A Numerical Charge Optimization Framework for Methodical Charge Design of Advanced Sheet Moulding Compound Structures

Schwalm, H.[1,2], Harvey, D.[1,2], and Hubert, P.[1,2]*
[1] Department of Mechanical Engineering, McGill University, Montréal, Canada
[2] Research Centre for High Performance Polymer and Composite Systems CREPEC, Montréal, Canada
* Corresponding author (pascal.hubert@mcgill.ca)

Keywords: *Sheet Moulding Compound, Process Simulation, Compression Moulding*

1 INTRODUCTION

High-performance carbon fibre composite components made from continuous fibre laminates are not only known for their excellent specific strength and stiffness, but also for their laborious processing. A highly promising alternative is found in long discontinuous fibre composites based on a randomly oriented strands (ROS) architecture [1–3], often supplied as Carbon Fibre Sheet Moulding Compounds (CF-SMCs). These materials consist of chopped fibre tows randomly spread in-plane and are usually supplied pre-impregnated with a partially cured quick-curing thermoset resin. The ROS architecture allows for high fibre volume fractions with sufficiently long fibres to ideally transfer loads between fibre and matrix, while their discontinuous nature makes them usable in highly efficient compression moulding processes [1]. Along with the choice of carbon fibres as a reinforcement, these materials can be considered distinctly different from classical SMCs, usually reinforced by glass fibres at lower fibre contents. As such, CF-SMC strike an excellent balance between mechanical performance and processability, making them an option for medium to high production volume structural parts, where composites were previously too cost inhibitive to be considered.

ROS composites and their mechanical performance have been extensively studied at the coupon level. Their stiffness is usually comparable to their continuous fibre quasi-isotropic analogs, while strength is significantly lower [1,2,4]. Stress concentrations at the ends of the load bearing strands tend to trigger the matrix-dominated failure modes ROS are known for [3], making them seem brittle and notch-insensitive. Besides the specifics of the material architecture itself, the processing has a significant impact. Depending on the initial shape and placement of the charge inside of the mould, the flow paths and directions of the material vary, causing a preferential alignment of the strands, giving the final part anisotropic properties [1,5]. Furthermore, high in-mould flow has been linked to a more irregular morphology [2] with fibre waviness, kinking, swirling, and tow distortion, as well as defects such as matrix cracking, voids and, in extreme cases, excessive resin percolation with incomplete mould filling [5]. Due to the inherent heterogeneity of the material, these morphology irregularities are not necessarily the source of failure during testing [2,3], but they do contribute to the overall high variability in the data [6]. Finally, weld lines, where two or more flow-fronts meet during moulding have been shown to greatly decrease local strength [7]. The tight packing of long fibres resists intermingling of the strands, and the lack of reinforcement across the interface causes a strong performance knockdown compared to the otherwise well performing material. Consequently, charge design has been called one of the most important tools to control part performance during processing [7]. Yet, methods for charge design, especially for complex structures, have not been extensively explored.

To take full advantage of the potential of CF-SMCs, a low-flow moulding approach is often preferred [1,2]. Here, the charge already closely resembles the final part shape. In its most extreme form, this requires complex shapes to be cut from the SMC roll and preformed much like in continuous fibre processing [1]. This is in stark contrast to the simple bulk moulding approach used for classical SMCs, where the charge shape is not controlled. Wasting offcut material around those shapes directly contradicts the economical motivation behind SMCs. Hence, an optimal charge design method must consider both its impact on mechanical performance and its practical implications for the processing step.

To find the best solution to a given problem, most optimization schemes iteratively attempt to find the values of the design variable $x = [x_1 \quad \cdots \quad x_N]$ which minimizes the value of an objective function $f(x_1, \dots , x_N)$. This function, also often called cost function or fitness function, quantifies how well the objective is met during each iteration. The choice of optimization scheme, which governs how the design variables are altered based on the cost, has a sizable impact on the computational performance of the problem [8]. Most relevant to this work are gradient-based optimization and evolutionary optimization, also called genetic algorithms. As the name suggests, the former requires the computation of the derivative of the cost and then steps the design variables along the downward direction of that gradient. However, derivatives may not be available in many cases and the algorithms tend to get stuck in local minima. Evolutionary optimization instead processes multiple combinations of design variables as part of a population and then evolves the population by recombining and mutating the fittest solutions. This biomimetic approach may require more runs than gradient-based algorithms, but requires less in-depth knowledge of the problem, lends itself well to parallelization of the runs within a population and can be extended to multi-objective optimization [9].

