Separation Science and Technology
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713708471

Design Analysis of Polymer Filtration using a Multi-Objective Genetic Algorithm

K. R. Fowler[a]; E. W. Jenkins[b]; C. L. Cox[b]; B. McClune[a]; B. Seyfzadeh[c]
[a] Department of Mathematics, Clarkson University, Potsdam, NY, USA [b] Department of Mathematical Sciences, Clemson University, Clemson, SC, USA [c] Invista Corporation, Dalton, GA, USA

To cite this Article Fowler, K. R. , Jenkins, E. W. , Cox, C. L. , McClune, B. and Seyfzadeh, B.(2008) 'Design Analysis of Polymer Filtration using a Multi-Objective Genetic Algorithm', Separation Science and Technology, 43: 4, 710 — 726
To link to this Article: DOI: 10.1080/01496390701870713
URL: http://dx.doi.org/10.1080/01496390701870713

Separation Science and Technology, 43: 710–726, 2008

ISSN 0149-6395 print/1520-5754 online
DOI: 10.1080/01496390701870713

Design Analysis of Polymer Filtration using a Multi-Objective Genetic Algorithm

K. R. Fowler,[1] E. W. Jenkins,[2] C. L. Cox,[2] B. McClune,[1] and B. Seyfzadeh[3]

[1]Department of Mathematics, Clarkson University, Potsdam, NY, USA
[2]Department of Mathematical Sciences, Clemson University, Clemson, SC, USA
[3]Invista Corporation, Dalton, GA, USA

Abstract: Filtration of particle debris is an important component of the polymer fiber melt-spinning process. The filter lifespan is determined by the pressure drop across the filter, which increases as debris accumulates inside the filtration medium. The cost of filter replacement is high, as is the cost of a loss of the finished fiber product due to debris inclusion in the spun fiber. We use a multiobjective genetic algorithm to examine the trade-off curve that evolves from these competing goals. A "blackbox" simulator models the debris deposition, and we choose filter porosity and pore diameter as the design variables. We provide numerical results and analysis for two sets of competing objectives.

Keywords: Polymer processing, debris filtration, multi-objective, genetic algorithm

INTRODUCTION

Modeling of an entire fiber-spinning process is of great interest to researchers in the Center for Advanced Engineering Fibers and Films (CAEFF). Initial funding for CAEFF has been provided through the Engineering Research Centers Program of the National Science Foundation, and CAEFF is transitioning to a self-supported fiber and film research center. More information can be found at the CAEFF website (1).

Received 21 August 2007, Accepted 2 November 2007
Address correspondence to E. W. Jenkins, Department of Mathematical Sciences, Clemson University, Clemson, SC, USA. E-mail: lea@clemson.edu

One of the primary missions of CAEFF is to aid industrial partners in the development and manufacture of new polymer products. Towards this end, members of CAEFF have developed simulation tools for the various stages of fiber production, from the initial polymer melt to the finished product (see Fig. 1). One of the simulation tools is a three-dimensional model of an extrusion filter, which separates debris particles from the polymer before the polymer is spun into a fiber. Fibers are formed by pushing the molten polymer through a spinneret, which is a multi-pored extrusion device. The polymer solidifies into a fiber once it exits the spinneret and is exposed to quench air. The properties of the fiber are severely compromised if debris particles remain in the polymer during this part of the process, and this may result in the loss of an entire production unit. In addition, replacing the filter is costly; it often requires that the entire spinline be taken out of service. While in-line filter replacement processes are being introduced, Hookway (2) states that the trend in industry is to use "staged filtration," i.e., combining in-line filters with a filter at the spinneret.

A rough estimate of the cost associated with downtime due to filter replacement is as follows. From (3), the profit on one metric ton of polyester has averaged, over the last five years, \$100 (a modest estimate). Taking a typical flow rate of PET to be 2.5 g/min/hole, and figuring 3000 holes in a single spinneret, gives a loss in profit per hour of \$45 when a filter above a spinneret gets clogged. A large fiber-producing plant may have several spinning lines with each containing 100 spinnerets. Furthermore, PET is a high volume, low-cost commodity polymer. The financial impact of short filter life on low volume, high profit (e.g. functional) fiber production would be significantly greater.

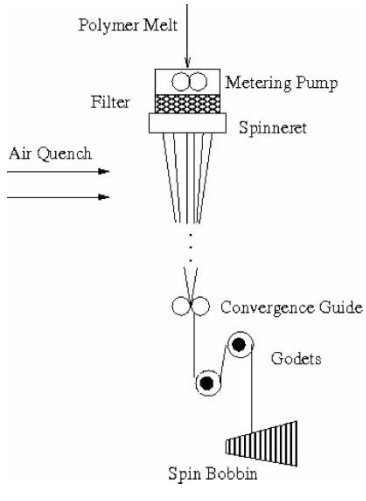


Figure 1. Fiber melt-spinning process.

The filter is often composed of a sintered metal, compressed with sufficient force to produce a cake material, or layers of wired mesh, with mesh spacings small enough to trap particles a few microns in diameter. Important design parameters thus include the number of distinct filter layers and the characteristics of each layer. These characteristics include the length of the layer, the porosity, and the average diameter of the pore spaces. The overall lifespan of the filter depends not only on the filter design specifications but on the debris profile for the polymer being processed.

