HU ISSN 1586–2070

JOURNAL OF COMPUTATIONAL AND APPLIED MECHANICS

A Publication of the University of Miskolc

VOLUME 6, NUMBER 2 (2005)

MISKOLC UNIVERSITY PRESS

EDITORIAL BOARD

LOCAL EDITORIAL COUNCIL

Journal of Computational and Applied Mechanics, Vol. 6., No. 2., (2005), pp. 237–256

PARTICLE SWARM METHOD AS A NEW TOOL FOR STRUCTURAL OPTIMIZATION

KÁROLY JÁRMAI
Department of Materials Handling and Logistics, University of Miskolc
3515 Miskolc-Egyetemváros, Hungary
altjar@uni-miskolc.hu

[Received: March 25, 2004]

Dedicated to István Páczelt on the occasion of his 65th birthday

Abstract. A new and promising optimization technique is introduced: the particle swarm optimization (PSO). In this evolutionary technique the social behavior of birds is imitated. The technique is modified in order to be efficient in technical applications. It calculates discrete optima, uses dynamic inertia reduction and craziness at some particles. The efficiency of the technique is shown in the optimum design of a stringer-stiffened shell under bending and compression. The PSO is built into an interactive program system, where several optimization techniques are employed. The program system includes multiobjective optimization techniques as well. Results show that PSO is a reliable and robust technique to find optima with highly non-linear constraints. In the cost calculation 2D and 3D curve fitting is employed to determine the production time.

Mathematical Subject Classification: 74P10
Keywords: particle swarm method, structural optimization

1. Introduction

The optimum design process has the following three main phases:

- preparation: selection of candidate structural versions defining the main characteristics to be changed, formulation of design constraints and cost function,
- solution of the constrained function minimization problem by using efficient mathematical methods,
- evaluation of results by designers, comparison of optimised versions, formulation of design rules, incorporation in expert systems.

These phases show that structural optimization has the following three main parts: cost function, design constraints, and mathematical method.

In this paper we focus on the mathematical technique and show its application.

There is a great number of methods available for single objective optimization as it was described in Farkas & Jármai [1]. Methods without derivatives include: Complex [2], Flexible Tolerance, and Hillclimb. Methods with first derivatives include: Sequential Unconstrained Minimization Technique (SUMT), Davidon-Fletcher-Powell, etc. Methods with second derivatives include: Newton, SQP. There are also other classes of techniques like optimality criteria methods, or the discrete methods like Backtrack, the entropy-based method [3, 4]. Multicriteria optimization is used when several objectives are important to find the compromise solution [5].

The general formulation of a single-criterion non-linear programming problem is the following:
minimize

$$f(x) \quad x = \{x_1, x_2, ..., x_N\} \tag{1.1}$$

subject to

$$g_j(x) \leq 0, \qquad j = 1, 2, ..., P \tag{1.2}$$

$$h_i(x) = 0, \quad i = P+1, ..., P+M \tag{1.3}$$

$f(x)$ is a multivariable non-linear function, $g_j(x)$ and $h_i(x)$ are non-linear inequality and equality constraints, respectively.

In the last two decades some new techniques have appeared, e.g. the evolutionary techniques, the genetic algorithm [6], the differential evolution technique [7, 8], the particle swarm algorithm [9], and the ant colony technique [10, 11].

Some other high performance techniques such as leap-frog with the analogy of potential energy minimum [12, 13, 14], similar to the FEM technique, have also been developed.

2. The particle swarm algorithm

2.1. Preliminary remarks. A number of scientists have created computer simulations of various interpretations of the movement of organisms in a bird flock or fish school [15]. The Particle Swarm Optimization (PSO) algorithm was first introduced

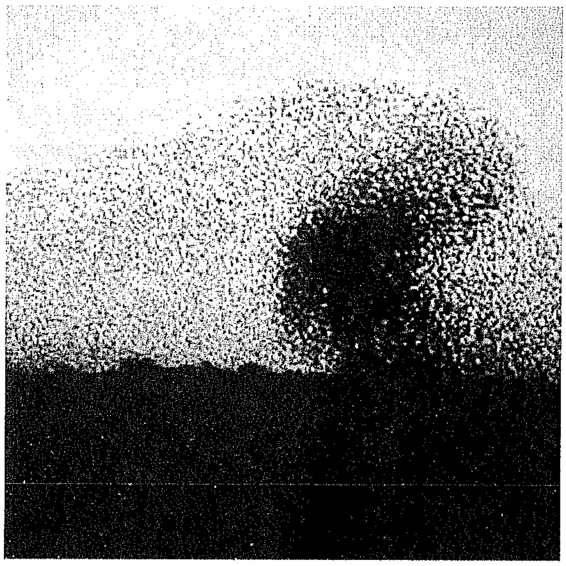

Figure 1. Bird swarm

by Kennedy [16]. The algorithm models the exploration of a problem space by a population of individuals; the success of each individual influences their searches and those of their peers. In our implementation of the PSO, the social behavior of birds is imitated. Individual birds exchange information about their position, velocity and fitness, and the behavior of the flock is then influenced to increase the probability of migration to regions of high fitness [9]. A bird swarm is visible in Figure 1.

Particle swarm optimization has its roots in two main component methodologies. Perhaps more obvious are its ties to artificial life in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming. Particle Swarm optimizers are similar to genetic algorithms in that they have some kind of fitness measure and start with a population of potential solutions (none of which are likely to be optimal), and attempt to generate a population containing fitter members.

In theory at least, individual members of the school can profit from the discoveries and previous experience of all other members of the school during the search for food. This advantage can become decisive, outweighing the disadvantages of competition for food items, whenever the resource is unpredictably distributed in patches. Social sharing of information among conspeciates offers an evolutionary advantage: this hypothesis was fundamental to the development of particle swarm optimization.

Millonas [15] developed his models for applications in artificial life, and articulated five basic principles of swarm intelligence. The first one is the proximity principle: the population should be able to carry out simple space and time computations. The second one is the quality principle: the population should be able to respond to quality factors in the environment. The third one is the principle of diverse response: the population should not perform its activities along excessively narrow channels. The fourth one is the principle of stability: the population should not change its mode of behavior every time the environment changes. The fifth one is the principle of adaptability: the population must be able to change its behavior mode when it is worth the computational price.