In the case of the SMC charge design problem, evaluating the objective function is a heterogeneous and complex problem with multiple possible implementations. Furthermore, the resulting charge design is intended to be used by an automated production line and further optimized based on the feedback from it. Hence, data exchange within the framework bridges multiple different hardware and software systems and eventually controls robotic components. In the field of robotics, skill-based software architectures have recently gained traction as a way to tackle such requirements [10,11]. They hierarchically decompose the capabilities of a system into skills, modelling them as human-like abilities that can be orchestrated to fulfill a task. Primitive skills, represent the most basic capabilities of the system as intuitive symbolic units. Skills are combinations of primitive skills and can be used as solution-neutral building blocks for a task [11].

This paper thus presents the software framework at the core of a numerical optimization-based approach that is currently being developed to methodically design CF-SMC charges. Its software architecture is detailed in the following. A basic example for its use is introduced, which considers processing effort as well as the implications processing has on mechanical performance.

2 SKILL-BASED OPTIMIZATION SOFTWARE FRAMEWORK

Ultimately, the proposed framework will be implemented as a control system for a highly integrated and automated production line for CF-SMC components. For any given part design, a charge design is calculated and optimized. From its solution, instructions for automatic ply cutters and robot arms assembling the charge are extracted. Autonomously converting a part design into an CF-SMC charge and optimizing this configuration can be done in a plethora of ways using different objectives and design variables. The given example focuses on balancing the trade-off between processing effort and flow induced defects purely based on numerical process modelling, but the framework is also designed to be easily reconfigurable. For instance, it can be restructured to pursue the same objective based on on-line sensor feedback or to optimize the charge for a desired stiffness anisotropy through guided flow.

That also means that it must adapt to the capabilities of the system it is being executed on, depending on its respective computational capabilities or whether it is connected to a real production system and which robotic actuators, or sensor systems are available. This is enabled by the modular skill-based software architecture shown in Figure 1. The framework is split into three layers. At the lowest level, the primitive skillset of the system contains the actual bits of software to be executed, each comprising one basic capability only, and wrapped in semantically coherent skills to ensure their compatibility. The order in which these skills are executed is defined on the superordinate scheduling layer. Here, tasks and skills are modelled as finite state machines to modularize the capabilities of the system. The planning layer houses the task controller and skillset manager, which are responsible for the assembly and supervised execution of the task, respectively. The inner workings of the layers are explained in more detail in the following. The example application shown in Figure 1 is explored in Section 3.