Previous CAEFF work on filtration (4, 5) focused on the development of model equations, numerical simulation, and comparisons of the simulations with experimental data. Pressure drop values were calculated for several distinct fluid and filter types, and the results suggested that the simulator accurately modeled the deposition process. In this work, we seek to understand the effects of design parameters on the performance of the filter. We measure the performance of a filter by the lifetime of the filter (determined by the maximum allowable pressure drop across the filter) and the effectiveness of the filter in debris removal. Maximizing the filter lifetime, while simultaneously minimizing the mass of debris that escapes the filter, defines an optimization problem with competing objectives. Clearly, a filter will last longer if it traps nothing.

For this preliminary study, we use the initial porosity of the filter and the average pore size diameter as the design variables. We analyze the performance using a multiobjective genetic algorithm to generate a set of Pareto optimal solutions (6, 7). A design engineer can use these Pareto solutions to find a preferred design as opposed to a completely optimal design, which may not even exist as improvement in one of our objective functions necessarily leads to degradation in the other. We consider a nondominated, sorting genetic algorithm (NSGA-II) (8) to generate the tradeoff curve.

The optimization problem is a simulation-based or "black-box" problem, since an evaluation of the objectives requires output from the extrusion filter simulator. This is a challenging class of problems because gradient information, which is traditionally used to locate critical points, is not available. Further adding to the complexity of this work is the fact that one simulation can take anywhere from a few seconds to several hours, depending on the filter parameters.

Traditional methods often combine competing objectives into a single functional. Weights can be incorporated into the single objective to balance the relative goals of the design. However, it is not always the case that the single objective accurately achieves the same purpose as the multiobjective approach (9–11). Moreover, the single objective function leads to one optimal point rather than a family of points that characterize the design space. This work is a first attempt to use optimization algorithms in conjunction with a three-dimensional polymer filtration simulation tool. This initial study is meant to help us understand the interplay between our chosen design parameters and the capabilities of the filter.

The paper is organized as follows. In Section 2, we describe the simulation tool. In Section 3, we discuss the multiobjective functionals and describe the genetic algorithm (GA). In Section 4, we provide numerical results along with profiles of the design space. We summarize our findings and provide direction for future work in Section 5.

FILTRATION MODEL

A schematic of the fiber melt-spinning process is given in Fig. 1. The metering pump is used to maintain a constant flow rate; large pressure drops due to debris loading can adversely affect the pump. In addition, debris particles must be removed before the polymer reaches the spinneret. After that point, any remaining debris particles will be spun into the fiber, leading to degradation in fiber properties.

The filtration code (i.e., the "simulator") used for this work is based on results from CAEFF researchers. In (12), Edie and Gooding studied one-dimensional filtration equations to predict pressure drop across a filter medium. This work was extended to three dimensions by Seyfzadeh, Zumbrunnen, and Ross (5), which led to the development of the simulator. A restricted version of the filtration simulator is available for academic use only; a full-featured licensed version is also available. More information is available from the CAEFF website.

The material entering the filter consists of molten polymer, unmelted polymer gel particles (formed when the polymer does not completely melt in the heating/mixing stage), and other debris such as metal particles. It is assumed that the density of the mixture is the same as the polymer density, i.e., the density of the debris is negligible in comparison to the polymer density. It is also assumed that the thickness of the filter is small so that effects due to gravity can be ignored (5, 12).

The flow through the filter is governed by the continuity equation (mass conservation) and Darcy's law, modified to account for the non-Newtonian behavior of the polymer melt. The debris entering the filter is specified in terms of a mass fraction relative to the mass of the mixture.

As the flow enters a computational cell, the mass of captured debris of each distinct material type is calculated. This mass is determined by comparing the debris particle sizes against the average pore diameter of the cell. All of the debris particles larger than the pore diameter are partitioned into an array. A retention function is then used to determine how many of these particles will be captured by the filter. Any particles not captured within that computational cell are transported into adjacent cells.

The porosity is updated by subtracting the volume of deposited debris over the time step from the porosity value at the beginning of the time step. The volume is obtained by multiplying the mass deposited by the density of that particular debris type. After the porosity is updated, a new average pore diameter for the computational cell is calculated.

The new values for porosity and pore diameter are used to calculate the permeability and pressure drop. The permeability parameter k is currently modeled using a Blake-Kozeny relationship that depends on the average diameter of the filter pore size, d_p, and the current filter porosity, η. The filter porosity is a dimensionless quantity that measures the volume of the void space to the total volume of the medium. The relationship for k, derived in (13) and used in (5, 12), is

$$k = \frac{d_p^2 \eta^3}{150(1-\eta)^2}.$$

This permeability relationship is valid for fluids with Reynolds number less than 1. The Reynolds number associated with our simulations, calculated using the formula in (14), is around 10^{-5}.

The simulator is written in Matlab. For one particle size/porosity pair, an execution takes approximately 160 minutes on a Dell Latitude D620 laptop with an Intel core duo 1.83 Ghz processor. Additional details about the filtration simulator can be found in (5).

Debris loading is characterized statistically, as in (2). Truncated normal statistical distributions provide flexibility in fitting experimental measurements. Inflow distributions are specified using a standard normal probability density function,

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

a left truncated normal distribution of the form

$$f_{LTN}(x) = \begin{cases} 0 & -\infty < x \le x_L \\ \dfrac{f(x)}{\int_{x_L}^{\infty} f(x)dx} & x_L < x < \infty \end{cases}$$

and a right truncated normal distribution

$$f_{RTN}(x) = \begin{cases} \dfrac{f(x)}{\int_{-\infty}^{x_R} f(x)dx} & -\infty < x \le x_R \\ 0 & x_R < x < \infty \end{cases}$$

respectively.

