ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2019.1.121

MENDEL Soft Computing Journal

SPIRAL EXTRUSION DIE DESIGN USING MODIFIED DIFFERENTIAL EVOLUTION ALGORITHM

Michal Pluhacek✉, **Michal Hrdy**, **Adam Viktorin**, **Tomas Kadavy**, **Roman Senkerik**

Faculty of Applied Informatics, University in Zlin, Czech Republic
pluhacek@utb.cz✉, hrdy@utb.cz, aviktorin@utb.cz, kadavy@utb.cz, senkerik@utb.cz

Abstract

In this work, a spiral extrusion die for industrial production of plastic foil has been designed using a modified differential evolution algorithm. The proposed method managed to provide a die design that was compliant with all demands of the foil manufacturer and lowered the production cost. Third-Party software is used to compute the die characteristics from the geometry designed by modified differential evolution.

Keywords: *spiral die, extrusion, industrial application, differential evolution, heuristic computing.*

Received: 09 April 2019
Accepted: 31 May 2019
Published: 24 June 2019

1 Introduction

The extrusion of plastic and polymer materials [1,2,3] has changed our everyday lives in many aspects. It allowed mass production of various plastics products, structure, and constructions. A spiral extrusion die is nowadays the crucial part of many manufacturing processes. However, with the increasing demand on the cost effectivity of the production, energy savings, and other demands, the process of designing a satisfactory spiral die is becoming very complex and computationally expensive.

In this work, the challenge set out by a manufacturing company was to find a valid design of a spiral die for an industrial application of plastic foil production. The typically used (deterministic) methods were unable to obtain a valid solution. Therefore, an innovative approach, based on metaheuristics, was used.

Metaheuristics, notably the evolutionary computational techniques are a modern tool for fast and effective solving of complex optimization problems with high computational demands and complex fitness search space. Among the many popular metaheuristics, the Differential Evolution (DE) algorithm [4,5,6,14] holds a prominent place as one of the most consistently well-performing methods. Recent examples of successful applications of differential evolution for solving the real-world problem were reported in [7], [8] or [9].

In this work, a modified differential evolution algorithm is proposed and utilized alongside a third-party modeling software to design a spiral die for an industrial application.

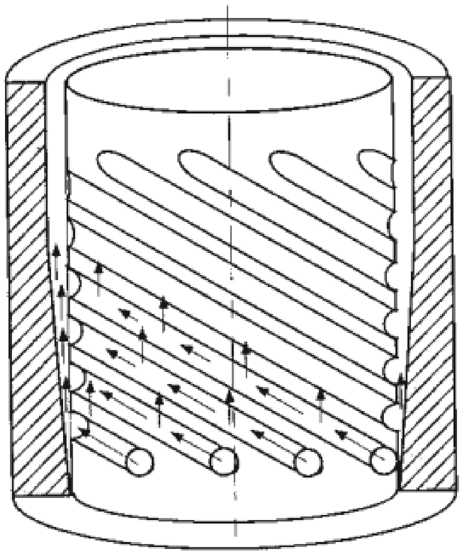
The rest of the paper is structured as follows: The next section introduces the spiral extrusion die and the modeling software used in this work. The tunable parameters of the design are described in section three. The cost function is defined in section four. The proposed modified differential evolution algorithm is described in section five. Following are the results discussion and the conclusion.

2 The Spiral Extrusion Die

In this work, a spiral extrusion die for industrial production of plastic foil (cling film) was designed. During the procedure of foil production, the material is extruded thru the channels on the spiral die. The depth of the channels decreases in the direction to the reference end of the die. The spiral die is mainly popular for its low-pressure demands and good melt distribution.

The spiral die allows production from various materials and in many conditions. A depiction of a spiral extrusion die is given in Fig. 1 alongside with cross-section schematics that depicts the flow of the material in the die. The material flows from the bottom part of the die to the top, following the spiral.

