much broader or open-ended in scope, often requiring the devising, analyzing, and testing of a series of design alternatives, before subsequent design analysis can begin. The analysis step, in either case, can be greatly enhanced if the student has a working knowledge of the utilization of optimization methods at their disposal. Single-criterion optimization methods have been shown to help analyze various design alternatives, helping selecting the best or optimal alternative.

Single Criterion Optimization Methods Introduced:
Four basic methods of optimization are introduced in this course to support the students analysis and design work, including algebra/calculus, exhaustive search, spreadsheet solvers, and genetic algorithms. All four methods are worked into this lower-division mechanics classroom in the sequence shown above.

1) **Algebraic/Calculus Method:** This method works well where there is a continuous function that is easily differentiable. Technology students typically take calculus as a co-requisite making broad application of these methods difficult. Goldberg[7] points out that

Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition

calculus-based methods are based on the existence of "quadratic objective functions, ideal constraints and ever-present derivatives".

2) **Exhaustive Search Method:** This method works very well in that the method is logical and all of the students have a good knowledge of spreadsheets.
3) **Spreadsheet Solver Method:** This method is typically unknown to all but one or two students coming into the course. After lecture and laboratory introduction, the speed with which optimal solutions can be obtained is welcomed compared with methods 1) and 2) above.
4) **Genetic Algorithms Method:** This method has both single-criterion and multiple-criteria capability, making it ultimately applicable to a much broader base of mechanics problems.

Mechanics Problems Utilizing Optimization:

There are many problems in the mechanics world that can utilize optimization methods. Mechanics and optimization textbooks[1,2,3,4,5,6,7] carry a plethora of analysis and design problems that can also serve as good optimization problems and a sampling of these excellent texts is included in the Bibliography of this paper. A typical optimization will require a design to have a constant stress, minimum weight, and/or minimum cost, though there are many optimization fitness functions that can be envisioned. All four methods mentioned above have applicability to these classes of optimization problems with a single objective function, f(x), and one or more constraints, where x is a member the constraint set.

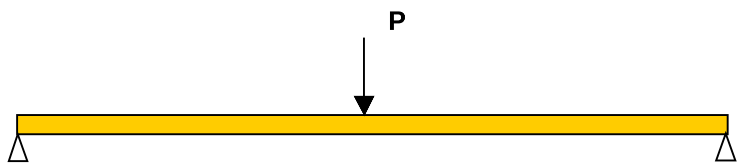
Very few mechanical elements experience a constant stress state throughout the entire element, though this condition would better utilize the element's material. Parts are typically loaded such that there is one point that has the maximum stress while the other material is stressed at a lower level. Thus, the material that is stressed less than the maximum stress level is not utilized optimally. This class of optimization problem requires a optimization fitness function that is looking to output a variation of stress across the element equal to or below some set threshold, usually set by establishing a suitable design factor or factor of safety with respect to the yield strength of the material, Syt. The stress states in various portions of the mechanical element are first parameterized and then the cross-section is varied locally to produce an almost-constant stress state.

Initial mechanical element designs rarely exhibit the minimum possible weight. Often low weight can mean improved utility and/or reduced cost due to material savings. For beams of a given length, the cross-sectional area can be minimized, giving minimum weight for a given material. It is possible, of course, that changing the mechanical element's material from steel (density ~0.3 lb/ft^3) to aluminum (density ~0.1 lb/ ft^3) can give lower weight even though the cross-section for the aluminum is larger than that of the steel.

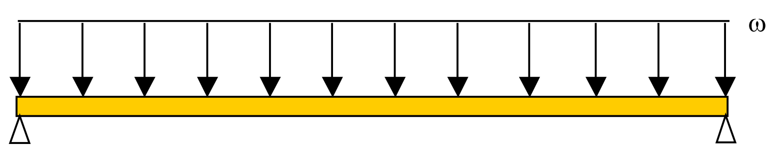
Assemblies of two-force mechanical linkages into pinned trusses made of a given material can utilize both minimization of cross-sectional area and producing constant stress as joint optimization goals.

Cost minimization of individual parts and multiple part assemblies is often a major criterion in the marketability of a product or machine. Minimization of part count in an assembly, by eliminating redundant load paths and/or unloaded elements, is one approach to providing minimum cost. Minimizing weight sometimes can imply minimum cost, though extreme caution must be taken with the minimum weight approach with higher-cost, lighter-weight materials like titanium and beryllium.

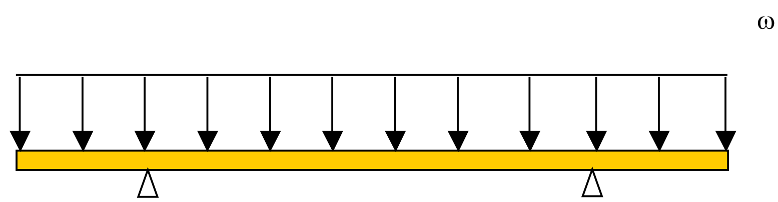
Optimization Problem Descriptions:
An optimization learning progression from a simply supported beam with a central load to a simply supported beam with a distributed load to the more complex overhung beam with a distributed load is presented in this paper. These problems enable students to utilize analysis methods with given parameters, utilizing lower-division mechanics course lecture material and experimental procedures, to initially solve them. Subsequent discussion of optimization broadens both their interest and deepens their understanding of these problems. An overview sketch of these three mechanics problems involving beams is shown in Figure 1 below.



A) Simply Supported Beam with Central Load



B) Simply Supported Beam with Distributed Load



C) Overhung Beam with Distributed Load

Figure 1: Three mechanics problems involving beams utilized for optimization learning

Detail Optimization Example Using a Simply Supported Beam with Central Load:
This simple, straight-forward, and standard mechanics problem provides students an understanding of optimization in both analysis and design modes. The sketch of the loaded beam depicted in Figure 1A is analyzed using mechanics principles. This analysis, utilizing free-body, instantaneous load, distributed load, and moment diagrams is shown in Figure 2 below.