Basic to the paradigm are n-dimensional space calculations carried out over a series of time steps. The population is responding to the quality factors pBest and gBest (gBest is the overall best value, pBest is the best value for a particle). The allocation of responses between pBest and gBest ensures a diversity of response. The population changes its state (mode of behavior) only when gBest changes, thus adhering to the principle of stability. The population is adaptive because it does change when gBest changes.

The method is derivative free, and by its very nature the method is able to locate the global optimum of an objective function. Constrained problems can simply be accommodated using penalty methods.

2.2. **Description of the Particle Swarm Algorithm.** The system is initialized with a population of random potential solutions. Each potential solution is assigned a randomized ´velocity' and is called a particle. (It has position in the space, i.e. it is a point in the solution space and it has a velocity. So it is analogous to a particle in physics which flies around in 3-D space.)

These particles are then 'flown' through the (hyper) space of potential solutions.

Each particle keeps track of the coordinates in the hyperspace for which it has achieved the best solution and its best fitness (call it pBest) so far.

In the ´global' version of the optimiser gBest is the overall best value with its location. This particle is the leader.

At each time step the 'velocity' of each particle is changed (accelerated) towards its pBest and gBest fellows. This acceleration is weighted by a random term. The idea is that all the particles swarm towards where the current best solutions are. The random factor prevents the swarm getting stuck in the wrong place – insects around a light.

A new position in the solution space is calculated for each particle by adding the new velocity value to each component of the particle's position vector.

The user specifies an acceleration constant and a maximum velocity.

Eventually, the swarm of potential solutions hovers around the best solution position. In the case of a neural net, this best 'particle' would be the optimum set of weights. (The weights are the particle's coordinates in the weight hyperspace.)

This method of search is a rival of the Genetic Algorithm in finding reasonable solutions to NP-hard problems.

For a given particle, in the N dimension search space:

let $x = (x_1, ..., x_N)$ be its current position,
let $v = (v_1, ..., v_N)$ be its current velocity,
let $p_i = (p_{i,1}, ..., p_{i,N})$ be the best position it has found so far,
let $p_g = (p_{g,1}, ..., p_{g,N})$ be the best position found so far in its neighborhood.

In other words for each particle the following information is available:

(a) It has a position and a velocity,
(b) It knows its position, and the objective function value for this position,
(c) It knows its neighbors, best previous position and objective function value (variant: current position and objective function value),
(d) It remembers its best previous position.

From now on, to put (b) and (c) in a common frame, we consider that the ´neighborhood' of a particle includes this particle itself.

At each time step, the behavior of a given particle is a compromise between three possible choices: (1) Following its own way, (2) Going towards its best previous position, (3) Going towards the best neighbor's best previous position, or towards the best neighbor (variant).

Define the new velocity by

$$v_d = \left(v_1^d, ..., v_d^d, ..., v_N^d\right) , \tag{2.1}$$

with

$$v_d^d = c_1 v_d + \text{rand}(0, c_2)\,(p_{i,d} - x_d) + \text{rand}(0, c_3)\,(p_{g,d} - x_d) , \tag{2.2}$$

define the new position by

$$x_d = x + v_d . \tag{2.3}$$

For $d = 1$ to N

$$v_d = v_d + \varphi_1(p_{i,d} - x_d) + \varphi_2(p_{g,d} - x_d) \ , \tag{2.4}$$

where φ_1 and φ_2 are said to be 'random positive numbers', without indicating *when* the randomness occurs: *inside* the d-loop (case 1), or *before* the d–loop (case 2). Unfortunately, several authors do use case 1, but nevertheless conclude that the new velocity vector is globally defined by

$$v_d = c_1 v + \text{rand}(0, c_2)\,(p_i - x) + \text{rand}(0, c_3)\,(p_g - x) \ , \tag{2.5}$$

which is true only for case 2. In this case, all the possible vectors are in the same plane, as in case 1 they define a complete volume in the search space.

There are different versions of Particle Swarm Optimization algorithms, but they can all be seen from an information point of view: what kind of information each particle has access to, and how it uses it (Shi & Eberhard 1998a,b).

2.3. The three methods.

2.3.1. *Random search rPSO.* The dimensionality of the search space H is N. It is supposed that H is finite, so that for each dimension k there is a minimum value $x_{min,k}$ and a maximum one $x_{max,k}$. At a time step, the particle uses no (variable) information at all.

2.3.2. *Constricted version cPSO.* At each time step, the pieces of variable information a given particle knows and can transmit are:

- its current position $x(t)$ and the corresponding objective function value,
- its best position found so far, $p_i(t)$, and the corresponding objective function value.

2.3.3. *Adaptive version aPSO.* At each time step, the pieces of variable information a given particle knows and can transmit are:

- its current position $x(t)$ and the corresponding objective function value,
- its best position found so far, $p_i(t)$, and the corresponding objective function value,
- its previous position (to estimate its improvement),
- its neighborhood size,
- the swarm size (global information).

To summarize, depending on the algorithm, each particle knows:

- no (variable) information at all (rPSO),
- only local information (cPSO),
- a bit more local information and some global information (swarm size) (aPSO).

Previously, the PSO algorithm was applied to analytical test functions, mostly univariate or bivariate without constraints [17]. In addition, multimodal problem generators were described by Kennedy & Spears [17]. Kennedy [18] used the PSO as an optimization paradigm that simulates the ability of human societies to process knowledge. The algorithm models the exploration of a problem space by a population of individuals; individuals' successes influence their searches and those of their peers. There were attempts to improve the efficiency of the PSO by hybridising the algorithm with various other search methods [18, 19].

Lately, the PSO was successfully applied to the optimum shape and size design of structures by Fourie & Groenwold [20 , 21, 22]. An operator, namely craziness, was re-introduced, together with the use of dynamic varying maximum velocities and inertia. An attempt was also made to optimize the parameters associated with the various operators in the case of generally constrained non-linear mathematical programming problems [23].