(a)



(b)

Figure 1: The spiral extrusion die (a) and a cross-section view (b)

2.1 The Model

The Virtual Extrusion Laboratory software module “Spiral Die” [13] was used to produce the extrusion die characteristics. A visualization of the pressure change in the spiral die created by the Virtual Extrusion Laboratory software is depicted in Fig. 2. The software provides the extrusion die characteristics based on a provided geometry of the die. Finding the optimal die geometry was the aim of this work.

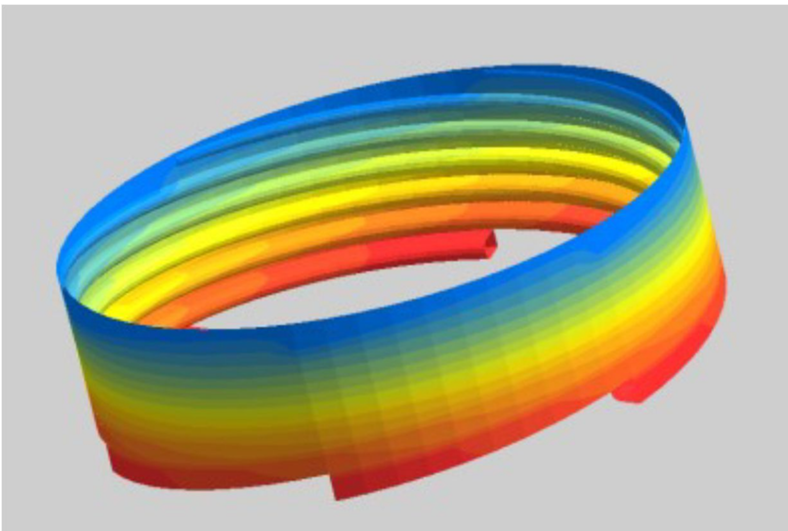


Figure 2: The pressure changes in the spiral die – visualization

3 Spiral Die Geometry Parameters

The spiral die geometry is heavily constrained for both manufacturing and functional reasons. That leaves only four tunable parameters for the optimizer. Despite the low-dimensionality of the problem, the search-space is very complex, and the design optimization is very computationally expensive. The four tunable parameters that were optimized in this work are described further here.

3.1 The Die Input and Output Gap

The output gap G_{te} (see Fig. 3) is the size of the gap between the body and mandrel of the die at the top end (output) of the die. Similarly, the input gap G_{ts} is measured at the opposite, lower opening (input) of the die. The size of the input gap G_{ts} is mainly affecting the speed of material dispersion. Both parameters affect the material flow, velocity, and pressure.
The defined size range for the input and output gap is 0.5–5 mm.

3.2 Input Channel Depth

This parameter d_{cs} represents the depth of the channel dc (Fig. 4) measured at the input opening (reference start) of the spiral die. The defined range for this parameter is 1 – 56 mm.