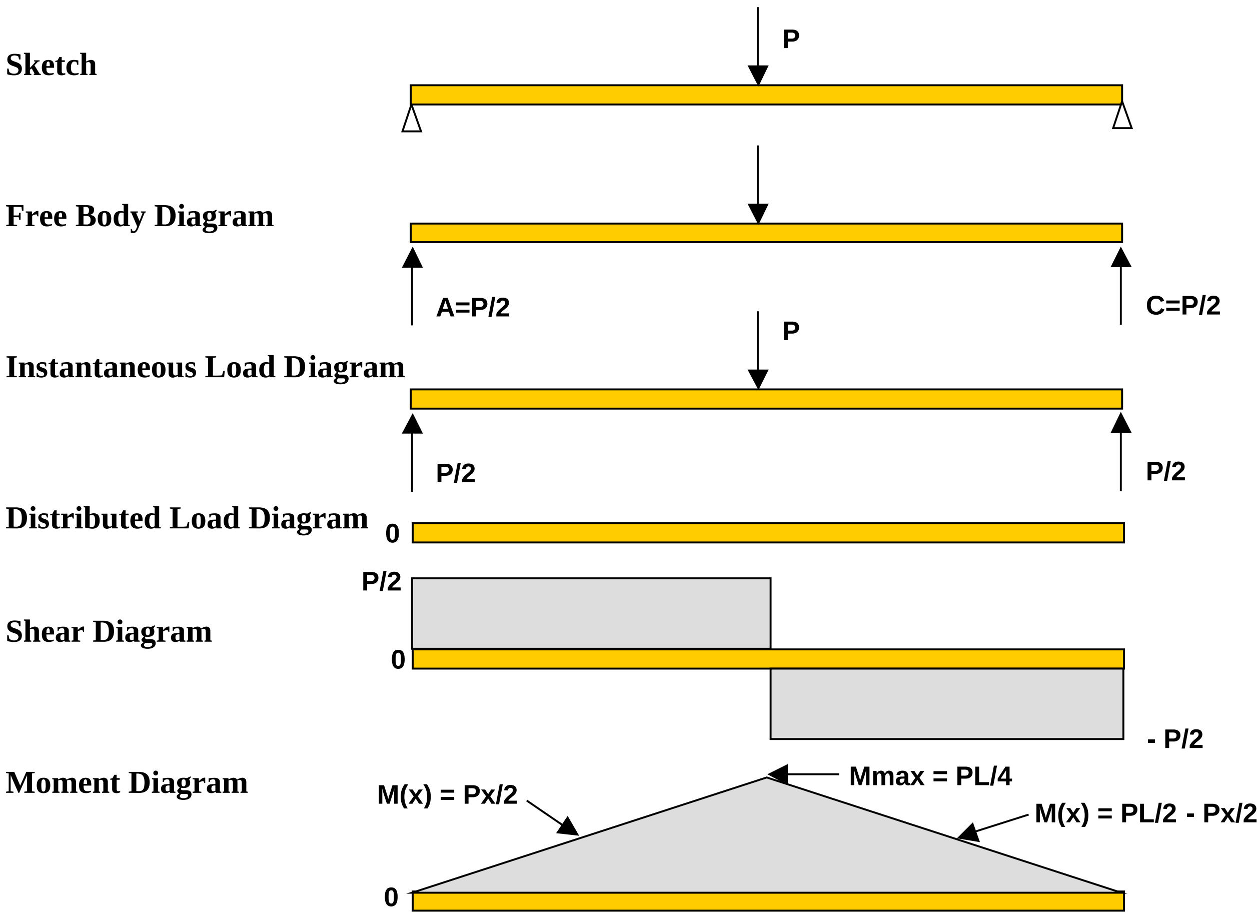


Figure 2: Simply supported beam with central load

The flexural bending stress, σ(x), is proportional to the bending moment, M(x), and inversely proportional to the section modulus, Z(x) as shown in equation 1 from Shigley, Mischke[1].

$$\sigma(x) = M(x) / Z(x) \qquad \text{Eq 1}$$

For a constant cross-section beam, the section modulus, Z(x), is constant along x according to equation 2, where b(x) represents the width of the beam at any location, x and h(x) represents the height of the beam at any location, x.

$$Z(x) = b(x) * h(x)^2 / 6 \qquad \text{Eq 2}$$

A non-optimal solution often produces a constant Z(x) because of utilization of standard cross-section materials. The constant Z(x) leads to a σ(x) profile that mirrors M(x) with low values at each support and a maximum value at midspan as shown in Figure 3 below.

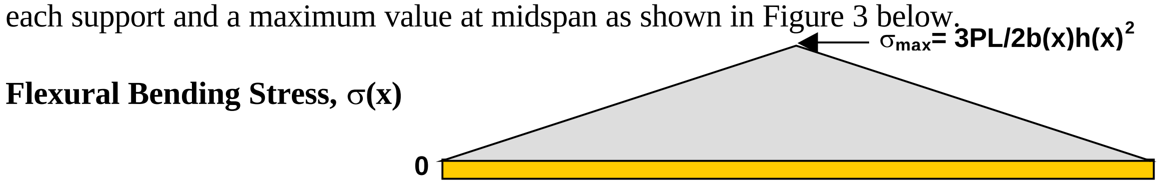


Figure 3: A simply supported beam with a flexural bending stress, σ(x), that mirrors M(x) for a constant cross-section beam.