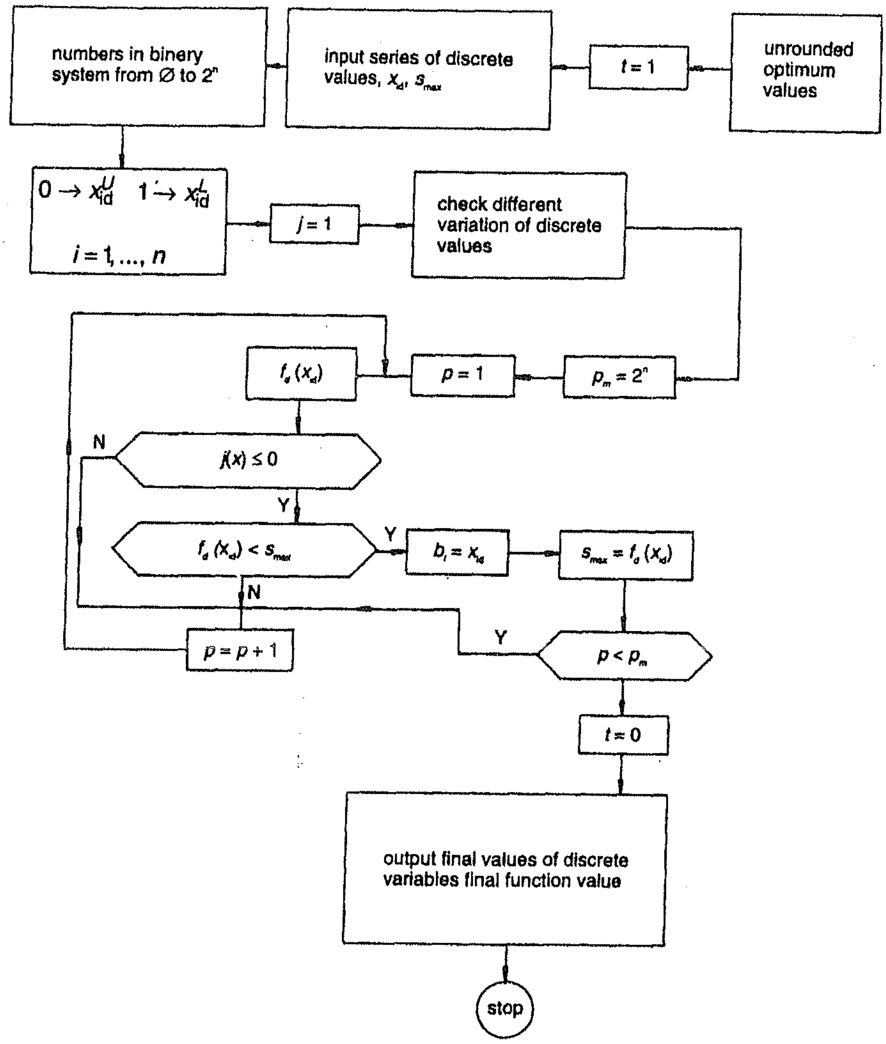

Figure 2. Flow chart of secondary discretization

3. Discretization after continuous optimization

To make the search more practicable it is advisable to use discrete member sizes. After continuous optimization a secondary search is necessary to find discrete optimum sizes in such a way that not only the explicit and implicit constraints are satisfied but the merit function takes its minimum as well. It is assumed that the optimum discrete sizes are near the optimal continuous ones [24].

Starting from the optimum continuous values, the secondary search chooses the nearest discrete sizes for each continuous size from the series of discrete values. The number of chosen discrete sizes for one continuous size can be two, three or more. The possible variations can be obtained using binary, ternary or larger systems. In our numerical example we use the binary system, two discrete sizes, upper and lower, belonging to one continuous value. In a binary system the figure zero means the upper discrete size, the figure one means the lower one. The first $2n$ number in a binary system gives all possible variations. Each variation is tested, whether the explicit and implicit constraints are satisfied, and the optimal values minimizing the merit function are determined. The flow chart of secondary discretization can be seen in Figure 2.

The number 0000 means the lower discrete values of all variables, the number 1111 means the upper discrete values of all variables. The other numbers in the binary system are the variants of the possible discrete solution. The solution is the tested variant that gives the minimum objective function value.

4. Stringer stiffened cylindrical shell loaded by axial compression and bending

4.1. **Aim and variables.** The aim of the optimization is to find the minimum value of the cost function due to non-linear constraints. The variables are: the height of stiffener h_s, number of stiffener n_s and the thickness of the shell t.

4.2. **Constraints.**

4.2.1. *Shell buckling (unstiffened curved panel buckling) DNV (1995)[25].* The stresses caused by compression and bending are as follows

$$\sigma_a + \sigma_b = \frac{N_F}{2R\pi t_e} + \frac{H_F L}{R^2 \pi t_e} \leq \sigma_{cr} = \frac{f_y}{\sqrt{1+\lambda^4}}, \tag{4.1}$$

where N_F is the compression force in N, H_F is the bending force in N, L is the length of the shell in mm, R is the radius of the shell in mm, t_e is the reduced shell thickness in mm and f_y is the yield stress in MPa.

Slenderness λ can be calculated making use of the equations

$$\lambda^2 = \frac{f_y}{\sigma_a + \sigma_b}\left(\frac{\sigma_a}{\sigma_{Ea}} + \frac{\sigma_b}{\sigma_{Eb}}\right); \qquad t_e = t + \frac{A_s}{s}; \quad s = \frac{2R\pi}{n_s} \tag{4.2}$$

$$\sigma_{Ea} = C_a\,(1.5 - 50\beta)\,\frac{\pi^2 E}{10.92}\left(\frac{t}{s}\right)^2 \tag{4.3}$$