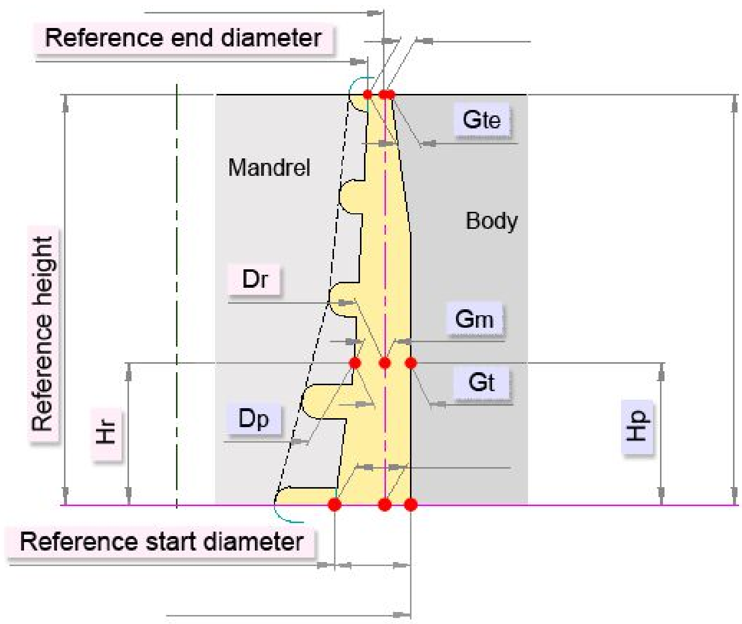


Figure 3: Cross-section view of the flow channel

3.3 Channel Radius

This parameter R_c defines the radius of the spiral channel on its whole length (see Fig. 5). It is a critical parameter for the flow and volume of the melted material. The defined range for the radius value is 1–6 mm.

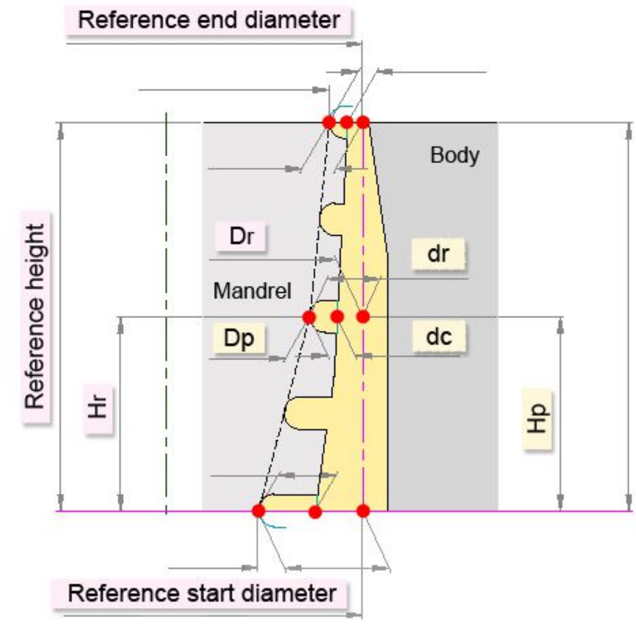


Figure 4: Cross-section view of the flow channel

4 Cost Function Definition

During the complex procedure of designing an extrusion die, several criterions need to be taken into consideration and assigned a priority value (weight). The quality of the die is primarily dependent on the distribution of volume (material) flow Q_v of the melted material on the reference end of the die. Another criterion is a low-pressure loss P_d and an even material distribution, judged by the release speed V_l. A penalization variable is the shear stress S_s. The value of shear stress at several points in the die should not drop under a given threshold value. Following is a more detailed description of this criterions and quality indicators. All described values are obtained from the above mentioned commercial solver, based on the proposed geometry of the spiral die.

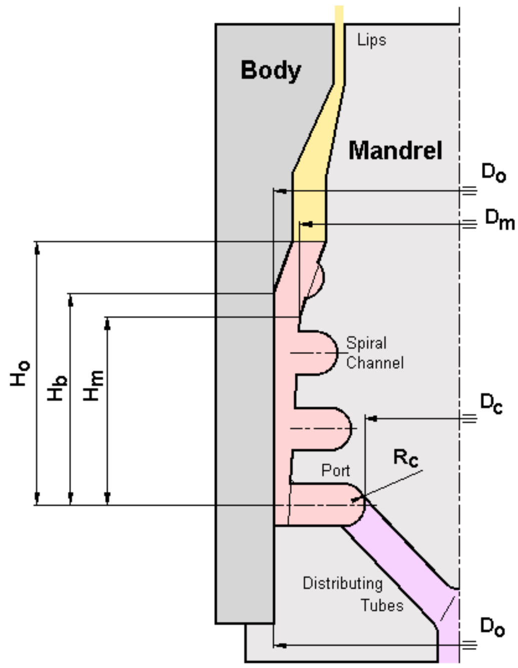


Figure 5: Partial cross-section view of the spiral die

4.1 The Pressure Loss

The model computes the pressure loss (P_a) in MPa. The value is typically in units or dozens of MPa. The goal for the design is to minimize the pressure loss. Minimizing the pressure helps avoid undesirable heating of the die and other parts of the machinery. Further, machinery that allows higher working pressures is generally more costly and has increased energetic demands.