Filters are often composed of multiple layers. Each layer is characterized by a porosity, an average pore diameter, and a retention function. The retention function models the efficiency of the filter by providing the probability that a particle of a given size is captured by the filter. The graph of a typical retention function is displayed in Fig. 2. The retention functions are found by fitting empirical data to a growth function of Boltzmann-Sigmoidal type with four fitting parameters (5).

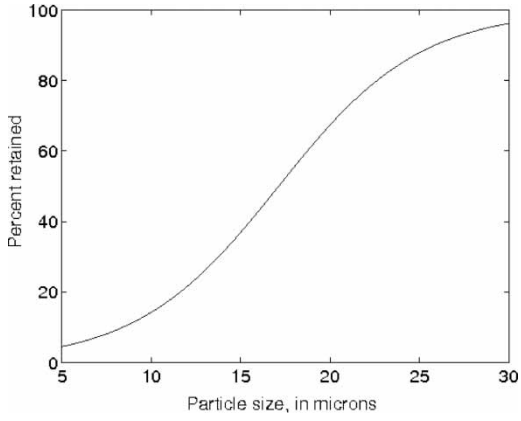


Figure 2. Typical filter retention function.

As the filter clogs, the volume of the void space decreases, leading to a decrease in the permeability. The pressure drop must then increase in order to maintain a constant mass flow rate throughout the filter. The large values of pressure required at the inlet may lead to damage of the metering pump (see Fig. 1), so the filter must be replaced once the pressure drop reaches a threshold value. The threshold value is set at 35e6 Pa in the simulator, which is comparable to existing industry thresholds (5).

OPTIMIZATION STRATEGY

Our goal is to maximize the lifetime of a filter while simultaneously minimizing the amount of debris that escapes. These are competing objectives relative to the design parameters porosity (η) and pore diameter (d_p). For instance, if $\eta = 1$, then the filter is completely open and no debris is trapped inside the filter; thus the filter lasts indefinitely. If $\eta = 0$, then the no debris escapes from the filter but the filter lifetime is zero.

We pose the optimization problem as

$$\begin{aligned} &\max J_1(x, w) \\ &\min J_2(x, w) \\ &\text{subject to } x \in \Omega, \end{aligned} \tag{1}$$

where J_1 represents the lifetime of the filter, J_2 is a measure of the mass of debris escaping the filter, $x = \{\eta, d_p\}$, w is output from the simulator, and Ω defines the bound constraints component-wise on the decision variables x_i. Specifically, $\Omega = \{u \in IR^2 : L_i \le x_i \le U_i\}$, $i = 1, 2$, where the upper and lower bounds L_i and U_i satisfy $0 < L_i < U_i < \infty$.

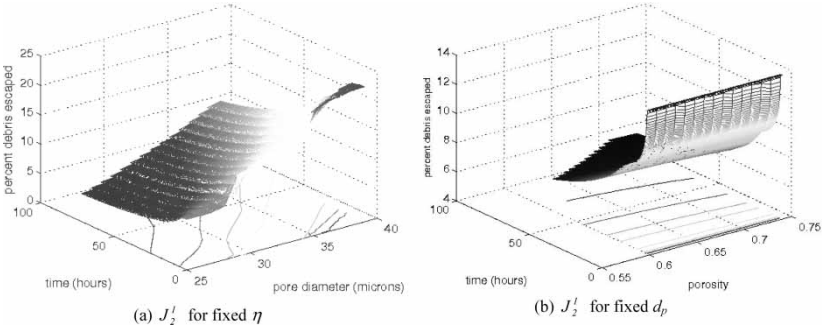
Approaching Eq. (1) using the NSGA-II algorithm, described below, leads to a set of Pareto points. Pareto solutions are those for which improvement in one objective can only occur with the worsening of another objective. We consider two distinct formulations of Eq. (1), which differ in the definition of J_2. We describe the formulations below.

Problem Formulations

Both formulations define J_1 as the time it takes for the pressure drop to reach 35e6 Pa. At this time, the filter is considered effectively clogged and must be replaced, as continuing to operate the spinline could result in catastrophic damage to the pumping mechanism. The goal of the second objective, J_2, is to account for the debris that escapes the filter and thereby degrades the quality of the fiber. We consider two representations for this goal, labeled J_2^I and J_2^{II}. In J_2^I, we minimize the percent of total debris that escapes the filter. This percentage is the ratio of the mass of escaped debris to the mass of incoming debris. We use this as a performance objective as other authors have considered this measure in their work. In fact, Hookway (2) estimates the number of particles that escape per 1000 lbs of polymer. In J_2^{II}, we minimize the total mass of debris that escapes the filter.