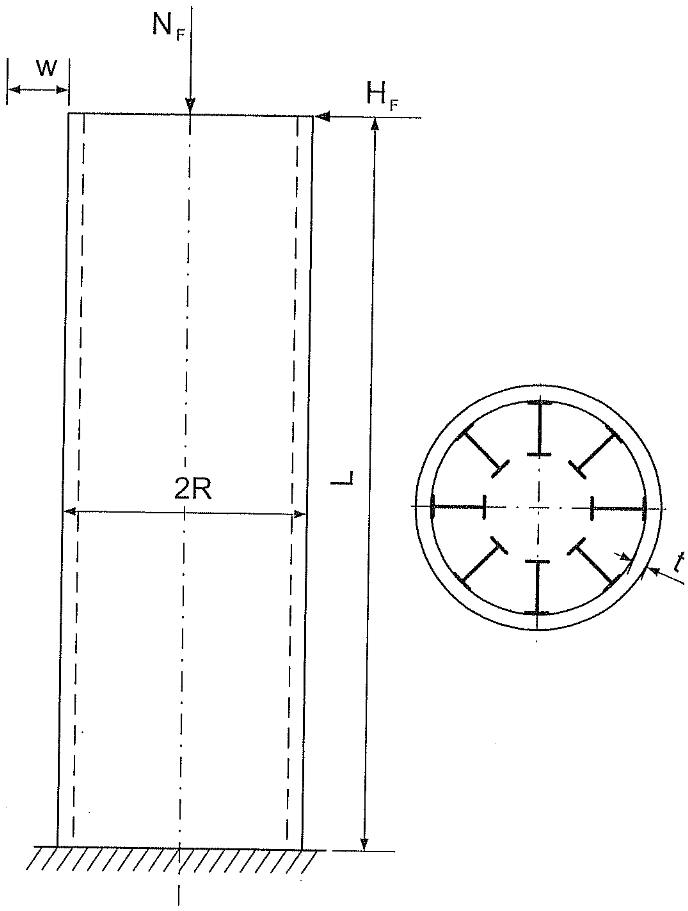

Figure 3. Stringer-stiffened shell

$$C_a = 4\sqrt{1+\left(\frac{\rho_a \xi}{4}\right)^2}; \qquad Z = \frac{s^2}{Rt}0.9539 \tag{4.4}$$

$$\rho_a = 0.5\left(1+\frac{R}{150t}\right)^{-0.5}; \qquad \xi = 0.702Z \tag{4.5}$$

$$\sigma_{Eb} = C_b\,(1.5-50\beta)\,\frac{\pi^2 E}{10.92}\left(\frac{t}{s}\right)^2 \tag{4.6}$$

$$C_b = 4\sqrt{1+\left(\frac{\rho_b \xi}{4}\right)^2}; \qquad \rho_b = 0.5\left(1+\frac{R}{300t}\right)^{-0.5}. \tag{4.7}$$

Note that the residual welding distortion factor is $1.5 - 50\beta = 1$ when $t > 9$ mm.

4.2.2. *Stringer panel buckling.*

$$\sigma_a + \sigma_b \le \sigma_{crp} = \frac{f_y}{\sqrt{1+\lambda_p^4}} , \tag{4.8}$$

$$\lambda_p^2 = \frac{f_y}{\sigma_{Ep}} , \qquad \sigma_{Ep} = C_p \frac{\pi^2 E}{10.92}\left(\frac{t}{L}\right)^2 , \tag{4.9}$$

$$C_p = \psi_p \sqrt{1+\left(\frac{0.5\xi_p}{\psi_p}\right)^2} , \qquad Z_p = 0.9539\frac{L^2}{Rt} , \tag{4.10}$$

$$\xi_p = 0.702 Z_p , \qquad \gamma_s = 10.92\frac{I_{sef}}{st^3} , \tag{4.11}$$

$$\psi_p = \frac{1+\gamma_s}{1+\frac{A_s}{s_e t}} ; \tag{4.12}$$

according to ECCS [26]

$$s_E = 1.9t\sqrt{\frac{E}{f_y}} , \qquad \text{where} \qquad s_e = \begin{cases} s_E & \text{if} \quad s_E < s \\ s & \text{if} \quad s_E > s \end{cases} . \tag{4.13}$$

I_{sef} is the moment of inertia of a cross-section containing the stiffener and a shell part of width s_e. For a stiffener of rolled I-section

$$I_{sef} = I_y + A_S\left(\frac{h+t}{2} - z_G\right)^2 + s_e t z_G^2 \tag{4.14}$$

and

$$z_G = \frac{A_S(h+t)}{2(A_S + s_e t)} . \tag{4.15}$$

Horizontal displacement

$$w_h = \frac{ML^2}{3E\pi R^3 t_e} \le w_{allow} = \frac{L}{\phi} , \tag{4.16}$$

$$M = H_F L/\gamma_M , \qquad \gamma_M = 1.5; \; H_F = 0.1 N_F . \tag{4.17}$$

The limit for displacement is $\phi = 600$.

Numerical data: $N_F = 68000$ kN, $f_y = 355$ MPa, $R = 1850$ mm, $L = 15$ m.
The main parameters of the PSO are as follows:

Probability of craziness (% i.e: 0 - 100) CRAZY = 1.5
Cognitive learning coefficient $0.5 - 2C1 = 2.0$
Social learning coefficient $C2 = 1.4$
Dynamic inertia scale factor. Beta =1 standard PSO alpha BETA = 0.98
Starting value of omega (linear inertia scale factor) OMEGA = 1.0
Minimum allowable fitness FMIN = $-1.0E10$
No-improvement termination criterion (iterations) TITER = 10
Maximum allowed number of function evaluations MAXNF = 10000
Update dynamic inertia criterion (# function evaluations) UPDT = 20
Velocity update factor (1 is normal PSO) VF = 0.985
Minimum allowable omega value MINOMGA = 0.2

Level of craziness of a completely crazy particle, CRTYPE = 1
Moderately craze population CRTYPE = 2
Standard velocity rule VRTYPE = 1
Nico's rule $R2 = (1 - R1)$ original VRTYPE = 2
Random placement BTYPE = 1
Biased placement (minimum on the boundary) BTYPE = 2
Perform max. velocity check VCHK = 1
Don't have velocity check VCHK = 2

4.3. **Curve fitting.** Cost calculations are founded on industrial data. The main parameter is the time of a specific manufacturing element. This time (in min.) multiplied by the specific fabrication cost ($/min) gives the cost (in $). Data given by factories are discrete values and for the optimization we need functions. A curve-fitting program is needed to find the best function for the approximation of the given data. We have used 2D and 3D curve fittings made by the TableCurve 2D [27] and 3D [28] software. We would like to show the efficiency of curve-fitting approximations for the stiffener parameters and the plate forming time calculations.

