CROSS-SECTION OPTIMISATION FOR COLD-ROLLED STEEL BEAMS USING A GENETIC ALGORITHM

WOLF Christoph, STADLER Anna Theresia, BAUMGARTNER Werner

Johannes Kepler University of Linz (JKU), Institute of Biomedical Mechatronics, Linz, Austria, EU
christoph.wolf@jku.at

Abstract

Since the use of cold-rolled steel sections is a standard method in mechanical engineering and steelwork and cost efficiency is always a big issue, it is of great interest to address the highly non convex problem of cross-section optimisation. At a first glance classic genetic algorithms already seem to be suitable for this problem, because they are powerful search heuristics for solutions in non convex problems. But in addition to the great number of local optima, which can occur in the optimisation process, the problem is also constrained due to constructive needs and limitations in manufacturing. Constraints highly affect the ability of genetic algorithms to overcome local optima, especially under high selection pressure. This high selection pressure comes from the need to suit the given use case in terms of physical stability. Therefore we had to handle the restrictions more flexible, so that the algorithm can temporarily violate our stability criteria to overcome a local optimum, but will end up with a solution within our given boundaries. To achieve this we negatively coupled the penalty factor for stability criteria violations to the mutation strength, thus allowing adaptive radiation with rather free-wheeling restriction handling, followed by a rigid selection process approaching the optimal solution. Additionally we introduced an inbreed avoiding recombination system to speed-up the exploration of the fitness landscape. This yielded material savings of about 20 %. To speed up the method, parallelisation was applied and the algorithm could be implemented on a 60-core Linux cluster.

Keywords: Cold forming, evolution, restricted optimisation, parallel computing, cluster computing

1. INTRODUCTION

In modern mechanical engineering and steelwork the use of cold-rolled (roll-formed) steel sections is a standard method [1]. These sections should be mechanically stable on the one hand and cost efficient on the other hand. To decide what profile suits for a certain case is a constrained optimisation problem which is in general non convex, i.e. several local optima exist. It is a problem of restricted optimisation as such a profile has to meet several requirements with respect to the mechanical load. Parameters like moments of inertia, moments of resistance, torque of inertia, buckling-stability, slenderness ratio and the radii of inertia have to be in a range suitable for the material to withstand the loads. These mechanical constrains are typically handled using Lagrange-multiplyers, penalty-functions, barrier-functions, combinations of the former or methods eliminating the degrees of freedom [2, 3]. Finding the global optimum for such a problem is a hard task in general and there is no method that guarantees success. Besides great deluge algorithm, simulated annealing, Metropolis algorithm, threshold acceptance, hill climbing, ant algorithm, stochastic tunnelling and RANSAC-algorithm the most promising and wide used approach is that of the genetic algorithm or evolutionary algorithm. A genetic algorithm is a search heuristic that mimics the process of natural evolution [4-7]. This heuristics can be used to generate useful solutions to optimisation and search problems. In a genetic algorithm, a population of individuals (candidate solutions) is evolved toward better solutions with respect to a fitness function. Not surprisingly there were several attempts in the past to optimise cold formed steel profiles using genetic algorithms [8-13]. However it turned out to be rather complicated to optimise the general cross section but still overcome local optima. We recently established a genetic algorithm for this problem which uses some special adaptations in order to solve this special problem of free form optimisation [14]. We found that variable selection pressure implemented by the rule of 1/5 [6] is absolutely necessary. Highly advantageous is sexual recombination including an inbreed avoiding mate selection based on the genetic distance. We found a flexible handling of the mechanical restrictions, coupled to the mutation strength to be vital for overcoming local optima. It is well established, that separation of subpopulations of a species can lead to divergent evolution and therefore to the development of new species. In the case of genetic algorithms, this approach is also possible and improves the search in the parameter space. Here we show how our genetic algorithm can be parallelised for the implementation of subpopulations and therefore divergent evolution. The run time is reduced significantly when compared to the simple sequential calculation on a single high speed core.