The bending moment varies from zero at the supports to a maximum in the beam's midspan which mirrors the flexural bending stress of the beam span. A typical analysis problem will ask the student to find the maximum stress level in the beam for a given loading. A typical design problem will ask the student to size the beam to produce a given or determined upper-bound stress. Both methods result in large portions of the beam's material being underutilized from a stress standpoint and thus at a non-optimal design state.

Constant Flexural Bending Stress as Optimization Goal:
It is clear from the above discussion that a constant-stress beam will have flexural bending stress graph that is a horizontal line as shown in Figure 4 below.

Constant Flexural Bending Stress, $\sigma(x)$

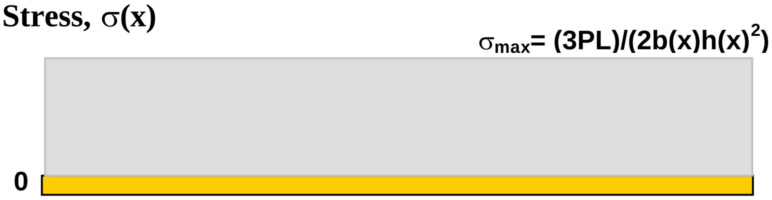


Figure 4: A simply supported beam with a constant flexural bending stress, $\sigma(x)$, for a beam that has variable cross-section along x.

Clearly more of the material in the beam is utilized more efficiently in Figure 4. This constant flexural stress state can be accomplished in several ways, by varying the section modulus along the beam's axis, for the loading shown. One method to accomplish this is to vary the beam width, b(x), while holding the beam height, h(x) constant. Another method is to vary the beam height, while holding the beam width constant. A third method is to vary both beam width and height together. In this paper, achieving the constant stress state in Figure 3 will be explored using the method of varying the beam width, b(x).

Varying Beam Width, b(x), to Produce a Variable Beam Section Modulus:
Simple algebra allows the students to solve for beam width, b(x) for the first half of the simply supported beam in terms of the other parameters, with the result shown in Equation 3.

$$\sigma(x) = M(x) / Z(x) = (P\ x / 2) / (b(x) * h(x)^2 / 6) \Rightarrow b(x) = (3\ P\ x) / (h(x)^2 * \sigma_{max}) \qquad \text{Eq 3}$$

Thus the underlying optimization problem becomes determining the values of b(x) at each location x to produce the desired level of constant stress, in this case σ_{max}.

1) Algebraic/Calculus Method:
Students can perform the algebra and solve for the required b(x) at any location from 0 to L/2 to produce a constant stress from 0 to L/2. The simple algebra could easily transcend the needs for more robust methods to determine the necessary b(x) to produce a constant stress as a function of x. Even though this introductory optimization problem is simply solved mathematically by the students, the understanding of the problem solution by varying the cross-section provides a profound impact on future design and optimization problems. As mentioned above, with the cross-sectional beam dimensions constant along the length, the complexity of the problem is reduced. A non-trivial outcome is an enhanced student ability to interpret design analysis outputs

and apply these outputs to the actual beam design. The students observe from the output of Equation 3 that as x → 0, b(x) → 0 and at x = 0, b(0) = 0. This is clearly unworkable, since mounting features must exist and the beam supports and shear stresses would become larger in relation to the flexural bending stresses. After a discussion of the presence of lower-level shear stresses due to shear forces and the other stresses due to mounting features, this apparent paradox of requiring a zero-width part is resolved. Working on relatively simple optimization problems gave the students improved confidence as they approach progressively more difficult problems.

2) Exhaustive Search Method:
The exhaustive search method utilizes spreadsheets like Microsoft[8] Excel™ to allow multiple, manually-performed iterations to produce the desired result. These iterations take two basic forms.

The first form of the exhaustive search method requires the student to set up an initial table that has desired initial bounds and a coarse grid. After finding the best solution within those initial bounds, another table is set up beneath the initial table with the bounds are made progressively finer. After finding the best solution within these finer bounds, the process is repeated with finer bounds until a result with the desired precision is obtained.

The second form of the exhaustive search method requires only a single table and a student understanding of the binary search algorithm to accomplish. It was interesting to the author that only a few students had previously encountered the algorithm, though the majority understood the algorithm quickly. In this second form, the independent variable is varied manually (by the student) within the same cell using a high-level binary search algorithm until the desired precision result is obtained. The student enters a number, observes the results, and decides if the number needs to increase or decrease based on the result. This process was demonstrated during lecture with follow-up homework. Proper understanding of the underlying parametric equation was required, though, to support rapid convergence to an optimal solution through knowledge of which direction to vary the independent variable.

In the case of the simply supported beam with a central load, the solution requires the optimal beam width, b(x), be derived for each predetermined value of x. For example, given a 30 inch long simply supported beam with a central load of 1000 pounds, a maximum stress level of 5000 psi, and a beam height of 3.0 inches, at x = 1", the value of b(1") is determined by the exhaustive search method shown below in Figure 5. The first table produces a lower bound for b(1") of 0.01" and an upper bound of 1.01" as determined by the trend of the delta function, which is the difference between the calculated and goal values of stress at that location . The next table is created from the first using cut-and-paste methods with a finer granularity between these newly established bounds. The second iteration puts the bounds between 0.01" and 0.1", the third iteration puts the bounds between 0.06" and 0.07", and the fourth iteration puts the bounds between 0.066" and 0.067". The last iteration shows the bounds between 0.0666" and 0.0667", with the latter being the specified value due to the calculated stress being less than the goal stress. At the value of b(1") = 0.0667", the calculated stress is a couple of psi below the goal.