TableCurve 2D's built-in library includes thousands of equations, a wide array of linear and nonlinear models for any application from simple linear equations to high order Chebyshev polynomials. It contains a 38-digit precision math emulator for properly fitting high order polynomials and rationals. TableCurve 2D speeds up programming by generating actual function code and test routines for all fitted equations in FORTRAN, C, Basic, Pascal and VBA for Excel.

TableCurve 3D's surface fitting contains in addition to standard least squares minimization, three different robust estimations: least absolute deviation, Lorentzian minimization and Pearson VII Limit minimization. Its built-in equation set includes a wide array of linear and nonlinear models for any application:

Linear equations,
Polynomial and rational functions,
Logarithmic and exponential functions,
Non-linear peak functions,
Non-linear transition functions,
Non-linear exponential and power equations,
User-defined functions (up to 15).

It contains 453,697,387 built-in equations. Data input is up to 16,384 points in data table, 16.4 million points can be filtered into table using an averaging digital import filter. Visualization: up to 90,000 vertices can be plotted, resulting in ultra-high 3D surface resolution.

Stiffeners are rolled universal I-beams (UB), their properties are given in the catalogue of Profil Arbed [28].

All parameters of the UB section are calculated in function of the height of the profile.

The cross sectional area is calculated in the following way. Table 1 shows the given data for A and h_s.

Table 1. Data for A and h_s

h (mm)	A (mm^2)	h (mm)	A (mm^2)
152	2032	457	7623
178	2426	533	11740
203	3197	610	14390
254	3204	686	17840
305	3588	838	24680
356	4213	914	28560
406	5864		

TableCurve 2D gives several approximations. The one we have chosen is shown in Figure 4.

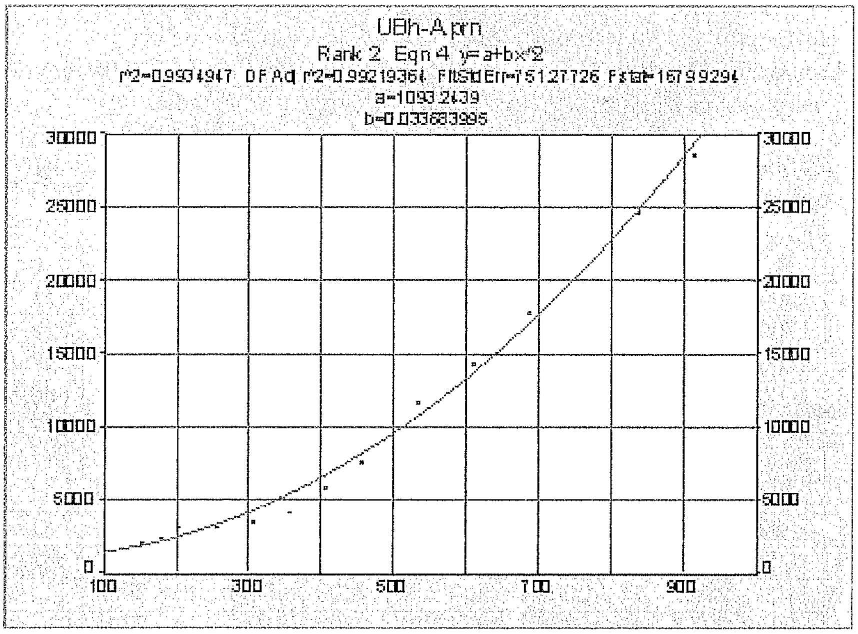

Figure 4. Curve fitting of A in the function of h_s

$$A = a + bh_s^2 \tag{4.18}$$

where the accuracy of the approximation is $r^2 = 0.99349$, $a = 1093.2439$ and $b = 0.0336839$.

The moment of inertia is as follows

$$\ln(I_y) = a + \frac{b}{\ln(h_s)} . \tag{4.19}$$

Here $r^2 = 0.99984798$, $a = 45.0061779$, $b = -156.528802$.

The web thickness of a stiffener can be approximated in the following way

$$t_f = \sqrt{a + bh_s^2}\ , \tag{4.20}$$

where $r^2 = 0.98995121$, $a = 34.55256581$ and $b = 0.000651875$.

The flange width of stiffener can be calculated by

$$b_f = \sqrt{a + bh_s^2}\ , \tag{4.21}$$

where $r^2 = 0.954195679$, $a = 4676.099669$ and $b = 0.111592698$.

Time of plate forming into cylindrical shape (T) can be calculated in function of radius (R) and the plate thickness (t). The forming time is given as a 3D function

$$\ln(T) = a + \frac{b}{t^{0.5}} + cR^{0.5} \tag{4.22}$$

and given in details in equation (4.25).

The industrial data of the Hungarian manufacturiong company (Jászberényi Aprítógépgyár, Crushing Machine Factory, Jászberény) are given in Table 2.

Table 2. Plate forming time T(min) in the function of radius R(mm) and thickness t(mm)

t	R	T	t	R	T	t	R	T	t	R	T
4	1500	145.4	10	1500	348.8	20	1500	485.0	30	1500	611.0
4	1700	151.0	10	1700	352.2	20	1700	490.3	30	1700	619.1
4	2000	161.4	10	2000	366.6	20	2000	507.3	30	2000	643.4
6	1500	211.0	10	2300	379.2	20	2300	525.3	30	2300	666.8
6	1700	220.5	10	2500	386.4	20	2500	536.1	30	2500	687.5
6	2000	229.0	10	3000	401.8	20	3000	556.5	30	3000	713.0
6	2300	236.2	10	3500	420.0	20	3500	579	30	3500	744.0
6	2500	244.3	15	1500	414.2	25	1500	561	40	1500	681.0
8	1500	280.5	15	1700	417.7	25	1700	569.1	40	1700	689.1
8	1700	286.2	15	2000	432.9	25	2000	593.4	40	2000	713.4
8	2000	297.2	15	2300	446.4	25	2300	616.8	40	2300	736.8
8	2300	303.5	15	2500	455.4	25	2500	637.5	40	2500	757.3
8	2500	312.5	15	3000	472.4	25	3000	663	40	3000	783.0
8	3000	325.0	15	3500	490.5	25	3500	694	40	3500	814
8	3500	336.5									

4.4. **Cost function.** The cost function includes the material, fabrication and painting costs [30]. The fabrication costs are calculated by the time of the process.