2. METHODS

2.1. Basic algorithm

The general algorithm was described recently [14]. Briefly, a profile having a finite number of local bends is described by a vector g of the x- and y- coordinates of all these bends

$$g = \begin{pmatrix} x \\ y \end{pmatrix} = \left(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n\right)^T \qquad (1)$$

with *x* representing the vector of all x-coordinates and *y* being the vector of all y-coordinates of the bends 1.*n*. Between the bends the tin is flat. For the reproduction an individual is selected and a copy of the genome is taken and subjected to mutation (see below) for the next generation. In the case of sexual recombination, the genomes of two individuals are combined (for details see [14]). The mutation of an individual *i*, i.e. a profile represented by g_i can be constructed by adding a vector of normal distributed values of the corresponding size to g_i as

$$g_i(t+1) = g_i(t) + N \cdot \mu = g_i(t) + \mu \cdot \begin{pmatrix} N_1 \cdot s_1 \\ N_2 \cdot s_2 \\ \vdots \\ N_{2n} \cdot s_{2n} \end{pmatrix} \quad with \quad N_i : N(0,1) \quad \wedge \quad s_i \in \{0,1\} \qquad (2)$$

so the 2*n* values of N_i are normal distributed random numbers with mean 0 and standard deviation 1. The values s_i which can be 0 or 1 represent the protection of the end coordinates; if for example the x-coordinate of a bend *j* must not be changed due to constructive demands, s_j is 0. The value *μ* is the mutation strength which was found to be a rather critical parameter. The mutation strength *μ* must be adapted continuously during the application of the algorithm. Therefore we applied the 1/5-method established by Rechenberg [6]. If the number of individuals in the generation *t*+1 that have higher fitness than their parents is significantly higher than 1/5th of the total number of individuals in this generation, the mutation rate is increased by a factor of 10. If on the other hand the better individuals are below 1/5th of the total number, the mutation strength is reduced by a factor of 3. These values were empirically obtained to yield good results. The fitness can be initially calculated as 1 / *A* where *A* is the cross section area of the profile, if the weight has to be minimised. To assure mechanical stability calculations for beams on two supports according to the DIN EN 1993 (Eurocode 3) were established. For given restrictions concerning deformations and carrying capacity, parameters of the profile like moments of inertia, the moments of resistance, torque of inertia, buckling-stability, slenderness ratio and radii of inertia were calculated for each profile under consideration. Dependent on the state of the evolutionary optimisation, i.e. on the mutation strength μ, violations of the stability criteria are allowed. For example with respect to the area moment of inertia around the x-axis I_x is calculated and compared to the minimal area moment of inertia $I_{x,min}$. Then a weighting factor h_{IX} is calculated according to

$$h_{IX} = min\left\{1, \exp\left[V \cdot \frac{I_x - I_{x,min}}{I_{x,min} \cdot \mu}\right]\right\} \tag{3}$$

with V being a steepness factor. A value of V equal one over ten times the number of variable entries in the genome was found empirically to be well suited. However, this factor might need individual adaptation by the operator to yield good results. Based on the different factors h_i representing the different mechanical constraints the actual fitness function can be calculated as

$$F = \frac{1}{A} \cdot \prod_i h_i \tag{4}$$

Finally the selection was done randomly based on selection probabilities [14]. For the implementation of separated subpopulations the algorithm is started several times with initial conditions which are identical besides the seed setting of the random number generator. Typically 30 independent subpopulations were used.