Exhaustive Search								
		1st Iteration						
x	**b(x)**	**h(x)**	**M(x)**	**Sigma(x)**	**Goal**	**Delta**		
1	0.01	**3**	**500**	33333.33	**5000**	**28333.33**	**Bound 1**	**0.01**
1	1.01	**3**	**500**	330.033	**5000**	**-4669.97**	**Bound 2**	**1.01**
1	2.01	3	500	165.8375	5000	-4834.16		
1	3.01	3	500	110.742	5000	-4889.26		
1	4.01	3	500	83.12552	5000	-4916.87		
1	5.01	3	500	66.5336	5000	-4933.47		
1	6.01	3	500	55.46312	5000	-4944.54		
1	7.01	3	500	47.55112	5000	-4952.45		
1	8.01	3	500	41.61465	5000	-4958.39		
1	9.01	3	500	36.99593	5000	-4963.00		
1	10.01	3	500	33.30003	5000	-4966.70		
		2nd Iteration						
x	**b(x)**	**h(x)**	**M(x)**	**Sigma(x)**	**Goal**	**Delta**		
1	0.01	**3**	**500**	33333.33	**5000**	**28333.33**	**Bound 1**	**0.01**
1	0.11	**3**	**500**	3030.303	**5000**	**-1969.70**	**Bound 2**	**0.11**
1	0.21	3	500	1587.302	5000	-3412.70		
1	0.31	3	500	1075.269	5000	-3924.73		
1	0.41	3	500	813.0081	5000	-4186.99		
1	0.51	3	500	653.5948	5000	-4346.41		
1	0.61	3	500	546.4481	5000	-4453.55		
1	0.71	3	500	469.4836	5000	-4530.52		
1	0.81	3	500	411.5226	5000	-4588.48		
1	0.91	3	500	366.3004	5000	-4633.70		
1	1.01	3	500	330.033	5000	-4669.97		
		3rd Iteration						
x	**b(x)**	**h(x)**	**M(x)**	**Sigma(x)**	**Goal**	**Delta**		
1	0.01	3	500	33333.33	5000	28333.33		
1	0.02	3	500	16666.67	5000	11666.67		
1	0.03	3	500	11111.11	5000	6111.11		
1	0.04	3	500	8333.333	5000	3333.33		
1	0.05	3	500	6666.667	5000	1666.67		
1	0.06	**3**	**500**	5555.556	**5000**	**555.56**	**Bound 1**	**0.06**
1	0.07	**3**	**500**	4761.905	**5000**	**-238.10**	**Bound 2**	**0.07**
1	0.08	3	500	4166.667	5000	-833.33		
1	0.09	3	500	3703.704	5000	-1296.30		
1	0.1	3	500	3333.333	5000	-1666.67		
1	0.11	3	500	3030.303	5000	-1969.70		
		4th Iteration						
x	**b(x)**	**h(x)**	**M(x)**	**Sigma(x)**	**Goal**	**Delta**		
1	0.06	3	500	5555.556	5000	555.56		
1	0.061	3	500	5464.481	5000	464.48		
1	0.062	3	500	5376.344	5000	376.34		
1	0.063	3	500	5291.005	5000	291.01		
1	0.064	3	500	5208.333	5000	208.33		
1	0.065	3	500	5128.205	5000	128.21		
1	0.066	**3**	**500**	5050.505	**5000**	**50.51**	**Bound 1**	**0.066**
1	0.067	**3**	**500**	4975.124	**5000**	**-24.88**	**Bound 2**	**0.067**
1	0.068	3	500	4901.961	5000	-98.04		
1	0.069	3	500	4830.918	5000	-169.08		
1	0.07	3	500	4761.905	5000	-238.10		
		5th Iteration						
x	**b(x)**	**h(x)**	**M(x)**	**Sigma(x)**	**Goal**	**Delta**		
1	0.066	3	500	5050.505	5000	50.51		
1	0.0661	3	500	5042.864	5000	42.86		
1	0.0662	3	500	5035.247	5000	35.25		
1	0.0663	3	500	5027.652	5000	27.65		
1	0.0664	3	500	5020.08	5000	20.08		
1	0.0665	3	500	5012.531	5000	12.53		
1	0.0666	**3**	**500**	5005.005	**5000**	**5.01**	**Bound 1**	**0.0666**
1	0.0667	**3**	**500**	4997.501	**5000**	**-2.50**	**Bound 2**	**0.0667**
1	0.0668	3	500	4990.02	5000	-9.98		
1	0.0669	3	500	4982.561	5000	-17.44		
1	0.067	3	500	4975.124	5000	-24.88		

Figure 5: Example of an exhaustive search to find the value of beam width at x = 1", b(1"), to produce a constant stress of 5000 psi. The value of b(1") = 0.0667" at x = 1" is the output, since this produces a stress slightly less that the established maximum.

Note that the first iteration in Figure 5 does not include an option for b(1") to have a value of 0.0" since this would cause a divide-by-zero MS Excel™ error. Instead the increments were made slightly greater than zero, in this case 0.01" as the minimum. The exhaustive search method for determining the required b(x) to produce a constant stress state is laborious since each individual value of x requires a mini-exhaustive search as shown in Figure 5 above. Depending on the granularity required for x, this process can produce a spreadsheet that can quickly become bulky.