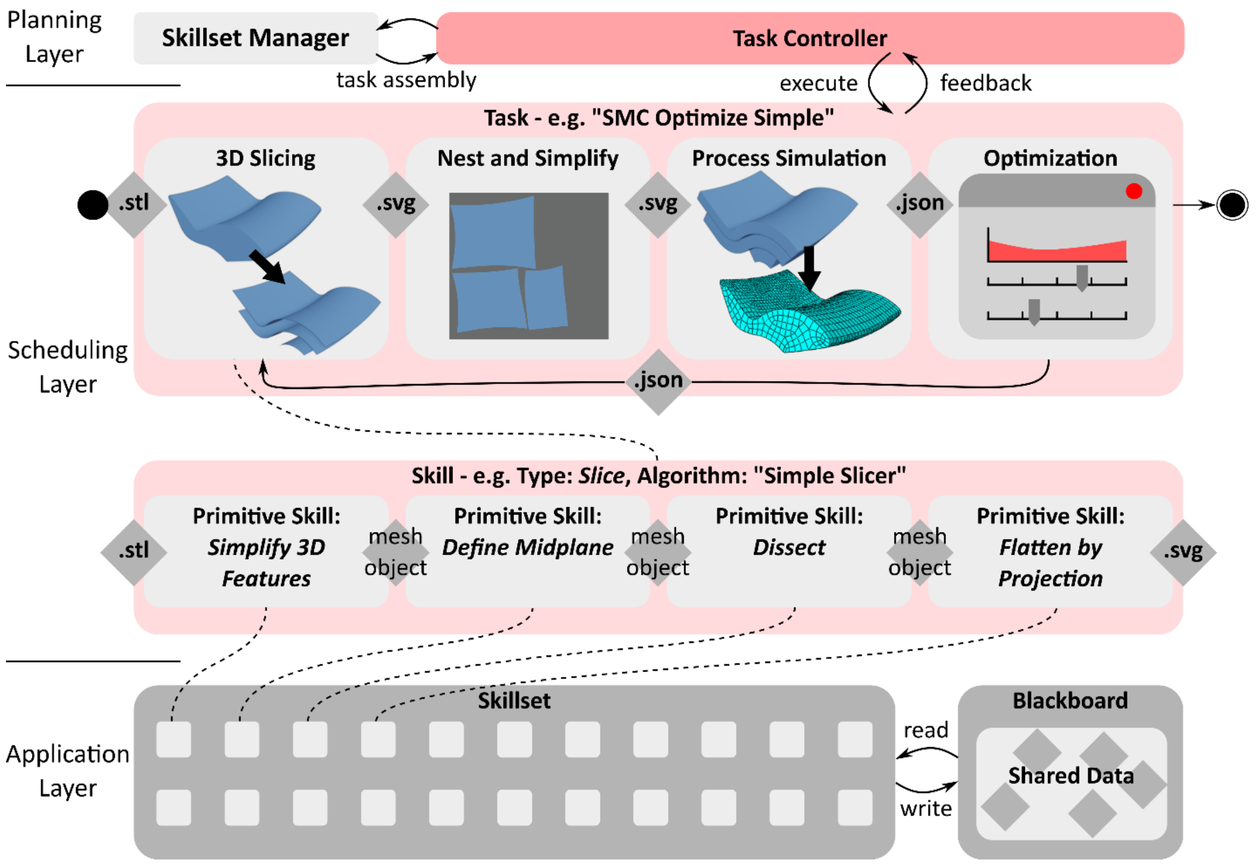

Figure 1: Skill-based software architecture for charge design optimization. The shown task is exemplary and can be arbitrarily reconfigured.

2.1 Use of Skills

As in similar architectures, primitive skills contain the capabilities of a system on its most basic level, where a further subdivision would be of no added benefit to its usability. Bringing these capabilities into a shared context is imperative, especially since SMC charge design optimization inherently marries multiple fields, from computational geometry over numerical solid mechanics and fluid simulation to robotics and automation. A skill can thus either contain all its required functions directly, or simply act as a wrapper for the software it controls via the software's application programming interface. It may also implement the control of and the communication with a robot or sensor system and parse the exchanged data into a format usable for the optimization. Primitive skills are implemented as classes that inherit basic functionalities for data interfacing and communication with the task controller. This encapsulation allows for their use without detailed knowledge of their inner workings and their reusability at different stages of the optimization.

2.2 Modularization on Multiple Levels of Abstraction

Building on the abstract modelling of the system capabilities through primitive skills, the scheduling layer is meant to further modularize and, thus, simplify the operation of the system. In fact, the states of the task state machine do not even refer to specific pieces of software. A task simply defines the type of skills to be executed, their order, and the format of the data handed from one skill to the next. As such, it may feature concurrencies and multiple optimization loops, which are easily configured without the need to program these features. The specific skill suiting a certain type features the same arbitrary state machine structure with defined data interfaces as tasks and can contain its own optimization loops. Here, two skills of the same type do not have to feature the same state machine structure, to allow for diverse problem-solving approaches, since highly sophisticated algorithms can likely benefit from more sub-division than a crude implementation addressing the same issue. The modularity of the framework allows for the quick and easy recombination of different skills to try out different methods based on the available software and hardware options. It also makes it easy to iteratively improve the implementation step-by-step, by working on one skill at a time.