4.2 Output Volume Flow

The value of the output volume flow is computed as a variance value σ2 (1) of material flow in several output points at the output end of the spiral die,

$$\sigma^2(\mathrm{Q_v}) = \sum_{n=1}^{N} \left(Q_{vn} - E\left(Q_v\right)\right)^2 \tag{1}$$

where Q_{vn} is a vector of the measured flow values on the output, and $E\left(Q_v\right)$ is the mean flow value (arithmetical average). The goal for the optimizer is to minimize this value. As mentioned above this parameter has the most significant impact on the quality of the design.

4.3 Channel Release Velocity

The speed of material flow at the release points. This value is given in mm/s. The quality criterion is the sum of squared differences E_s of the calculated release velocity V_{ln} and demanded release velocity V_{l0n} value at each release point (2).

$$E_v(\mathrm{V_l}) = \sum_{n=1}^{N} (V_{ln} - V_{l0n})^2 \tag{2}$$

In ideal case, the release velocity change at different points will follow a linear curve.

4.4 Shear Stress

The shear stress value is measured (in Pa) at several measuring points on the spiral die. The value should not drop under 30kPa. This value is important for the optimal movement of the material in the die. With too low values of shear stress, the material might start settling inside the die and burn. For this reason, a penalization P_{ss} is introduced into the cost function in the following form (3)

$$P_{ss} = \frac{30000 - S_s}{1000} \tag{3}$$

4.5 The Cost Function Completion

All above-described criterions are designed in such fashion that the optimal value is obtained by minimization of the criterions. Therefore, it is possible to complete the cost function as a summation of the criterions P_d, σ^2 and E_s plus the penalization value P_{ss}. However, most of the input parameters are real physical values that are measured in different units, and decimal multipliers of basic units and a balancing mechanism is needed. Weights are introduced (based on the magnitude of the measured quantities) to allow the summation of the criterions, leading to the cost function $CF(G_{te}, G_{ts}, d_{cs}, R_c)$ given by (4),

$$CF(G_{te}, G_{ts}, d_{cs}, R_c) = w_1 P_d + w_2 \sigma^2\left(Q_v\right) + w_3 E_s(V_l) + P_{ss} \tag{4}$$

where: $w_1 = 10^{-7}, w_2 = 10^7$ and $w_3 = 10^{-2}$.

5 Differential Evolution

The canonical 1995 DE [4] is based on the idea of evolution from a randomly generated set of solutions of the optimization task called population $\boldsymbol{P}$, which has a preset size of *NP*. Each individual (solution) in the population consists of a vector $\boldsymbol{x}$ of length D (each vector component corresponds to one attribute of the optimized task) and objective function value $f(\boldsymbol{x})$, which mirrors the quality of the solution. The number of optimized attributes D is often referred to as the dimensionality of the problem, and such generated population $\boldsymbol{P}$, represents the first generation of solutions.

The individuals in the population are combined in an evolutionary manner in order to create improved offspring for the next generation. This process is repeated until the stopping criterion is met (either the maximum number of generations, or the maximum number of objective function evaluations, or the population diversity lower limit, or overall computational time), creating a chain of subsequent generations, where each following generation consists of better solutions than those in previous generations – a phenomenon called elitism. The combination of individuals in the population consists of three main steps: Mutation, crossover, and selection.

In the mutation, attribute vectors of selected individuals are combined in simple vector operations to produce a mutated vector $\boldsymbol{v}$. This operation uses a control parameter – scaling factor F. In the crossover step, a trial vector $\boldsymbol{u}$ is created by selection of attributes either from mutated vector $\boldsymbol{v}$ or the original vector x based on the crossover probability given by a control parameter – crossover rate *CR*. Finally, in the selection, the quality $f(\boldsymbol{u})$ of a trial vector is evaluated by an objective function and compared to the quality $f(\boldsymbol{x})$ of the original vector and the better one is placed into the next generation.