The fabrication sequence is the following:

Fabrication of 5 shell elements of length 3 m without stiffeners. For one shell element 2 axial butt welds are needed (GMAW-C) (K_{F1}). The cost of forming of a shell element into the cylindrical shape is also included (K_{F0}).

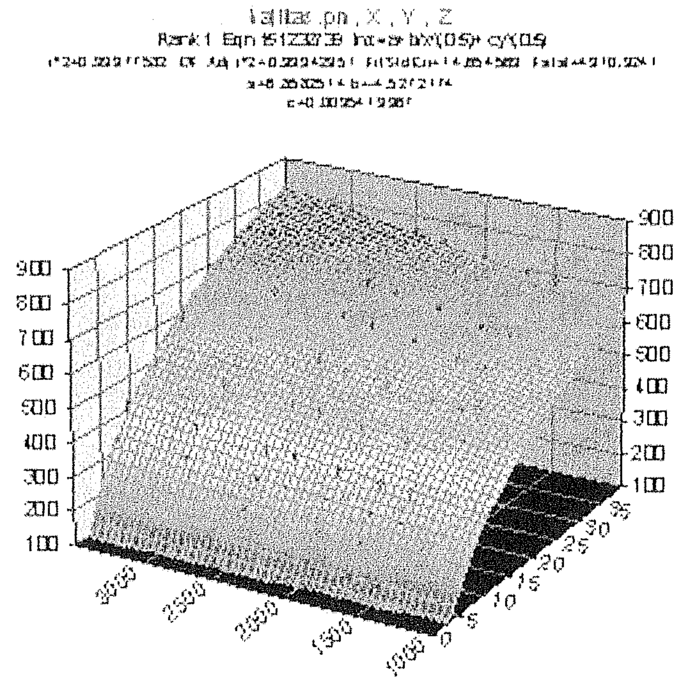

Figure 5. Curve fitting of plate forming time (T) in function of radius ($1500 < R < 3500$) and thickness ($4 < t < 40$)

Welding of the whole unstiffened shell from 5 elements with 4 circumferential butt welds (K_{F2}).

The preparation of longitudinal stiffeners depends on the type of stiffener. When rolled I profiles are used, no preparation cost should be considered.

Welding of n_s stiffeners into the shell with double-sided GMAW-C fillet welds. Number of fillet welds is $2n_s$. (K_{F3}).

4.4.1. *The material cost is for the shell and the stiffeners.*

$$K_M = k_{M1} 5 \rho V_1 + k_{M2} \rho n_s A_s L \tag{4.23}$$

where V_1 is the volume of the shell, t is shell thickness, $V_1 = 3000 \times 2R\pi t$; $\rho = 7.85 \times 10^{-6}$ kgmm^{-3}.

The shell and stiffener material costs can be different, but we use approximately the following values: $k_{M1} = 1.0$ \$/kg, $k_{M2} = 1.0$ \$/kg.

4.4.2. *The fabrication cost can be expressed as.*

$$K_f = k_f \sum_i T_i, \tag{4.24}$$

where K_f [\$] is the fabrication cost, k_f [\$/min] is the corresponding fabrication cost factor, T_i [min] are production times. It is assumed that the value of $k_f$is constant for a given manufacturer. If not, it is possible to apply different fabrication cost factors simultaneously in Equation (4.24).

The corresponding fabrication cost is as follows: $k_F = 1.0$ \$/min.

4.4.3. *Plate forming cost to reach the necessary curvature.*

$$K_{F0} = k_F \Theta e^{\mu}; \quad \mu = 6.8582513 - 4.527217 t^{-0.5} + 0.009541996 \left(2R\right)^{0.5} \tag{4.25}$$

Butt welding cost for one shell element

$$K_{F1} = k_F \left[\Theta \sqrt{\kappa \rho V_1} + 1.3 \times 0.1520 \times 10^{-3} t^{1.9358} \left(2 \times 3000\right) \right] , \tag{4.26}$$

where Θ is a difficulty factor expressing the complexity of the assembly and κ is the number of elements to be assembled

$$\kappa = 2; \; V_1 = 2R\pi t \times 3000; \qquad \Theta = 2 . \tag{4.27}$$

Welding cost of the whole unstiffened shell

$$K_{F2} = k_F \left(\Theta \sqrt{25 \rho V_1} + 1.3 \times 0.1520 \times 10^{-3} t^{1.9358} \times 4 \times 2R\pi \right) . \tag{4.28}$$

Cost of welding of stiffeners into the shell

$$K_{F3} = k_F \left(\Theta \sqrt{\left(n_s + 1\right) \rho V_2} + 1.3 \times 0.3394 \times 10^{-3} a_W^2 2 L n_s \right) . \tag{4.29}$$

The fillet weld size $a_w = 0.5t$, $a_{wmin} = 3$ mm.

$$V_2 = 5V_1 + n_s A_s L . \tag{4.30}$$

The cost of painting is

$$K_P = k_P \left(4R\pi L + (2h + 3b) n_S L\right); \qquad k_P = 14.4 \times 10^{-6} \$/mm^2 . \$/mm^2 . \tag{4.31}$$

The total cost is

$$K = K_M + 5K_{F1} + 5K_{F0} + K_{F2} + K_{F3} + K_P . \tag{4.32}$$

5. Results

The optimization is made considering 3 unknowns (h_s, n_s, t), 3 non-linear constraints (shell buckling, stringer panel buckling, horizontal displacement). For single-objective optimization the *total cost* (1^{st}) is considered. For multiobjective optimization the different parts of the total cost are also considered as independent cost functions: *material cost of the structure* (2^{nd}), *one shell element butt-welding cost and plate forming cost* (3^{rd}), *welding cost of the whole unstiffened shell and the stiffeners into the shell* (4^{th}), *painting cost* (5^{th}). The discrete value step for h_s is 10 mm, for n_s and t is 1 member, or mm.

Table 3 shows the optima determined by PSO changing the number of particles. It shows that one particle can find an optimum. Increasing the number, the reliability of the technique is better, but the computational time also increases. The 'best' solution (in boldface) is the smallest number of particle that finds the minimum.