3) Spreadsheet Solver Method:
Subsequent to the student's efforts with MS Excel™ utilizing the exhaustive search algorithm described above, the MS Solver™ Add-In to MS Excel™ is introduced. Classroom demonstrations have been very warmly received due to the rapid convergence of Solver™ to an optimal solution for single-criterion optimization problems. Lecture material and demonstrations show the students how to establish a single target cell as the objective function with the goal to maximize, minimize, or drive to a particular value. In the example below, the same problem of finding the optimal distribution of beam width, b(x) for a simply supported beam with a central load is addressed utilizing Solver™. For each location, x, the variable input cell, b(x), drives the other constant parameters to establish a flexural bending stress for that location, which is then compared with the target maximum allowable flexural bending stress. The "in-process" stress values are subtracted from the "target" stress values producing a column of "deltas". This spreadsheet arrangement is very similar to the Exhaustive Search spreadsheet model as can be observed in Figure 6 below.

x	b(x)	h(x)	M(x)	Sigma(x)	Goal	Delta
1	0.3	3	500	1111.111	5000	-3888.89
2	0.3	3	1000	2222.222	5000	-2777.78
3	0.3	3	1500	3333.333	5000	-1666.67
4	0.3	3	2000	4444.444	5000	-555.56
5	0.3	3	2500	5555.556	5000	555.56
6	0.3	3	3000	6666.667	5000	1666.67
7	0.3	3	3500	7777.778	5000	2777.78
8	0.3	3	4000	8888.889	5000	3888.89
9	0.3	3	4500	10000.000	5000	5000.00
10	0.3	3	5000	11111.111	5000	6111.11
11	0.3	3	5500	12222.222	5000	7222.22
12	0.3	3	6000	13333.333	5000	8333.33
13	0.3	3	6500	14444.444	5000	9444.44
14	0.3	3	7000	15555.555	5000	10555.56
15	0.3	3	7500	16666.666	5000	11666.67
						58333.33

Figure 6: Spreadsheet depiction of MS Solver™ Add-In module with initial beam width, b(x), values.

The single target cell is established with a formula of the sum of this column of "deltas" and the Set Target Cell to Value of zero is established. The spreadsheet depicted below has the parameter b(x) set initially to 0.3 inches giving a sum of the Delta column as 58333.33 initially. Figure 7 below depicts the Solver™ parameter setup screen, where the target cell, parameter cells, and constraints are established. In the screenshots that follow, cell H51 corresponded to the sum of the Delta column or as shown in Figure 6 above, 58333.33.

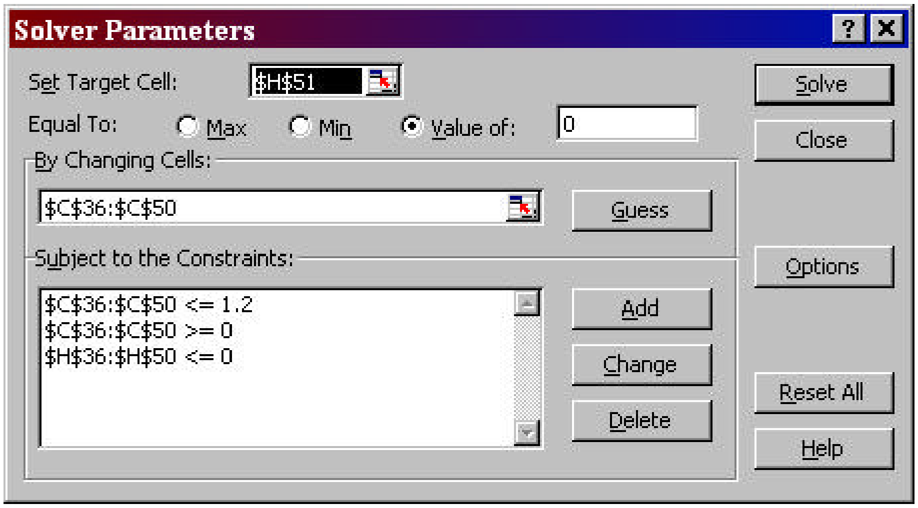


Figure 7: Solver™ Parameters for driving the sum of the deltas in cell H51 towards zero while varying cells C36 through C50 subject to the constraints shown.

MS Solver™ also provides the ability to set options such as Maximum Run Time, Maximum Number of Iterations, Precision, etc. using the Options button in the Parameter window as shown in Figure 8 below.

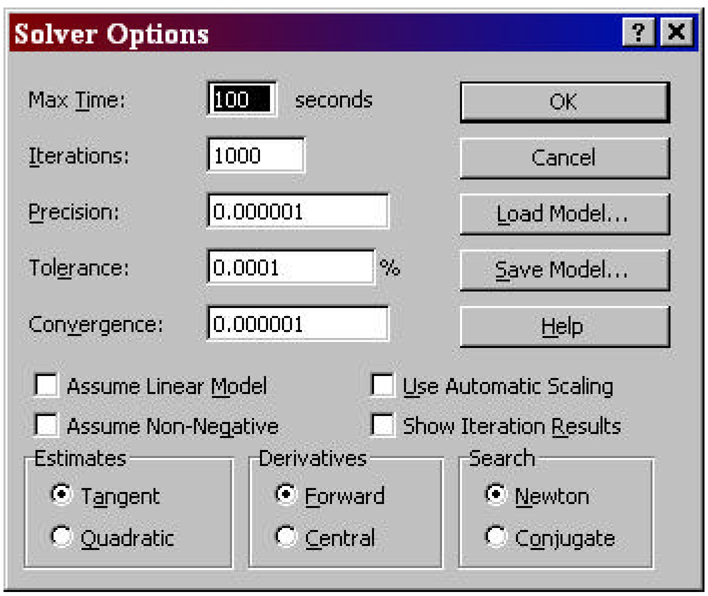


Figure 8: Solver™ Options to support driving the sum of the deltas towards zero

Several charts of the MS Solver™ outputs during and after optimization are shown below in Figure 9 with the trend clearly towards the exact solution of a linear variation of the beam width, b(x), with respect to x to produce a constant stress state in the simply supported beam with a central load.