From the basic description of the DE algorithm, it can be seen, that there are three control parameters, which have to be set by the user – population size *NP*, scaling factor F and crossover rate *CR*. It was shown in [10] and [11], that the setting of these parameters is crucial for the performance of DE. Fine-tuning of the control parameter values is a time-consuming task, and therefore, many state-of-the-art DE variants use self-adaptation in order to avoid this cumbersome task [5,6,12]. In this work, a simple adaptation of F and *CR* parameters is implemented in order to avoid the problem of the correct setup of these parameters, that would be computationally very expensive. The algorithm proposed with such change was titled Auto-Adaptive Differential Evolution (AADE).

5.1 Changes in AADE

The AADE algorithm implements a simple adaptive behavior for scaling factor F and crossover rate values *CR* since those influence the optimization process of the DE. For each mutation and crossover step, F_i and CR_i values are generated dynamically for each individual from a normal distribution with the mean value of F_m or CR_m and standard deviation of 1 (5).

$$CR_i = N[CR_m, 1] \text{ and } F_i = N[F_m, 1], \tag{5}$$

where F_m and CR_m are mean values of successful scaling factor and crossover rate parameters, respectively. During the selection step, values of F and *CR* that helped to produce a better offspring are stored in a corresponding memory ($\boldsymbol{S}_F$ and $\boldsymbol{S}_{CR}$). After each generation, the mean values of the contents of these memories are computed and stored into F_m and CR_m (6). For the first generation, F_m is set to 0.5 and CR_m to 0.8. It is also important to note, that F_i and CR_i values are bounded between 0 and 1 and if they are generated outside of that range, their value is set to the closest boundary value.

$$F_m = \text{mean}(\boldsymbol{S}_F), \boldsymbol{S}_F \neq \emptyset \text{ and } CR_m = \text{mean}(\boldsymbol{S}_{CR}), \boldsymbol{S}_{CR} \neq \emptyset. \tag{6}$$

Since the preliminary testing showed a problem with premature stagnation of the population, the stagnation restart of F_m and CR_m parameters were implemented. This restart resets F_m and CR_m values to the initial 0.5 and 0.8 respectively after 30 generations without an improvement (30 generations of population stagnation). The original mutation and crossover steps are updated only in a slight change of dynamical F_i and CR_i values. These two steps are depicted below.

5.2 Mutation

In the mutation step, three mutually different individuals $\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$ from a population are randomly selected and combined following the mutation strategy. The original mutation strategy of canonical DE is “rand/1” and is depicted in (7).

$$\boldsymbol{v}_i = \boldsymbol{x}_{r1} + F_i(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{7}$$

Where $r1 \neq r2 \neq r3 \neq i$, F_i is the scaling factor value, and $\boldsymbol{v}_i$ is the resulting mutated vector.

5.3 Crossover

In the crossover step, mutated vector vi is combined with the original vector $\boldsymbol{x}_i$ to produce the trial vector $\boldsymbol{u}_i$. The binomial crossover (8) is used in canonical DE.

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } U[0,1] \leq CR_i \text{ or } j = j_{rand} \\ x_{j,i} & \text{otherwise} \end{cases} \tag{8}$$

Where CR_i is the used crossover rate value, and j_{rand} is an index of an attribute that has to be from the mutated vector $\boldsymbol{v}_i$ (this ensures generation of a vector with at least one new component).

6 The Spiral Die Design Results

This section presents the results of AADE on an industrial spiral die design. The algorithm setup is given in Table 1.