Table 3. Different optima using different numbers of particles at PSO

Number of particles	Height of stiffener (mm)	Number of stiffener	Thickness of shell (mm)	Cost of the structure ($)
1	730	18	18	118693
2	730	20	16	116993
3	790	18	16	116975
4	760	19	16	117119
5	760	19	16	117119
10	750	19	16	115841
16	**750**	**20**	**15**	**113424**
20	750	20	15	113424
30	750	20	15	113424
60	750	20	15	113424
90	780	20	14	115266
120	780	20	14	115266
150	750	20	15	113424
300	780	20	14	115266
500	780	20	14	115266

Changing the limit for the number of stiffeners to $n_s < 30$ instead of 20, the results can be seen on Table 4. In this case the stiffeners get closer, their welding is more difficult, or can be impossible.

Table 4. Different optima using different number of particles at PSO with larger limit for n_s

Number of particles	Height of stiffeners (mm)	Number of stiffeners	Thickness of shell (mm)	Cost of the structure ($)
3	700	24	13	111385
16	**600**	**34**	**10**	**110502**
90	630	29	12	112547

The Particle Swarm Optimizer has been built into an interactive decision support program system [24], which contains the following single objective optimization methods

Flexible Tolerance (FT) method of Himmelblau [31],
Direct Random Search (DRS),
Hillclimb (HI) method of Rosenbrock [32],
Davidon-Flecher-Powell (DFP) method [33],
Particle Swarm Optimization (PSO) [34, 35].

The efficiencies of these methods are different. All of them use the same objective, constraint subroutines. For a problem like this, which is highly non-linear, several local minima exist. They find different ones. The advantage of Particle Swarm Optimization is that it can find optimum for a nonconvex problem. It has found the minimum cost structure. Table 5 shows the single objective optima.

Table 5. Different optima using different single objective optimization techniques

Method	h_s (mm)	n_s	t (mm)	Cost of the structure ($)
Flexible tolerance	890	15	16	115871
Direct random search	890	13	22	129496
Hillclimb	890	15	16	115871
Davidon Fletcher Powell	890	13	22	129496
Particle Swarm Optimization	**750**	**19**	**16**	**115841**

The interactive decision support program system contains several multiobjective optimization methods. They are the following:
Min-max method,
Global criterion method: type - 1,
Global criterion method: type - 2,
Weighted min-max method,
Weighted global criterion method,
Pure weighting method,
Normalized weighting method.

A description of methods is available in Jármai [24]. Weighting coefficients are similar to all five objectives 0.2 each.

The objective functions are as follows:
Total cost of the structure in \$, K (1^{st}),
Material cost of the structure in \$, K_m (2^{nd}),
Cost of forming and welding of shell elements in \$, $5(K_0 + K_1)$ (3^{rd}),
Welding cost of stiffeners in \$, $K_2 + K_3$ (4^{th}),
Painting cost in \$, K_p (5^{th}).

Table 6 shows the different multiobjective optima. The material cost is dominating, being 50-70 % of the total cost. The other three objectives are around 12-25 %. The height of stiffener is nearly the same for all optima; the number of stiffeners and the shell thickness changes in an opposite way due to the necessary stiffness. The greatest conflict is between the total and the painting costs. The painting cost minimum gives the greatest shell thickness t.

Table 6. Multiobjective optima for the stringer stiffened shell

Method	h_s (mm)	n_s	t (mm)	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
1^{st}	880	17	14	116321.5	73565.8	12479.9	15466.7	14809.0
2^{nd}	890	13	22	129496.9	72627.2	17019.6	27260.2	12589.6
3^{rd}	890	20	11	118337.9	80464.4	10488.7	10719.9	16664.8
4^{th}	890	14	20	126678.2	73160.2	15947.7	24398.3	13171.8
5^{th}	890	11	26	134433.7	71561.2	19104.7	32342.3	11425.3
Min-max	860	17	16	121579.8	73956.7	13693.1	19339.2	14590.5
Global type I	880	19	12	117610.2	77228.8	11180.3	13240.5	15960.5
Global type II	880	16	16	119443.6	73103.0	13693.1	18414.1	14233.3
Weighted Min-max	860	17	16	121579.8	73956.7	13693.1	19339.2	14590.5
Weighted global	880	16	16	119443.6	73103.0	13693.1	18414.1	14233.3
Pure weighting	880	17	14	116321.5	73565.8	12479.9	15466.7	14809.0
Normalized weighting	880	19	12	117610.2	77228.8	11180.3	13240.5	15960.5

6. Conclusions

The particle swarm method is an efficient tool of structural optimization. It can find the global optimum for the problems where the constraints are highly nonlinear, where the feasible region is nonconvex. The algorithm was modified to find discrete values for practical problems. PSO has been built into an interactive program system, which contains other optimization techniques, like Flexible tolerance, Complex, Hillclimb, Davidon-Fletcher-Powell and Direct random search. The efficiency of PSO is shown on a stiffened shell design problem, where there are stringer stiffeners and its loading is compression and bending. The multiobjective optimization gives several optima, considering five objectives, the total cost and the cost elements at the same time. There are conflicts between the objective functions and the different minima of objectives mean different structural sizes and numbers of stiffeners.

Acknowledgement. The research work was supported by the Hungarian Scientific Research Foundation grants OTKA T38058 and T37941. The project was also supported by the Hungarian-South African Intergovernmental S&T Co-operation program. The Hungarian partner is the Ministry of Education, R&D Deputy Undersecretary of State, the South African partner is the Foundation for Research Development. The help of Mr. Derren Wood from the University of Pretoria is gratefully acknowledged in the Particle Swarm Optimization algorithm with the dynamic inertia reduction version.