Optimization Landscapes

To gain insight into the design space, we generated representative optimization landscapes for each of J_2^I and J_2^{II}. These landscapes helped to identify the behavior of the objective functions. In Fig. 3(a), we fixed $\eta = 0.65$ and ran the simulator for values of $d_p \in [25, 39]$ μm, using increments of 1 μm, to collect the percentage of debris escaping at each time step. More debris escapes the filter at the beginning of the simulation and the longer the filter operates, the smaller the percentage of debris that escapes. For



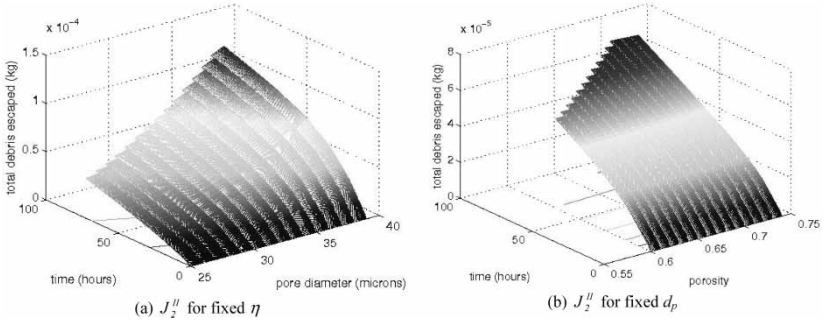
(a) J_2^I for fixed η (b) J_2^I for fixed d_p

Figure 3. Values of J_2^I functional for fixed d_p, η.

example, when $d_p = 25$ μm, then 8.6% of the debris in the slurry at the first time step escapes the filter and by the end of the simulation, 4.8% of the total debris has escaped. When considering $d_p = 39$ μm, 23% of the debris in the slurry at the first time step escapes the filter and by the end of the filter lifetime, 9.2% of the total debris has escaped. The difference in the lifetimes of the filters is 72.1 hours and 100.8 hours, respectively. The snapshot of this landscape shows that the filter lifetime is extended at the expense of possibly severely degraded fibers. The landscape also shows three distinct plateaus at the beginning of the time history. This corresponds to the three distinct debris types and their respective particle sizes. Note that at the end of the simulation, the percent of escaped debris varies almost linearly as a function of pore diameter.

Alternatively, in Fig. 3(b), we fix $d_p = 29.4$ μm and sample points for $\eta \in [0.6, 0.74]$ using increments of 0.01. Here, when $\eta = 0.6$, the filter lasted 66.4 hours and at the end of the simulation, 6.2% of the debris had escaped. Notice, however, that 13.8% of the debris escapes during the initial time step. At the upper value for porosity, the same percentage of debris escapes during the initial time step with 5% of the total debris escaping by the end of the simulation. Although this is a lower percentage, the filter lasts 111.2 hours before the pressure drop criterion is met. Thus the filter lasts nearly 70% longer and as a result, more total debris is actually being pushed through.

Similarly, in Figs. 4(a) and (b), we generated landscapes for J_2^{II} and collected the total mass of escaped debris at each time step for fixed values of porosity and pore diameter, respectively. In Fig. 4(a), for a fixed porosity of $\eta = 0.65$, 4.1622e − 5 kg of debris escapes when $d_p = 25$ and 1.1204e − 4 kg of debris escapes for the upper bound, $d_p = 39$. Moreover, the amount varies nearly linear at the final time as a function of d_p. We see similar behavior in Fig. 4(b). In this case the amount of debris escaping the filter is 4.9638e − 5 kg for the lower bound on porosity and 6.9613e − 5 kg for the upper bound. The total mass of escaped debris at the final time clearly varies less with changes in porosity than with changes in pore diameter.



(a) J_2^{II} for fixed η (b) J_2^{II} for fixed d_p

Figure 4. Values of J_2^{II} functional for fixed d_p, η.

Genetic algorithms

Genetic algorithms are part of a larger class of evolutionary algorithms and are classified as population based, global search heuristic methods (15, 16). Genetic algorithms are based on biological processes such as survival of the fittest, natural selection, inheritance, mutation, and reproduction. Design points are coded as "individuals" or "chromosomes", typically as binary strings, in a population. Through the above biological processes, the population evolves through a user specified number of generations towards a smaller fitness value. We outline the algorithm below.

1. Choose an initial population either randomly or by seeding a random initial population with points using an engineering perspective
2. Evaluate the fitness of each individual
3. Iterate (produce generations)
   a. Select individuals to reproduce
   b. Perform crossover and mutation
   c. Evaluate the fitness of the new individuals
   d. Replace the worst ranked individuals with the new offspring.

During the selection phase, better fit individuals are arranged randomly to form a mating pool on which further operations are performed. Crossover attempts to exchange information between two design points to produce a new point that preserves the best features of both "parent points". Mutation is used to prevent the algorithm from terminating prematurely to a suboptimal point and is used as a means to explore the design space.

Termination of the algorithm is based on a prescribed number of generations or when the highest ranked individual's fitness has reached a plateau. Genetic algorithms are often criticized for their computational complexity and dependence on optimization parameter settings, which are not known a priori. However, if the user is willing to exhaust a large number of function evaluations, the GA can help gain insight into the design space and locate initial points for fast, local single search methods.

For this work, we use a MATLAB implementation of the NSGA-II obtained from the Mathworks File Exchange (17). We point the reader to (18) for the details of the algorithm. We use 20 individuals in each of 5 generations, and we use the default parameter settings for this implementation.

NUMERICAL RESULTS

We assume that the polymer fluid transporting the debris particles has a power-law index (n) of 0.9. The density of the melt is assumed to be 0.00135 kg/cm^3. There are three materials associated with particle debris, based on possible sources for debris particles. The first material has

density 0.0089 kg/cm^3, the second, 0.004 kg/cm^3, and the third, 0.001 kg/cm^3. The sources for the debris particles include the polymer melt and bits of machinery that move into the fluid during processing. The relative concentration of the debris particles in the melt are assumed to be 1.5 ppm for the first material, 0.5 ppm for the second material, and 1 ppm for the third material.