Table 1: AADE algorithm setup

dim	CR_{init}	F_{init}	Population	CF evaluations
4	0.8	0.5	50	20 000

Algorithm pseudo-code 1: AADE

```
Set NP, CRm = 0.8, Fm = 0.5, and stopping criterion;
G = 0, xbest = {}, stag_counter = 0;
Randomly initialize population P = (x1,G,…,xNP,G);
Pnew = {}, xbest = best from population P;
while stopping criterion not met
for i = 1 to NP do
stag_counter++;
xi,G = P[i];
Generate Fi and CRi by (1);
vi,G by mutation (3);
ui,G by crossover (4);
if f(ui,G) ≤ f(xi,G) then
xi,G+1 = ui,G;
Fi → SF, CRi → SCR;
stag_counter = 0;
else
xi,G+1 = xi,G;
end if
xi,G+1 → Pnew;
end for
Compute new Fm and CRm by (2);
if stag_counter = 30 then
Reinitialize Fm and CRm;
end if
P = Pnew, Pnew = {}, xbest = best from population P;
end while
return xbest as the best found solution
```

The proposed method was tested on 30 independent runs. The results overview is presented in Table 2. Further, the convergence of the best run is depicted in Fig. 6. Finally, the best-obtained solution was tested for validity using above-presented criteria.

Table 2: AADE algorithm results (30 runs)

Best	Worst	Median	Mean	Std. dev.
94.6769	175.506	94.6775	110.843	36.1476

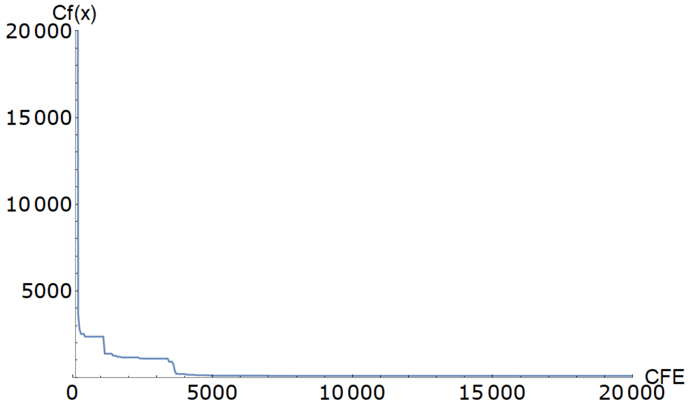


Figure 6: The solution convergence of the proposed method (best run)

Graphical visualization of the resulted characteristic is presented alongside a commentary. Based on the provided evidence (Fig. 7 – 9), it is possible to acknowledge the quality and validity of the design.

The sheer pressure course inside the spiral channel is given in Fig. 7. It is critical that the pressure does not drop back below the 30 kPa threshold. Therefore, the design is valid.

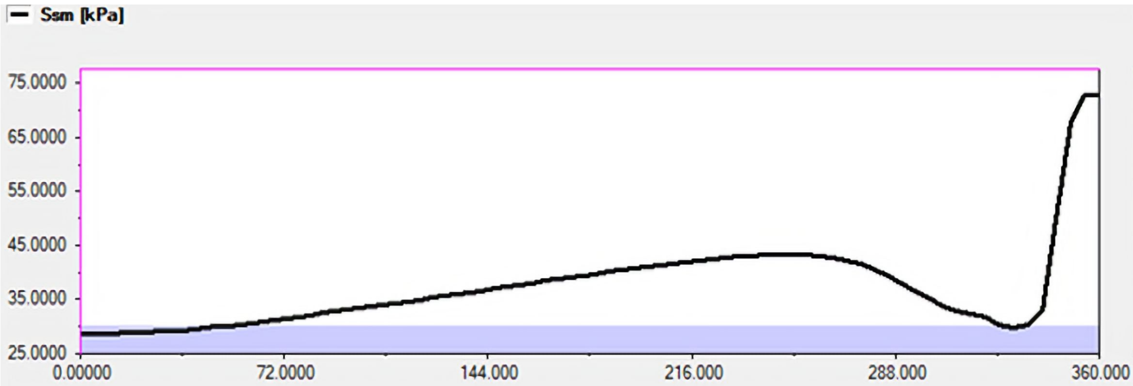


Figure 7: The course of shear pressure inside the spiral channel (x – time (s))

Channel release velocity is supposed to follow a linear curve in the ideal (unrealistic) case. In Fig. 8, the channel release velocity of the designed die is presented. The course is linear enough for valid die design.