2.3 Task Control and Assembly

All new skills and primitive skills are registered with the skillset manager, either automatically at start-up or manually during run-time. The latter is practical for long runs, where the improved implementation of a skill can substitute the old one. Once the task controller is prompted to start executing a task, it configures the state machine and its transitions and then requests specific skills for all the required types of skills from the skillset manager. From then it executes and supervises the execution of the skills and their subordinate primitives. From the standardized control interface built into each skill and primitive, its status, data inquiries and outputs are available to the task controller. Besides passing on data from one skill to the next, it also grants the skills access to shared data on a blackboard object. Based on the status feedback, it addresses the skills in the predefined order.

3 CHARGE DESIGN APPLICATION EXAMPLE

The approach introduced in the following represents a basic implementation of the charge design optimization problem, using minimal design variable dimensions, and is only based on modelling the process. Keeping in mind the modular nature of the framework, more optimization loops within or following the current one may be introduced and on-line process optimization through feedback from real-world processing data can be implemented.

3.1 Optimization Scheme

The bulk moulding approach for preparing conventional SMC charges is not controlled enough for low flow moulding or any processing concept in which the charge shape is deliberately adapted to control the final part attributes. Rather, the charge is assembled from multiple plies cut from the SMC roll and stacked to form a macro-scale laminate. How accurate and intricate this laminate is, can be broadly quantified by the charge coverage φ_{cover}. Depending on the exact implementation of the skills, the design variable x may either be linked to φ_{cover} directly, or to a multi-dimensional design variable x, where:

$$\varphi_{\mathrm{cover}} = f(x_1, \dots, x_N). \tag{1}$$

The interdisciplinary nature of the charge optimization, in which a small change in the design variable can likely cause an algorithm to take a completely different approach to a problem, means it is unlikely that smooth gradients exist in the solution space. Hence, evolutionary optimization algorithms are favored in this example, especially for heterogeneous multi-dimensional design variables.

3.2 3-Dimensional Slicing Skill

Since compression moulding can be used for complex 3-dimensional (3D) parts, the shape of the plies of the macro-scale laminate are initially generated by a process inspired by the slicing algorithms used in additive manufacturing. Initially, the 3D part design is provided as a universally usable geometry file format. Its rotation is predetermined such that its vertical axis coincides with the mould closure direction and no design features, like undercuts, interfere with demoulding (Figure 2a).

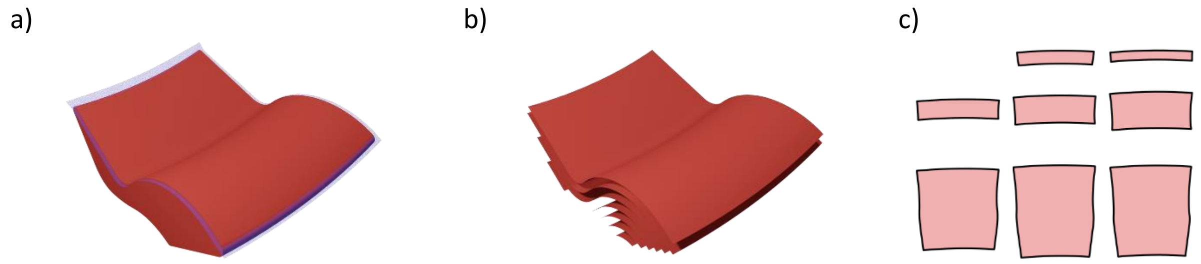

Figure 2: 3D slicing concept. a) Part design with reference plane marked in transparent blue. b) Slices in 3D-space based on the curvature of the reference plane. c) Flattened slices to be passed to the subsequent skill.

As the uncured SMC plies are drapable, the preform can be curved to better approximate the part shape. Unlike in traditional slicing algorithms that dissect the part based on a flat reference plane, this 3D-slicing requires the definition of a curved initial reference surface that is done either manually or algorithmically. Its in-plane direction is also the in-plane direction of the SMC plies, meaning it dictates the initial orthotropic fibre orientation distribution. A first slice of the geometry is found as the intersection between the reference surface and the part geometry. Multiple different algorithms for such Boolean operations for 3D geometries have been proposed and implemented in different software and programming modules. The modular nature of the skill-based architecture allows for easy benchmarking of different approaches. Two duplicates of the reference surface are shifted equal distances up and down the vertical axis and the slicing is repeated until it the surfaces no longer intersect the part (Figure 2b). Extruding these slices represents the extreme case of a highly accurate charge, which might not be feasible. Instead, the aim is to optimize the actual plies generated based on the shapes of the slices.