The filter is assumed to be a sintered metal filter, so there is a parameter in the model that represents the average size of a filter particle. We have assumed a linear relationship between the filter pore diameter and the filter particle diameter. As the porosity increases, so does the particle diameter. This alters the retention properties of the filter, as larger particle diameters imply larger pore spaces. The filter is single-layered, with circular cross-section one inch in diameter and one centimeter thick.

For both sets of numerical results, we define the bound constraints for Ω as $L_1 = 0.1$, $U_1 = 0.7$, and $L_2 = 20\ \mu\text{m}$, $U_2 = 40\ \mu\text{m}$. For the genetic algorithm, we set an initial population size of 20 and allow for 5 generations. The threshold value for the pressure drop is 35e6 Pa.

J_2^I

The best parameters found to maximize the lifetime of the filter were $[\eta, d_p] = [0.7, 32\ \mu]$ which lead to a lifetime of 102.5 hours while allowing 6% of the debris to escape. The final filter profile is given in Fig. 5. As expected, the profile shows that most of the debris is trapped at the inflow. The profiles for the remaining optimal points had similar debris distributions and are thus not provided.

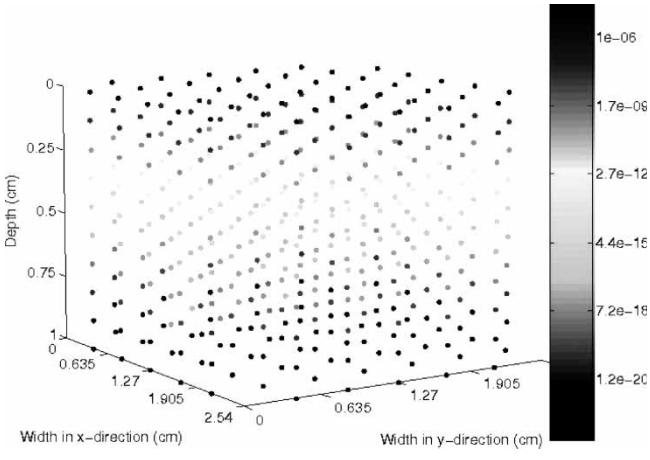


Figure 5. Mass fraction of trapped debris [0.68, 38.7 μm].

For minimizing the percent of debris through, $[\eta, d_p] = [0.61, 25.4\ \mu]$ which gave a filter lifetime of 62.3 hours allowing 5% of the debris to escape. Surprisingly, the parameters that lead to the largest percentage of escaped debris did not coincide with the longest filter lifetime. The parameters were $[\eta, d_p] = [0.59, 36.2\ \mu]$ which gave a lifetime of 77.9 hours while allowing 9% of the debris to escape the filter.

The tradeoff curve for optimizing Eq. [1] using J_2^I is given in Fig. 6. This scatter plot defines the Pareto points that an engineer could use to design or select a suitable filter for his process. For all the design points sampled, at least 5% of the debris escaped the filter. In particular, there is a cluster of design points that lead to a filter lifetime of approximately 60 hours while allowing approximately 5% of the debris to escape. Figure 6 indicates that using the percent of escaped debris as a measure of the filter performance does not allow for a straightforward decision. For example, there are a significant number of design points that allowed between 7–7.5% of the debris to escape while the filter lifetimes vary from 50 to 100 hours.

The scatter plots in Fig. 7 show profiles of both the filter lifetimes and the percentage of escaped debris as a function of the design variables. Note that in Figure 7(b), filter lifetimes of 60 hours and 100 hours correspond to a tightly clustered grouping of pore diameters and porosities. In Fig. 7(a), we see that the GA primarily sampled porosity values in the interval [0.6, 0.7] while the sampled pore diameter varied over the interval [24 μm, 40 μm].

J_2^{II}

The tradeoff curve for optimizing Eq. (1) using J_2^{II} is given in Fig. 8. The relationship between the total mass of the debris escaping the filter and

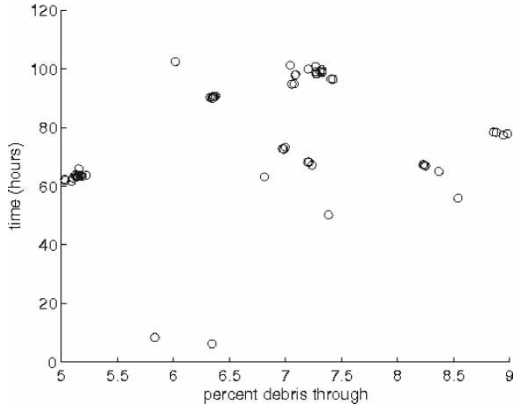
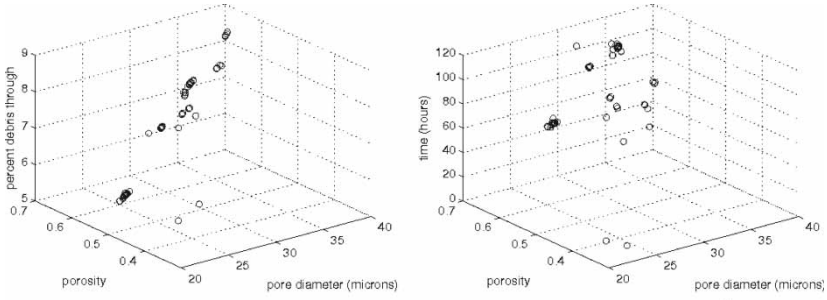


Figure 6. Tradeoff for J_2^I.