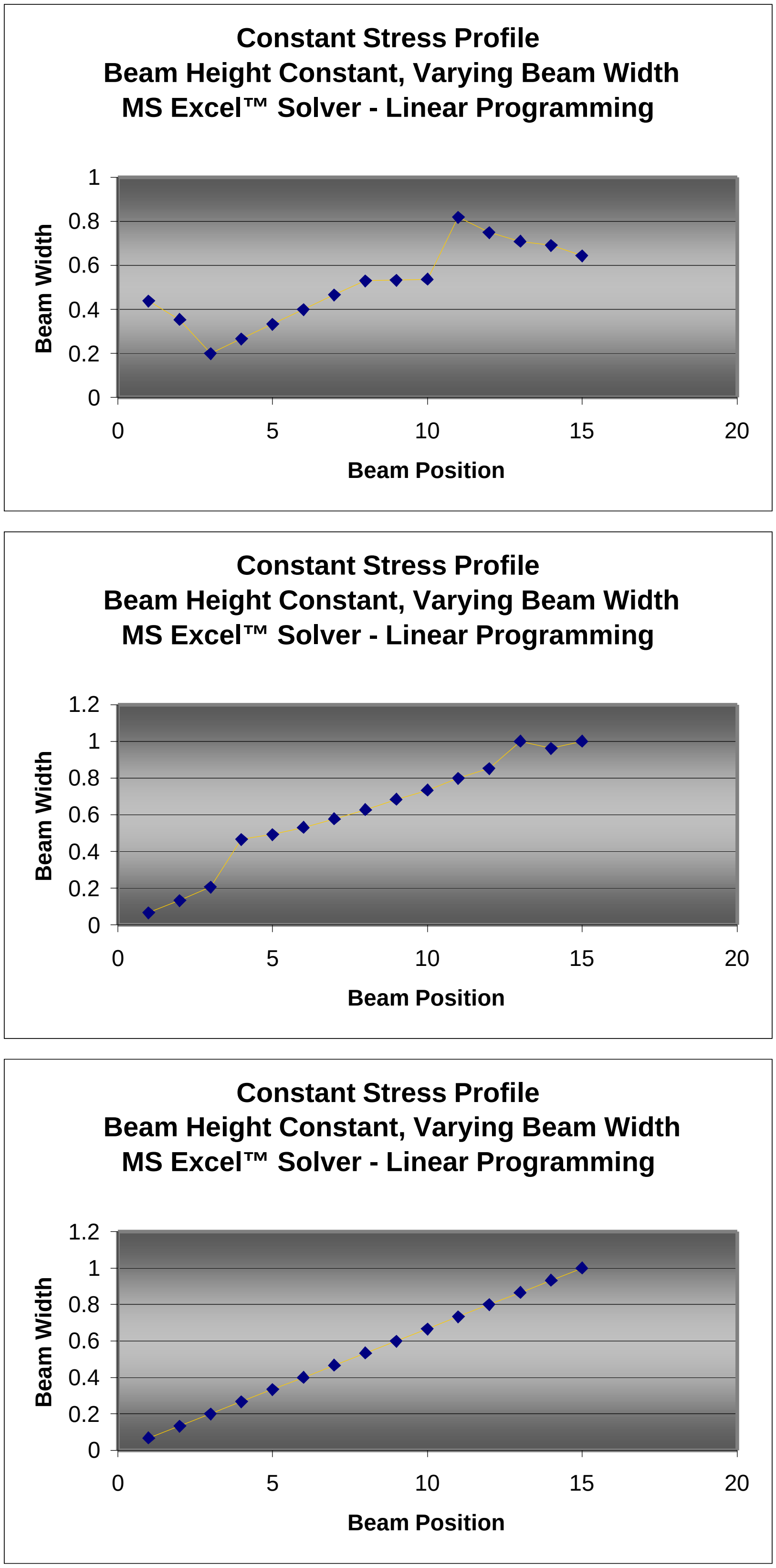


Figure 9: Three MS Solver™ output charts of beam width, b(x), versus beam position, x, for constant stress beam

4) Genetic Algorithms Method:

Difficult optimization problems (mathematically discontinuous, multiple objective, etc.) can be addressed by genetic algorithms (GAs). GAs are based on the pioneering work in population evolution of John Holland at the University of Michigan and one of his graduate students, David Goldberg[7]. GAs use a probability-based process for generating arbitrary size initial populations of possible solutions, which are evaluated in a survival-of-the-fittest, evolutionary methodology. Subsequent generations of possible solutions are derived from the best of the first generation results and allow for robust optimization of a wide variety of problems. Cross-over, mutation, elitism, and diversity work are terms from genetics that describe the GA's operation, however the details of these GA-related terms well-covered in the literature and will not be repeated here.

As mentioned earlier, typical optimization problems, have a single optimization objective function and can be addressed using other established methods. GAs are well-suited to problems that are mathematically difficult to model, because they search such a broad spectrum of potential solutions in an efficient manner. The theory of GAs is very briefly described in the upper and lower division mechanics classes, with more time spent on the application of GAs, specifically GeneHunter™ from Ward Systems Group[9]. The application similarities to and differences from MS Solver™ are pointed out in lecture.

For this same optimization problem, GeneHunter™ is utilized as an Add-In to MS Excel™. The fitness function (another name for the optimization objective) for the sum of the individual deltas is set in a target cell. As with MS Solver™, the goal is to drive this fitness function to a Value of zero. The adjustable parameters are the values of beam width, b(x), corresponding to each value of x in the cells of the spreadsheet. In this example, cells C17 through C31 are adjustable. The ranges for the values of b(x) are also specified as shown below in Figure 10 below. The search space for the beam width, b(x), includes an extended range that is 20% above the established upper limit of 1.0", allowing both Solver™ and GeneHunter™ to better converge.