2.2. Implementation on a 60-core Linux cluster

The algorithm was ported to a standard beowulf-type cluster with 60 CPU cores, to explore the benefits of massive parallelisation for this specific application. The cluster is composed of 14 IBM bladeservers (HS20) with 2 dualcore Intel Xeon CPUs and 4GB of DDR2 memory each as worker-nodes and one IBM XServer 3650 with 2 Intel Xeon dualcore CPUs of a more recent generation and also 4GB of DDR2-RAM as the headnode. The Layout of the cluster environment is rather traditional. The headnode has 2 network interfaces, one connected to the university intranet and one connected to the worker-nodes. The gigabit switch module of the bladecenter is used to interconnect the worker-nodes and the headnode in a private network environment. All nodes are running Ubuntu Server 14.04LTS and use OpenMPI for parallel-process-communication. Octave version 3.8.1 with the package 'mpi' was used as the runtime environment for the algorithm. A shared network filesystem is provided by the headnode, so that every worker-node can get the algorithm and the data-files directly from the HDD of the headnode. The headnode also provides a network-wide user authentication system and network time synchronisation, so that everything is consistent between the individual machines. Four versions of the algorithm were run: Parallel processing with forced symmetry of the profiles (mirrored on the Y-axis at $X = 0$), parallel processing without forced symmetry and sequential processing with and without forced symmetry. For the parallel processing 30 CPU-cores were used to run the optimization for one population each. At the end of the evolution process the best individual out of those 30 populations was picked. The sequential calculations were done on one CPU-core repeating the optimization process 30 times and then picking the best result out of those thirty runs. Computation times were saved to the results file to calculate the speed-up, which parallelisation actually yields.

3. RESULTS AND DISCUSSION

The efficiency of the algorithm, especially of our flexible penalty which tolerates violations of the restricting conditions dependent on the mutation strength was exemplified recently [14]. It can be also shown here when optimising a profile for a rail of a crane (**Figure 1A** and **B**). The profile has to fulfil the following conditions: The area moments of inertia have to be above I_{ymin} = 300 cm^4 and I_{zmin} = 360 cm^4; the section modulus of torsion has to be above I_{tmin} = 0.1 cm^3; the section moduli have to be above W_{zmin} = 35 cm^3 and W_{ymin} = 35 cm^3 respectively. The radii of inertia are demanded to be above r_{tymin} = 25 cm and r_{tzmin} = 25 cm in order to avoid

Euler-bending and buckling. Geometric restrictions were also given: For bolting the rail and for connection with a beam hoist the sections marked in **Figure 1A** with ellipses have to be protected, i.e. their positions must not be altered. If this profile, determined by its centre line (**Figure 1B**), is then subjected to an evolution algorithm with the strict penalty function which does not allow any violation of the above condition, the optimisation stops at a hardly improved profile. Especially the two folds on the upper sides persist, as any change of the tip coordinates will either increase the cross section area *A*, i.e. reduce the fitness or violate the mechanical restrictions. The advantages of the adaptive penalty functions were shown in detail in [14]. Two principally different results are shown here. In **Figure 1C**, no further restriction was made and a profile fulfilling all geometrical and mechanical restrictions was found which reduces the cross section area and thereby the weight by 21 % then related to the result. If additionally a symmetric profile is demanded, the profile depicted in **Figure 1D** can be found. The weight reduction is 18 %. Although this weight reduction is significantly smaller than in the asymmetric case, a symmetric profile might be more desirable than an asymmetric one. Generally it can be said, that restrictions of any mechanical or geometrical kind can be easily implemented in the way described here.