To evaluate and adapt the shape of the slices into usable SMC plies, their curved shapes are flattened to yield planar shapes, as shown in Figure 2c. This step essentially mimics the reverse of the draping step of the flat SMC plies into the curved preform. It can be achieved through different means with varying degrees of complexity, ranging from a simple projection onto a plane to numerical draping simulations. As is the case with the specific implementation of each step, the choice of the respective primitive skill is a balance of computational cost versus required intricacy of the solution.

It is worth noting that the slicing needs to occur during each instance of the optimization loop, since a given charge coverage φ_{cover} affects the distance d between slices. For a 100 % coverage configuration, d will be equal to thickness t_{SMC} of the chosen SMC, yielding a preform reminiscent of an additively manufactured part. For any lower amount of coverage, however, more slices are needed to assure the sufficient material volume of the charge. This is achieved by setting:

$$d = \varphi_{\text{cover}} t_{\text{SMC}}, \tag{2}$$

as illustrated in Figure 3a. As the actual plies are of thickness t_{SMC}, this eventually results in a preform that resembles the part design, but is simpler, slimmer and taller (Figure 3b). The assumption that the plies will mainly flow along the surface geometry of the slices to fill the mould (Figure 3c) is fair for very high degrees of coverage but might not apply to simpler charges. Hence, accurate numerical process modelling is imperative (Section 3.4).

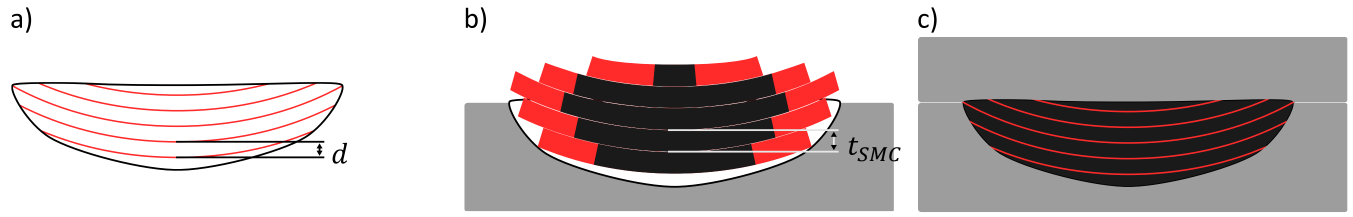

Figure 3: From part design to charge design. a) Part cross section with slices marked in red. b) Charge in bottom mould with extruded slices marked in red and SMC plies marked in black. c) After compression moulding, the plies effectively have the thickness of the initial slices.

3.3 Nesting and Simplification Skill

Simplifying the shape of the slices aims to decrease the cost of processing. In practice that means mainly avoiding large amounts of offcuts and complicated ply shapes that take a long time to cut. Both offcuts and cutting time can be quantified using 2-dimensional (2D) ply nesting algorithms as are used for many CNC applications. The underlying task is an irregular 2D bin packing problem, where a number of polygonal shapes, in this case all of the SMC plies, are fit into a larger rectangular shape representing the SMC roll. As an alternative to commercial software, freeware versions have been released, and solutions from literature can be implemented. Most are based on heuristics consisting of a placement strategy and an optimization of the order in which the shapes are inserted, to occupy the least amount of space possible in one direction. They return the nested configuration of the ply shapes to be evaluated as part of the overall charge optimization. Assuming the nesting algorithm pulls all plies to the left, we define the area of the used SMC material as rectangle with the same width of the roll and the length of the rightmost point of the nested shapes, measured from the beginning of the roll. How efficiently the material is used with a certain charge configuration is given by the offcut ratio:

$$\varphi_{\text{cut}} = \frac{A_{\text{rect}} - A_{\text{part}}}{A_{\text{part}}}, \tag{3}$$

where A_{rect} is the area of the rectangle and A_{part} is the area of SMC needed completely fill the volume of the part including some predefined excess material for flashing. The cutting effort can be evaluated based on the accumulated outline length of all plies, where coincident lines stemming from a good nesting result, translate to saved cutting time, since they only need to be cut once. Actually simplifying the shape of the slices into well nesting plies can be done in many ways and, if implemented algorithmically, represents an intricate computational geometry problem. The best fitting algorithm and the rules that govern the same, based on applied composite materials knowledge, will be further investigated in the future. Note that φ_{cover}, which controls the area of each ply, might have to be adjusted to achieve A_{part}, since the slicing algorithm returns an integer number of slices with varying sizes.