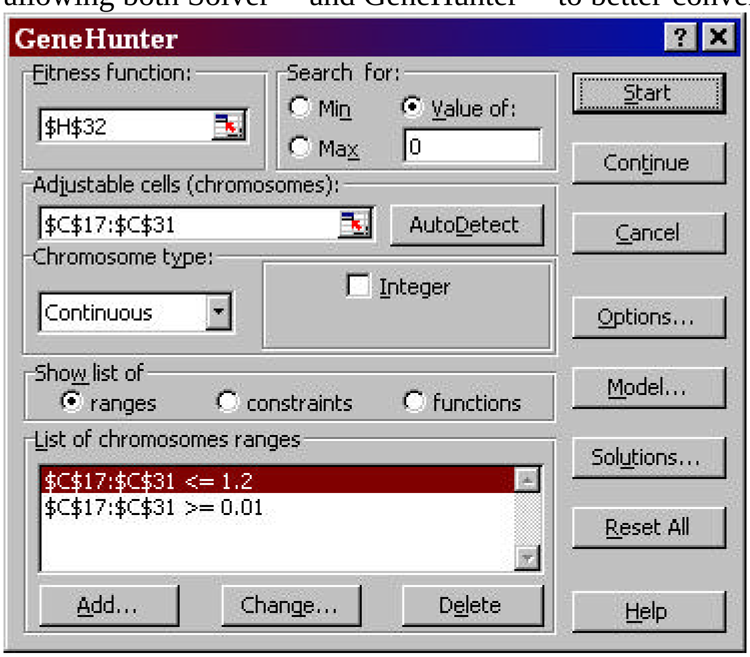


Figure 10: Genetic algorithm set-up screen using Ward System Group's GeneHunter™ to produce values of beam width, b(x), for a constant stress beam with cell H32 representing the sum of the deltas

The constraints of the intermediate parameters are specified in the GeneHunter™ Constraint screen. In this case, the stress constraint is set to be less than or equal to the upper stress limit with a high priority. The individual deltas are set also constrained to be less than or equal to zero. This latter constraint drives the solution from a safer, lower stress perspective. The constraints are shown in Figure 11 below.

Figure 11: Genetic algorithms method with constraints for stress (≤G17) and deltas (<= 0)

Genetic algorithms parameters specific to the optimization problem are set in the GeneHunter™ options screen shown in Figure 12 below. In this case, a population of 100 with a 16-bit long chromosome, a 90% cross-over probability, a 1% mutation rate, and a 98% generation gap are set. An elitist strategy is utilized where it is guaranteed that the top 2% of the chromosomes from the previous generation will be carried over to the next generation. Setting the generation gap to 98% dictates that 100% minus 98% (or 2%) will be carried forward.

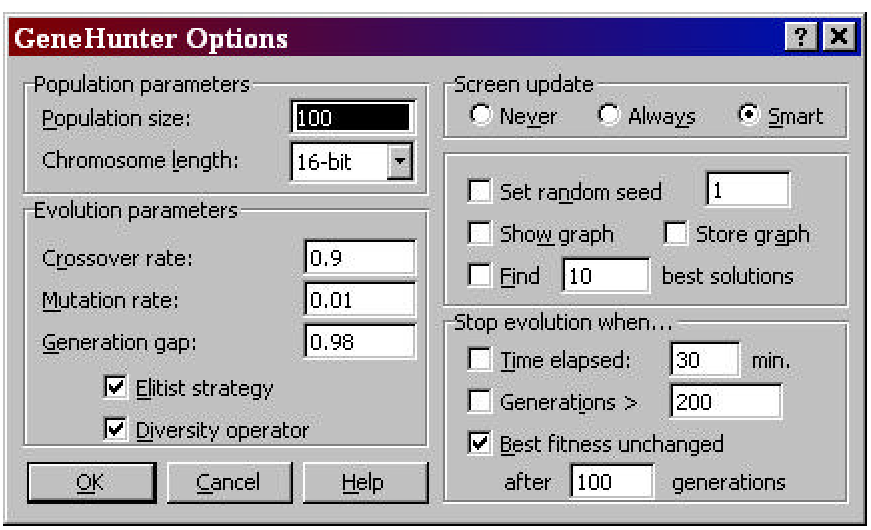
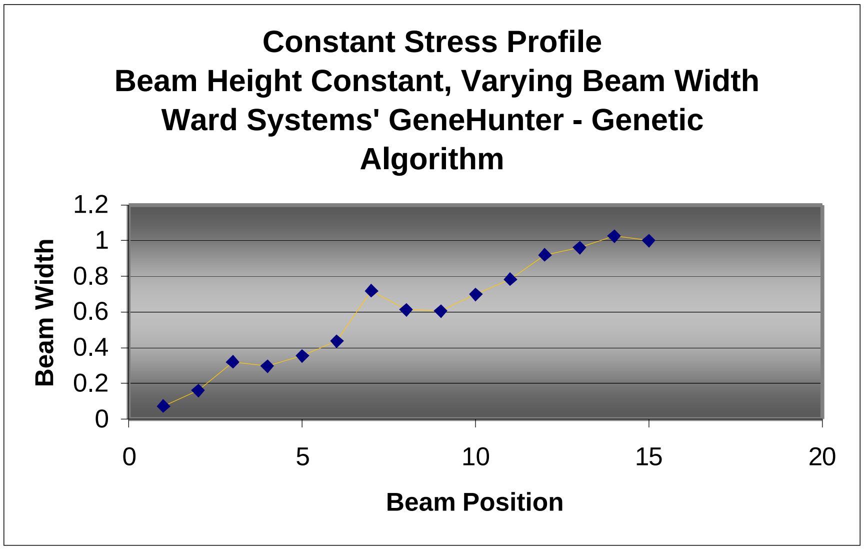


Figure 12: Genetic algorithms parameter screen

Several charts of the GeneHunter™ output during and after optimization are shown below in Figure 13 with the trend clearly towards the exact solution of a linear variation of b(x) with respect to x to produce a constant stress state in the simply supported beam with a central load.