(a) Percent debris Scatter plot for J_1 (b) Lifetime scatter plot for J_2^I

Figure 7. Scatter profiles of objective functions.

the lifetime of the filter has a linear trend, which is quite different than the behavior seen in Fig. 6. In particular, J_2^{II} is the actual mass of the escaped debris in comparison to J_2^I which does not account for the fact that most of the debris escapes at the beginning of the simulation, as shown in Figure 3. The best parameters found to maximize the lifetime of the filter were $[\eta,\ d_p] = [0.68,\ 38.7\ \mu\text{m}]$ which lead to a lifetime of 110.2 hours with $1.1302\text{e} - 4$ kg of debris escaping.

For minimizing the total mass of debris through, $[\eta, d_p] = [0.1, 28.5\ \mu\text{m}]$. For these parameters, the filter simulation only took one time step (0.1 hour). Compared to the values found using J_2^I, these values imply that J_2^{II} is a better measure of the competing objectives for the filter performance since in particular both J_2^{II} and d_p are small in terms of minimizing the escaped mass of debris.

Figure 9 shows the profile for the lifetime of the filter as a function of the decision variables for J_2^{II}. The GA samples a wide range of porosity values, in [0.1, 0.7], while the pore diameters appear to be more clustered. The scatter

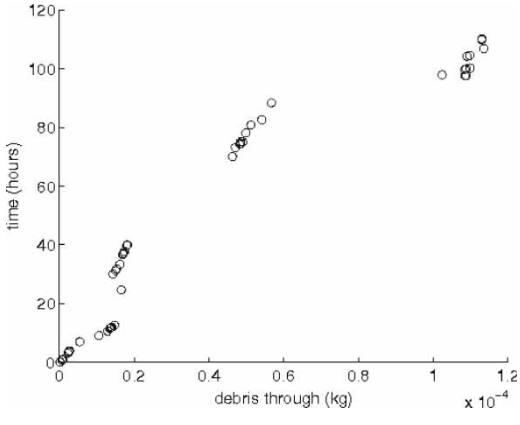


Figure 8. Tradeoff for J_2^{II}.

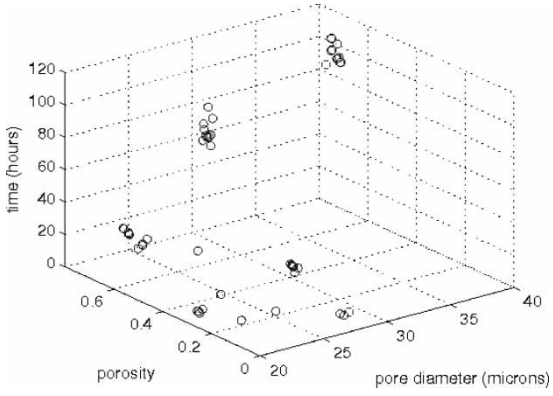


Figure 9. Scatter plot for J_2^{II}.

plots look similar when considering the mass of escaped on the vertical axis since the relationship between the amount of escaped debris and the filter lifetime is nearly linear.

Comparison of Filter Performance for Extreme Points

In this section we summarize the pressure drop behavior and the debris outflow distribution for the extreme points (i.e., optimal parameter sets) obtained from the optimization. Both the pressure drop history and the debris outflow distribution are measures of the filter performance but are not directly accounted for in the competing objectives. Recall that the best points found for maximizing the lifetime of the filter were $[\eta,\ d_p] = [0.7,\ 32\ \mu\text{m}]$, using J_2^I, and $[\eta,\ d_p] = [0.68,\ 38.7\ \mu\text{m}]$ for J_2^{II}. The best point for minimizing the percentage of escaped debris was $[\eta,\ d_p] = [0.61,\ 25.4\ \mu\text{m}]$. For minimizing the total mass of escaped debris, the best point found was $[\eta,\ d_p] = [0.1,\ 28.5\ \mu\text{m}]$; however, this parameter set resulted in a filter which lasted one time step (0.1 hour). Since the lifetime of the filter is so short for that parameter set, we exclude that point in this analysis.

Figure 10 shows the pressure drop behavior for the three parameter sets. As expected, the change in pressure drop is initially much more gradual for two parameter sets associated with the maximum lifetime of the filter. Viewing from left to right, the first curve in Fig. 10 is the pressure drop for the porosity/pore diameter pairing [0.61, 25.4 μm]. The second pressure drop curve is for the pairing [0.7, 32 μm], and the third curve is for the pairing [0.68, 38.7 μm].