REFERENCES

1. FARKAS, J., AND JÁRMAI, K.: *Analysis and optimum design of metal structures.* Balkema Publishers, Rotterdam, Brookfield, ISBN 90 5410 669 7, 1997.
2. BOX, M. J.: A new method of constrained optimization and a comparison with other methods. *Computer Journal*, **8** (1965), 42–52.
3. JARMAI, K.: *Topology optimization of tubular structures*, June 1-6. 1998. 5 lectures, Mechanics and Design of Tubular Structures, Advanced Professional School, Coordinated by J. Farkas & K. Jármai, International Centre for Mechanical Science, CISM; Udine, Italy, in K. Jármai and J. Farkas (eds.), Springer Verlag, Chapter 5. 225-284. ISBN 3-211-83145-2, 1998.
4. FARKAS, J., JÁRMAI, K., SNYMAN, J. A., and GONDOS, GY.: Minimum cost design of ring-stiffened welded steel cylindrical shells subject to external pressure, in Lamas, A., and Simoes da Silva, (eds.) Proc. 3rd European Conf. Steel Structures, Coimbra, 2002, Universidade de Coimbra, 2002, pp. 513-522.
5. OSYCZKA, A.: *Multicriterion optimization in engineering.* Ellis Horwood Limited, Chichester, 1984.
6. GOLDBERG, D. E.: *Genetic algorithms in search, optimization & machine learning.* Addison-Wesley Publ. Company, Inc., 1989.
7. STORN, R.: Contrained optimization. *Dr. Dobb's Journal*, May, (1995), 119-123.
8. STORN, R., AND PRICE, K.: *Differential evolution – simple and efficient adaptive scheme for global optimization over continuous spaces.* Technical Report TR-95-012, ICSI., 1995.
9. KENNEDY, J. and EBERHART, R. C.: Particle swarm optimization, Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway, NJ, vol. 4., 1995, pp. 1942-1948.
10. DORIGO, M., MANIEZZO, V., AND COLORNI, A.: The ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, **26** (1), (1996), 1-13.
11. DORIGO, M., DI CARO, G. and GAMBARDELLA, L. M.: Ant algorithms for discrete optimization. *Artificial Life*, **5** (3), (1999), 137-172.
12. SNYMAN, J. A.: A new and dynamic method for unconstrained minimization. *Appl. Math. Modelling*, **6**, (1982), 449-462.
13. SNYMAN, J. A.: An improved version of the original leap-frog method for unconstrained minimization. *Appl. Math. Modelling*, **7**, (1983), 216-218.
14. SNYMAN, J. A.: The LFOPC leap-frog method for constrained optimization. *Comp. Math. Applic.*, **40**, (2000), 1085-1096.

15. MILLONAS, M. M.: *Swarms, phase transitions, and collective intelligence.* in Langton, C.G., (ed) Artificial Life III. Addison Wesley, Reading, MA., 1994.

16. KENNEDY, J.: The particle swarm: social adaptation of knowledge, Proceedings of the International Conference on Evolutionary Computation, IEEE, Piscataway NJ, 1977, pp. 303-308.

17. KENNEDY, J. and SPEARS, W. M.: Matching algorithms to problems: an experimental test of the particle swarm and some genetics algorithms on the multimodal problem generator, Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, AK (USA), 4-9 May 1998 IEEE, New York, 1998, pp. 78-83.

18. KENNEDY, J.: Stereotyping: Improving Particle Swarm Performance with Cluster Analysis, Proc. of the 2000 Congress on Evolutionary Computation, IEEE Service Center, Piscataway, NJ, 2000, pp. 1507-1512.

19. LOVBJERG, M., RASMUSSEN, T. K. and KRINK, T.: Hybrid Particle Swarm Optimiser with Breeding and Subpopulations, Proceedings of the third Genetic and Evolutionary Computation Conference (GECCO-2001), 2001, pp 469-476.

20. FOURIE, P. C. and GROENWOLD, A. A.: Particle swarm in size and shape optimisation, International Workshop on Multidisciplinary Design Optimization, 7-10, Aug. 2000, Pretoria, South Africa, Proceedings, 2000, pp. 97-106.

21. FOURIE, P. C. and GROENWOLD, A. A.: Particle Swarms in Topology Optimization, World Congress on Structural and Multidisciplinary Optimization, Liaoning Electronic Press, Proceedings on CD file Paper No. 113, 2002, 6 p.

22. BOLTON, H. P. J., SCHUTTE, J. F. and GROENWOLD, A. A.: Multiple Parallel Local Searches in Global Optimization, Recent advances in parallel virtual machine and message passing interface, Balatonfüred, Hungary, in Dongarra, J., Kacsuk, P. and Podhorszki, N. (eds.), 2000,pp. 88-95.

23. KOVÁCS, GY., GROENWOLD, A., A., JÁRMAI, K. and FARKAS, J.: Analysis and optimum design of fiber reinforced composite structures. *Structural and Multidisciplinary Optimization*, (2004) (under publication)

24. JÁRMAI, K.: Single- and multicriteria optimization as a tool of decision support system. *Computers in Industry.* Elsevier Applied Science Publishers, **11**(3), (1989) 249-266.

25. DET NORSKE VERITAS (DNV): *Buckling strength analysis.* Classification Notes No.30.1., Hovik, Norway, 1995.

26. ECCS European Recommendations for Steel Construction, *Buckling of steel shells.* No.56. (1988), European Convention for Constructional Steelwork, Brussels

27. *TableCurve 2D*: Users' manual, Systat Software Inc., 2003.

28. *TableCurve 3D*: Users' manual Systat Software Inc., 2003.

29. *Profil Arbed Sales program*, Structural shapes. Arcelor Long Commercial, 2001.

30. JÁRMAI, K. and FARKAS, J.: Cost calculation and optimization of welded steel structures. *Journal of Constructional Steel Research*, **50**, (1999), 115-135.

31. HIMMELBLAU, D. M.: *Applied nonlinear programming.* Mc Graw-Hill Book Co., New York, 1971.

32. ROSENBROCK, H. H.: An automatic method for finding the greatest or least value of a function. *Computer Journal*, **3**, (1960), 175-184.

33. Rao, S. S.: *Optimisation theory and applications.* Wiley Eastern Limited, New Delhi, 1984.

34. Farkas, J. and Jármai, K.: *Economic Design of Metal Structures.* Millpress, Rotterdam, ISBN 90 77017 99 2, 2003.

35. Jármai, K., Farkas, J. and Virág, Z.: Minimum cost design of ring-stiffened cylindrical shells subject to axial compression and external pressure, 5^{th} World Congress on Structural and Multidisciplinary Optimization, Lido di Jesolo, Italy, Schönenfeld & Ziegler, Italian Polytechnic Press, Milano, ISBN 88-88412-18-2, Proceedings, 2003, pp. 63-64.