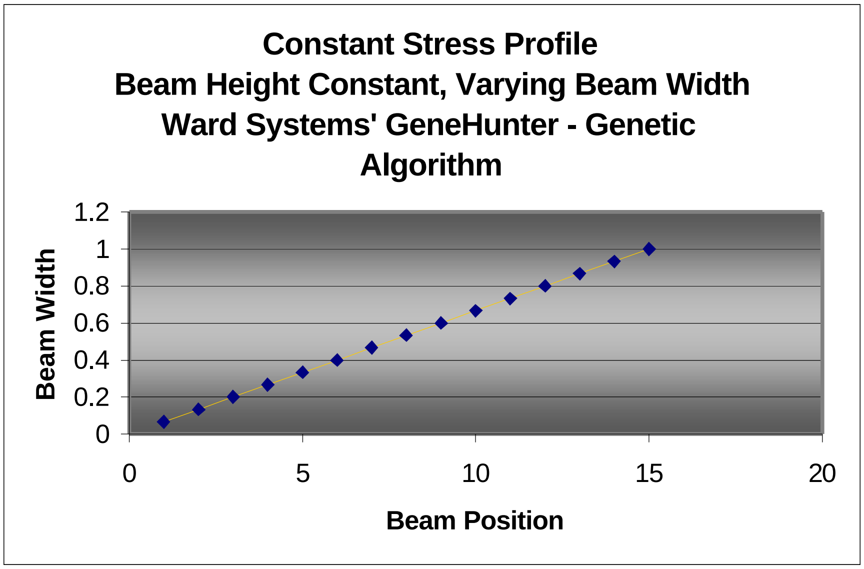


Figure 13: Two GeneHunter™ output charts for beam width, b(x), for constant stress beam

Conclusions:

A learning progression for introducing single criterion optimization methods into mechanics classes has been presented in this paper. The progression from a simply supported beam with a central load to a simply supported beam with a distributed load to the overhung beam with a distributed load was attempted with very good meeting of course learning objectives. The simply supported beam with a central load was utilized to present the various solution methods in detail. This progression of mechanics problems enables students to effectively utilize pre-existing lower-division mechanics course lecture material and experimental procedures to apply the optimization methods.

The author observed that introducing optimization broadened and enriched both their interest in and understanding of these problems. Student comments and ratings were very favorable. Generally the students appreciated the high degree of applicability of these methods to their future coursework and to their future profession. The author is exploring means to introduce the theory and practice of genetic algorithms to a wider audience of students through a web-based tutorial. Without the lecture time to develop the underlying theory of genetic algorithms, one top student commented that the GA method was "pretty black box" and spent office hour time understanding the underlying theory.

The above listed problems provide a series of increasingly-complex, single-criterion optimization problems that dovetail very well with the present lower-division engineering technology mechanics course lecture and laboratory learning objectives. A similar sequence is planned to be utilized in an upper-division mechanics course with additional project requirements (such as more extensive experimental and finite element analysis verification of theory).

Steuer[6] provides an extensive theoretical, computational, and applied treatment of multiple-criteria decision making (MCDM) problems. This MCDM class of optimization problem is planned to be introduced and utilized as part of a possible final project in the upper-division experimental mechanics course. MCDM problems could include such competing objective functions as to provide a target for a simply supported beam's mid-span deflection as a first objective function while also providing for a near-constant-stress second objective function. Many other MCDM's applicable to upper-division mechanics courses are possible and under consideration.

This paper has reported on the beginning of a process to introduce optimization theory and have the lower- division and upper-division engineering technology students seriously utilize a broad array of optimization methods. Future applied research in mechanics optimization will focus on devising optimization projects for students with non-linear elements as well as with multiple, varied elements and multiple criteria. These optimization problems will be introduced to assist driving capability deeper into the mechanics curriculum.

Bibliography:

1. Shigley, J.E, Mischke, C.R., (2001), Mechanical Engineering Design (6th ed.), N.Y., New York: McGraw-Hill.
2. Mott, R.L. (2001). Applied Strength of Materials (4th ed.). Englewood Cliffs, NJ: Prentice-Hall.
3. Arora, J.S. (1989), Introduction to Optimum Design, McGraw-Hill Inc., N.Y., New York.
4. Siddall, James N. (1982), Optimal Engineering Design, Marcel-Dekker, Inc., N.Y., New York.
5. Chong, E.K.P., Zak, S.H. (2001), An Introduction to Optimization, John Wiley & Sons, Inc., N.Y., New York.
6. Steuer, R.E (1986), Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley & Sons, Inc., N.Y., New York.
7. Goldberg, D.E., Genetic algorithms in search, optimization, and machine learning, Addison Wesley Longman, Inc., Reading, Massachusetts
8. Microsoft, Inc website (January, 2002), http://www.microsoft.com, Redmond, Washington
9. Ward Systems Group website (January, 2002), http://www.ward.net, Frederick, Maryland.

Biography:

WILLIAM K. SZAROLETTA, P.E.
Professor Szaroletta is an assistant professor of mechanical engineering technology at Purdue University. A member of ASEE, he has 18 years industry experience in engineering and project management positions, with 12 awarded patents. He received his B.S. Degree in Mechanical Engineering from University of Michigan, Ann Arbor in 1977, M.S. Degree in Engineering (Product Design) from Stanford University in 1984, and a Masters in Applied Mathematical Sciences Degree (Computer Science) from University of Georgia in 2000. He has 6 years university teaching experience, where his current applied research interests are rapid product design engineering, experimental mechanics laboratory automation, and optimization utilizing genetic algorithms.