Representative profiles for the three types of debris material are shown in Figs. 11 and 12. The left plot in Fig. 11 corresponds to inflow distributions

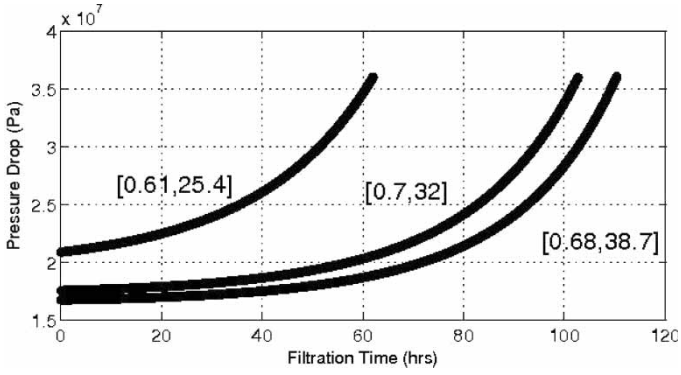


Figure 10. Pressure drop curves for optimal points [0.61, 25.4 μm], [0.7, 32 μm], and [0.68, 38.7 μm] (left to right).

while the three other plots depict the outflow distributions for the three extreme points. The outflow plots show that the filter captures debris particles larger than the prescribed pore diameter, d_p, while a fraction of the smaller debris escapes.

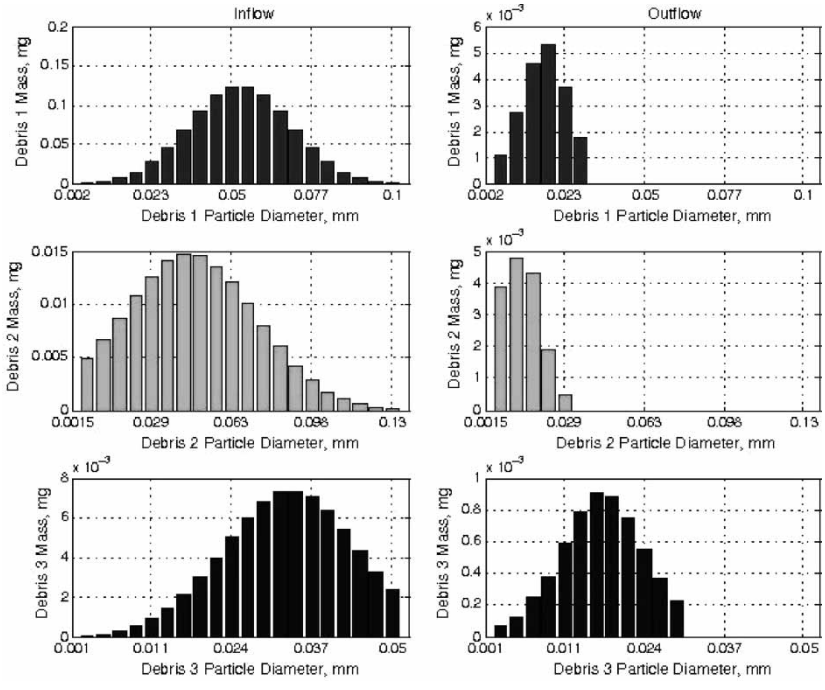


Figure 11. Inflow debris distribution (left bar charts) and outflow distribution for $\eta = 0.7$, $d_p = 32$ μm (right bar charts).

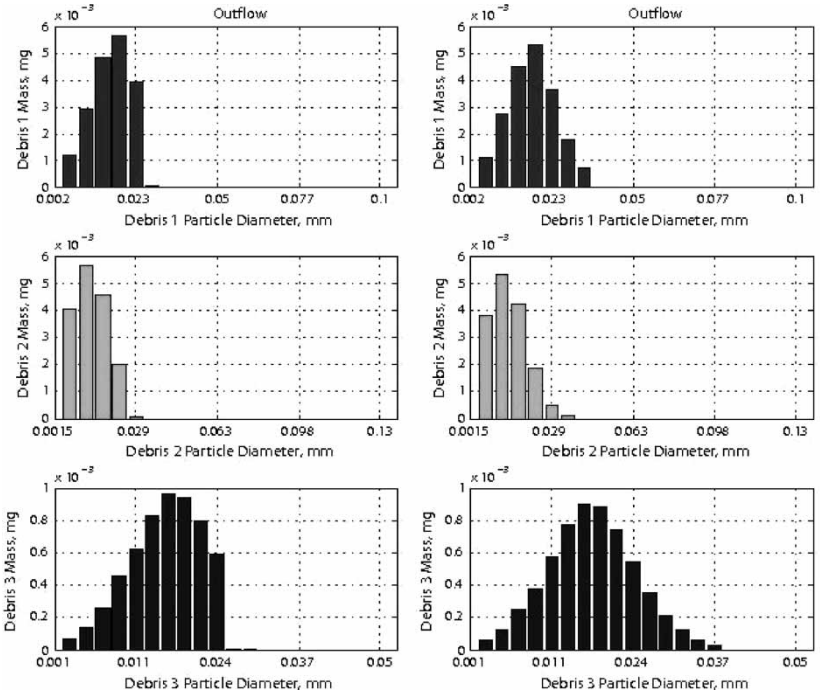


Figure 12. Outflow distribution for $\eta = 0.61$, $d_p = 25.4$ μm (left bar charts) and outflow distribution for $\eta = 0.68$, $d_p = 38.7$ μm (right bar charts).

CONCLUSIONS

We have analyzed a three-dimensional filtration model using a genetic algorithm to evaluate competing multi-objective functions. The multi-objective functions measure quantities that are important to process engineers involved in fiber manufacturing. This study has enabled us to better understand the tradeoffs inherent in filter design relative to the parameters we selected, porosity and pore diameter. The scatter profiles demonstrate that subsets of the design space give values for the objective functions that are minimally separated. These clusters of objective function values perhaps define the best resolution to the inherent objective conflict in this problem.