3.4 Process Simulation Skill

To evaluate the quality of the charge, the simplified plies are mapped back onto the plies in 3D-space and extruded to form a 3D-representation of the charge design. Combined with the original part design as the mould, it can then be used in a compression moulding process simulation. Such simulations for short fibre composites are widely used in industry and implemented in commercially available simulation tools. In the case of ROS, the long fibres and high fibre content significantly complicate the flow behaviour, limiting the applicability of these tools. Nonetheless, proven methods of varying computational cost exist in the literature [12,13]. The ideal choice of process model for the given optimization application will depend on the required level of detail of its output and its respective computational efficiency, as optimization requires multiple runs. In any case, a Lagrangian displacement field U describing the discretized flow of the charge during moulding harbours the necessary performance metrics for the optimization. Modelling the exact stresses is not necessary for this initial implementation.

3.5 Objective function development

The overall objective of the optimization is to find the best possible price-to-performance ratio in the processing of a given CF-SMC part. Given the most relevant processing aspects outlined above, that means identifying the sweet spot between minimizing the amount of material wasted due to offcuts and minimizing the performance knockdown due to flow induced defects. Thus, the objective function to be minimized is defined as follows:

$$\mathrm{cost} = f(\varphi_{\mathrm{cut}}) + f(U), \tag{4}$$

The purpose of the two terms is to identify the relevant value ranges of their respective constituents, normalize and weigh them to bring them into the same industrially applicable context. In the case of the φ_{cut}, this is relatively straightforward, as the cost of the wasted material can be directly related to the ideally predicted overall production cost per part. While the given approach considers offcuts as lost, reusing them for the bulk moulding of non-structural parts may be an interesting approach to explore in the future. Furthermore, the costing calculation will consider the preforming time needed per part, including the cutting and stacking times of the plies, which are expected to elongate the overall cycle time with added complexity. The actual impact of the simulated in-mould flow on the part quality, however, is non-trivial and will thus be represented by a quantifiable proxy derived from U.

Both objective function terms must be weighed on a case-by-case basis, depending on the application of the part. For example, when producing safety-critical parts that push the strength boundaries of the material, the cost is expected to be less dominating factor. For parts in stiffness driven applications – a property which is not as affected by high-flow moulding – cost is likely much more relevant.

Since the nesting algorithm described in Section 3.3 is a non-trivial geometric problem, its solution $f(\varphi_{cut})$ is likely not continuous. Once a gradual change in the design variable results in one of the ply dimensions exceeding the width of the sheet, for example, the previously optimal nesting pattern may no longer be eligible and the new solution can result in a much higher amount of offcuts and, thus, a lower fitness. Furthermore, as the implementation of these problems often relies on genetic algorithms, it may not even be necessarily deterministic. Given enough time to converge and a sufficient design space, however, it is to be expected that simpler shapes will overall trend toward a lower amount of wasted material with reasonable repeatability of the heuristic solution.

Since $f(U)$ involves modelling complex physical phenomena numerically, some degree of discontinuity may be expected due to a change in geometrical features triggering unforeseen flow interactions. Yet, assuming Stokes flow due to the high viscosity of the matrix material, their impact on the displacement and strain within the mould should be minor. These considerations affirm the potential of differential evolution as an initial optimization scheme. Figure 4 shows the outcome of a simple test of the optimization framework with crude models as its skills. Using φ_{cover} directly as the design variable, we expect the contribution of U to the cost to decrease with increased coverage and the contribution of U to increase. They are for now modelled as arbitrary cubic functions of φ_{cover}. Adding the unevenness of the nesting result as static 1-dimensional Perlin noise, the overall cost features multiple local minima and one global minimum. By adding some dynamic noise to the outcome of every run, we represent the non-deterministic behavior of the nesting. For a population size of 10, a common differential evolution algorithm approximates the minimum well, despite the noise. The required number of iterations heavily depends on how the tolerance of the convergence criterion, i.e. the standard deviation of the population after convergence, compares to the magnitude of the dynamic noise. The shown optimization task converged within 8 iterations, with the tolerance and the magnitude of the dynamic noise both set to 0.04.