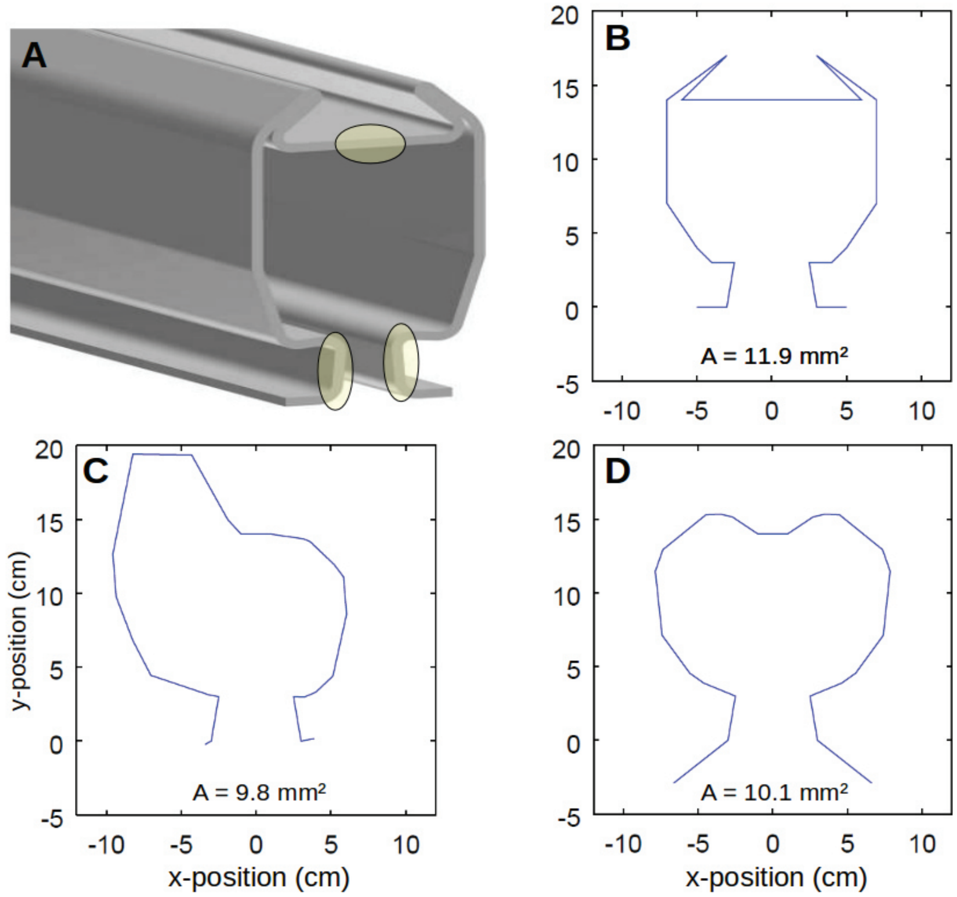

Figure 1 Optimisation example of the profile for a rail of a crane. A) Original profile where the protected areas are indicated by ellipses. Here connection parts have to be attached. B) Centre line of original profile showing the positions of the bends. C) Centre line of the optimised profile. D) Centre line of the optimised profile forcing symmetry. The cross section areas are indicated for a wall thickens of 1.75 mm.

The computation times on the cluster are shown in **Table 1**. These times were all measured on compute nodes. The parallelisation speed-up is 16.8 for symmetric and 13.3 for asymmetric profiles. We also carried out a run with the sequential algorithm on the headnode which took 16 minutes for symmetric and 25 minutes for asymmetric profiles. This reduction of computation times is due to better system performance (faster CPU) and reduced overhead (no network file system). Since the servers of which our cluster is composed are rather old we also tested the performance of the sequential computations on a standard office computer (intel core i5.8 GB of DDR3-RAM, SATA3 SSD) and got the calculations for symmetric and asymmetric profiles done within 17 minutes in total. This shows that calculation times can be extremely reduced through parallelisation and modern hardware. Thus we can expect a speed-up of at least factor ten so on a convenience cluster formed by modern office PCs with 30 cores in total the parallel computations would be done in less than two minutes. Previous work with a parallel computation on a 6-core office computer [14] has already shown, that these expectations are realistic. Thinking of more complicated load cases or calculations for a whole structure, rather than a single beam, the speed-up of massive parallelisation may become even more significant, as the ratio between overhead and actual calculation times improves for larger problems [15].