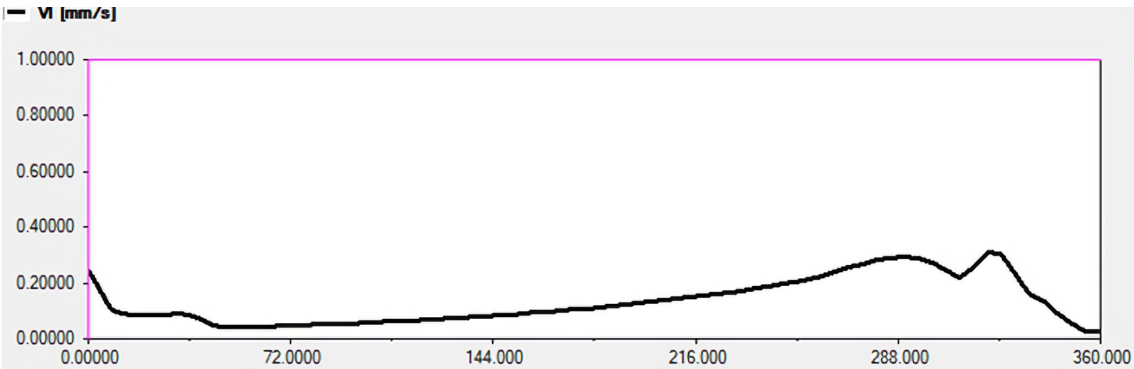


Figure 8: The course of shear pressure inside the spiral channel (x – time (s))

The most significant quality indicator is the uniformity of material volume flow on the output of the die (Fig 9). The achieved variance value $\sigma^2 = 0.675\%$ is excellent and more than satisfactory for this particular application.

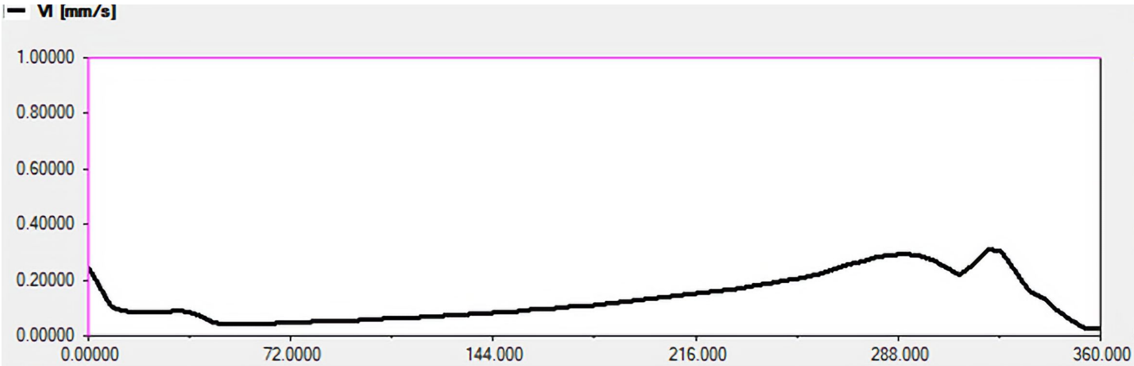


Figure 9: The course of material volume flow with a computed variance of the mean (x – time (s))

Finally, the pressure loss in the designed spiral die was approx. 5 MPa. All parameters are therefore satisfactory, and the design is valid.

7 Conclusion

In this work, an extrusion spiral die for the production of plastic foil was designed using a modified differential evolution algorithm called AADE. The initial experiments with standard deterministic methods (carried out by the manufacturing company) were unsuccessful in producing a valid design. Therefore, a metaheuristic, namely the differential evolution algorithm, was chosen to solve the optimization problem.