Our study underscores the usefulness of computational simulations for polymer and filter manufacturers. In a matter of days, we were able to analyze 100 design configurations for filters. The debris profile was specified, but we could easily switch our formulation to study 100 distinct debris profiles for a given set of filter parameters. The ability to focus a design study will help industry effectively study manufacturing processes and make cost-effective decisions.

A genetic algorithm was chosen for this work since gradient information is unavailable due to the black-box dependence on the simulator. The GA allowed for an "organized" search of the design space. The design space is small, though, as several other parameters exist that fully define the filter. A more complete study of the filtration process is warranted, which we believe is best accomplished using appropriate optimization algorithms. This study will be complicated due to the interactions between the design variables, but we believe it will provide valuable insight.

Towards this end, future work will incorporate other design parameters, including multiple filter layers, the thickness of individual layers, and the particle sizes and concentrations of the debris. Consideration of these additional variables will sharpen the focus of the design objectives and lead to better resolutions of the tradeoffs in filter design and usage. We also have not included costs for filter replacement or product loss. We expect to have objective functions associated with these costs in future studies.

In addition to increasing the number of decision variables, we will investigate penalty approaches to account for the escaped debris. Such formulations allow an engineer to specify a constraint on the amount of escaped debris which is incorporated as a penalty on the objective function. This will lead to a single-objective function, with costs associated with each component. Our optimization problem could then easily be expressed as minimizing cost, and we could also take advantage of single-search derivative-free optimization algorithms. These algorithms typically use fewer function evaluations than population-based approaches, which would reduce our computational costs.

The computational expense of the optimization may also be alleviated with the use of surrogate functions in which a pseudo-landscape is built using known objective function values (19). In that case, fast gradient-based methods can be applied to a pseudo-landscape that is updated by a structured exploration of the design space. This initial study and the smooth features of the landscapes imply this problem is a good candidate for a surrogate modeling approach.

This initial optimization study used generic data values which are characteristic of those used in practice. Future studies will be conducted with actual industry data to ensure the effectiveness of this approach in a realistic environment. Moreover, an additional simulator, written in C as opposed to MATLAB, is currently in development at CAEFF. The simulator implemented in C would help decrease simulation time and facilitate the use of more sophisticated optimization approaches already in use in industry.

ACKNOWLEDGMENTS

We thank Thomas Hemker of Simulation and Systems Optimization, Technische Universitat Darmstädt for many fruitful discussions. The authors thank graduate student Adrian Cox for help in implementing truncated

distribution functions. This work was partially supported by the ERC program of the National Science Foundation under Award Number EEC-9731680.

REFERENCES

1. CAEFF http://www.clemson.edu/caeff.
2. Hookway, D.C. (1996) How to design your deep bed polymer filter. *Filtration and Separation*, pp. 161–166.
3. Indo Rama Synthetics (India), Ltd. Analyst Meeting, Quarter 2 & H1 2006–07: Results and the opportunities ahead. http://www.indoramaindia.com/IRSL/Presenations/IRSL Q2%20%20FY2006%20-%.
4. Cox, C.L., Jenkins, E.W., and Mucha, P.J. (2005) Modeling of debris deposition in a polymer extrusion filter. In Proceedings of PPS-21, Leipzig, Germany.
5. Seyfzadeh, B., Zumbrunnen, D.A., and Ross, R.A. (2001) Non-Newtonian flow and debris deposition in an extrusion filter medium. In *Proceedings of Plastics–The Lone Star*; Society of Plastics Engineers, pp. 340–344.
6. Sawaragi, Y., Nakayama, H., and Tanino, T. (1985) Theory of Multiobjective Optimization. Number 176 in *Mathematics in Science and Engineering*; Academic Press, Inc.: Orlando, FL.
7. Steuer, R.E. (1986) *Multiple Criteria Optimization: Theory, Computations, and Application*; John Wiley and Sons: New York.
8. Srinivas, N. and Deb, K. (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2 (3): 221–248.
9. Cohon, J.L. (1978) *Multiobjective Programming and Planning;* Academic Press: New York.
10. Messac, A., Ismail-Yahaya, A., and Mattson, C.A. (2003) The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25: 86–98.
11. Savic, D. (2002) Single-objective vs. multiobjective optimisation for integrated decision analysis. In *Proceedings of iEMSs* 2002.
12. Edie, D.D. and Gooding, C.H. (1985) Prediction of pressure drop for the flow of polymer melts through sintered metal filters. *Industrial and Engineering Chemistry Process Design and Development*, 24: 8–12.
13. Bird, R.B., Stewart, W.E., and Lightfoot, E.N. (1960) *Transport Phenomena*; Wiley: New York.
14. Pearson, J.R.A. and Tardy, P.M.J. (2002) Models for flow of non-Newtonian and complex fluids through porous media. *Journal Non-Newtonian Fluid Mech.*, 102: 447–473.
15. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*; Kluwer Academic Publishers: Boston, MA.
16. Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*; The University of Michigan Press: Ann Arbor.
17. Seshadri, A. (2006) NSGA-II: A multi-objective optimization algorithm. http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId = 10429.
18. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2): 182–197.
19. Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V., and Trosset, M.W. (1999) A rigorous framework for optimization of expensive function by surrogates. *Structural Optimization*, 17: 1–13.