Table 1 Computation times on the cluster

-	Asymmetric	Symmetric
Parallel	3.9 min	2.2 min
Sequential	52 min	37 min

4. CONCLUSION

We were able to implement an optimisation algorithm to the highly non-convex problem of cross section optimisation of steel beams based on evolutionary algorithms. A new adjustable penalty function to calculate the fitness was established. This new function is superior to the standard penalty function as escapes from local optima are made possible. For a more complete search of the parameter space isolation of several subpopulations was implemented. This can be parallelised easily on a computer cluster. Taken together we found a biomimetic approach to optimise cold-rolled steel beam profiles in order to exploit the material almost ideal for obtaining cost efficient and producible sections which can bear the specified mechanical load. Different restrictions can be implemented easily, thus for example the results of finite element (FEM) simulations can be used as constraints. The values from the FEM can be compared to the allowed values and the weighting factors h_i for the calculation of the corrected fitness function can be obtained. Due to the simple way to parallelise the algorithm, this can be done with reasonable calculation times. Thus, the algorithm described could help saving resources and energy as well as costs in mechanical engineering and structural steel work in the future.

ACKNOWLEDGEMENTS

The authors thank the framework of the Austrian Center of Competence in Mechatronics (LCM) and the voestalpine Finaltechnik Krems for financial support.

REFERENCES

[1] HALMOS, G.T. *Roll forming handbook*. 1st edition. Taylor & Francis. USA 2005 584 p.

[2] SUN, W., YUAN, Y. *Optimisation Theory and Methods*. Nonlinear Programming. 4th edition. Springer Science+Business Media LLC, USA 2006 688 p.

[3] KUHN, H.W., TUCKER, A.W. *Nonlinear Programming*. In Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 31 July - 12 August, USA. Edited by Neyman J. USA: University of California Press. 1951 481-492.

[4] HOLLAND, J.H. *Adaptation in natural and artificial systems. An introductory analysis with applications to biology, control, and artificial intelligence*, 1st edition. MIT Press, USA 1995 211 p.

[5] GEN, M., CHENG, R. (2000) *Genetic algorithms and engineering optimisation. In Wiley Series in Engineering design and automation*, 1st edition. Edited by Hamid RP, Wiley, USA 495 p.

[6] RECHENBERG, I., EIGEN, M. *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. 1st edition. Frommann-Holzboog, Germany 1973 170 p.

[7] HAUPT RL, Haupt SE (2004) *Practical genetic algorithms*, 2nd edition. Wiley-Interscience, USA 272 p.

[8] LU, W., MÄKELÄINEN, P. *Augmented Lagriangian genetic algorithms for optimal design of hat-shaped cold-formed steel profile*. In 9th international conference: Modern building materials, structures and techniques, 16-18 May, Lithuania. Edited by Skibniewski MJ, Vainiunas P, Zavadskas EK. Lithuania Vilnius Gedinimus Technical Univ Press, Technika 2009 p.:998-1004.

[9] LU, W., MÄKELÄINEN, P. Fuzzy optimisation of cold-formed steel sheeting using genetic algorithms. *Journal of Constructional Steel Research.* 2006, Vol. 62, pp. 1276-1281.

[10] LU, W. *Optimum design of cold-formed steel purlins using genetic algorithms*. PhD thesis, Helsinki University of Technology, Espoo 2003.

[11] LEE, J., KIM, S., PARK, H., WOO, B. Optimum design of cold-formed steel channel beams using micro Genetic Algorithm. *Engineering Structures.* 2005 Vol. 27, pp. 17-24.

[12] LEE, J., KIM, S., SEON PARK, H. Optimum design of cold-formed steel columns by using micro genetic algorithms. *Thin-Walled Structures.* 2006, Vol. 44, pp. 952-960.

[13] GRIFFITS, D., MILES, J. Determining the optimal cross-section of beams. *Advanced Engineering Informatics* 2003, Vol. 17, pp. 59-76.

[14] ESTERHAMMER, F., WOLF, C., STADLER, A.T., BAUMGARTNER, W. Optimizing the Cross Section of Cold-Rolled Steel Beams Using a Genetic Algorithm: Avoiding Local Optima Using Adaptive Mutation Control, Flexible Restriction Handling and Inbreed Avoiding Mating Strategies. *Peertechz J Comput Sci Eng.* 2016, Vol. 1: pp. 001-011.

[15] BAUKE, H., MERTENS, S. *Cluster Computing.* Berlin: Springer, 2006. pp. 10-18.