However, tuning of control parameters of the standard differential evolution is very computationally expensive and unsuitable for this application, given the computational limitations and time requirements; therefore, a simple adaptive differential evolution was proposed. The proposed method managed to produce high quality and valid design of a spiral die for this industrial application that has been successfully used. The newly produced design works with low pressures and improves the economic aspects of the foil manufacturing process.

The method presented in this paper seems to be very effective for this type of complex, soft constrained (penalized) and computationally expensive real-world optimization problem and supports the usefulness of evolutionary based metaheuristic methods for industrial applications. In future research, the possibilities of using different metaheuristics for this particular problem will be investigated, and a performance comparison will be compiled.

Acknowledgement: This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project no. LO1303 (MSMT-7778/2014), further by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2019/002. This work is also based upon support by COST (European Cooperation in Science & Technology) under Action CA15140, Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO), and Action IC1406, High-Performance Modelling, and Simulation for Big Data Applications (cHiPSet). The work was further supported by resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin (ailab.fai.utb.cz).

References

[1] Crawford, R. J. 1998. *Plastics engineering.* 3rd ed. Butterworth-Heinemann, Boston, USA. ISBN 0-7506-3764-1.

[2] Michaeli, W. and Hopmann C. 2016. *Extrusion dies for plastics and rubber.* 4th Edition. Hanser Publishers, Cincinnati, USA. ISBN 978-1-56990-623-1.

[3] Rauwendall, C. 2014. *Polymer extrusion.* 5th edition. Cincinnati: Hanser Publishers, Cincinnati, USA. ISBN 978-1-56990-516-6.

[4] Storn, R. and Price, K. 1997. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization* 11, 4, pp. 341–359. DOI: 10.1023/A:1008202821328

[5] Das, S. and Suganthan, P. N. 2011. Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* 15, 1, pp. 4–31. DOI: 10.1109/TEVC.2010.2059031

[6] Das, S., Mullick, S. S., and Suganthan, P. N. 2016. Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation*, 27, April 2016, pp. 1-30. DOI: 10.1016/j.swevo.2016.01.004

[7] Yildiy, A. 2019. Predicting the energy production of a rooftop PV plant by using differential evolution algorithm. *Turkish Journal of Engineering (TUJE)*, 3, 3, pp. 106–109. DOI: 10.31127/tuje.466953

[8] Badillo-Olvera, A., et al. 2019. Burst detection and localization in water pipelines based on an extended differential evolution algorithm. *Journal of Hydroinformatics.* DOI: 10.2166/hydro.2019.123

[9] Wang, L., et al. 2018. Effective electricity energy consumption forecasting using echo state network improved by differential evolution algorithm. *Energy*, 153: 15 June 2018, pp. 801–815. DOI: 10.1016/j.energy.2018.04.078

[10] Gamperle, R., Muller, S. D., and Koumoutsakos, P. 2002. A parameter study for differential evolution. In *WSEAS Int. Conf. on Advances in intelligent systems, fuzzy systems, evolutionary computation.* WSEAS Press, Stevens Point, WI, USA, Vol. 10, No. 10, pp. 293–298.

[11] Liu, J. and Lampinen, J. 2002. On setting the control parameter of the differential evolution method. In *Proceedings of the 8th international conference on soft computing (MENDEL 2002).* Brno, Czech Republic, pp. 11–18.

[12] Neri, F. and Tirronen, V. 2010. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33, 1–2, pp. 61–106. DOI: 10.1007/s10462-009-9137-2

[13] Virtual Extrusion Laboratory. 2016. Software, on-line: http://www.compuplast.com/product/vel-modules, Comuplast.

[14] Matousek, R., Popela, P. and Kudela, J. 2017. Heuristic Approaches to Stochastic Quadratic Assignment Problem: VaR and CVar Cases. *MENDEL*. 23, 1 (Jun. 2017), 73-78.
DOI: https://doi.org/10.13164/mendel.2017.1.073.