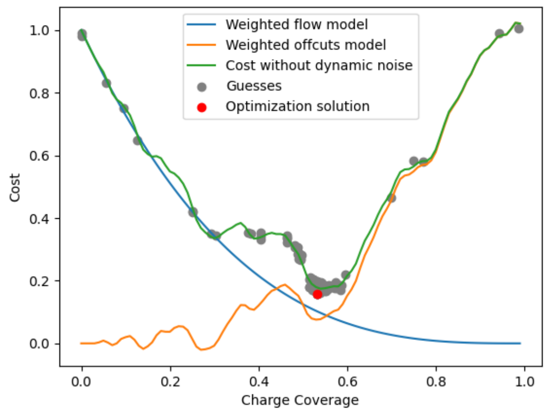

Figure 4: Evaluating the objective function using simple arbitrary models as skills.

3.6 Benchmarking

To validate the applicability of the optimization framework to a charge design problem, a simple benchmark is introduced. The proposed benchmark is designed to be non-analytical and feature the complexities of the charge design application, yet it still has a clearly optimal solution that the computation must find. Consider a simple extrusion with uniform thickness of $30t_{\mathrm{SMC}}$ and curvature as shown in Figure 5a.

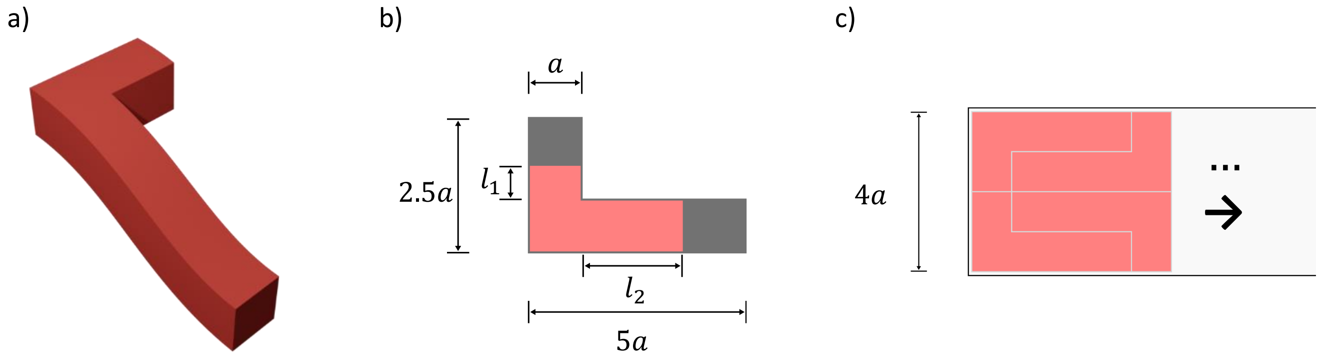

Figure 5: L-shape benchmark problem. a) Part design. b) Slice marked in grey and simplified ply marked in red. c) Nested configuration of ideal plies requiring low flow and no offcuts.

Using a reference plane with the same curvature as the part surface, each slice returned by the 3D slicing algorithm has the identical L-shape shown in Figure 5b. All SMC plies are cut from a roll with width $4a$. The limited complexity of the slices results in a significant simplification of the 2D ply design problem: A rectangular shape fitting the longer leg of the L-shape will result in no offcuts, but the material will have to flow to fill the mould. The other extreme would be to cut plies that completely cover the slice, but will result in offcuts, as they cannot be arranged in a rectangular shape within the confines of the roll dimensions. The trade-off solution considered in this example uses

L-shaped plies as shown in Figure 5b, and varies the length of both legs l_1 and l_2 to find an optimum balance between offcuts and flow. If a cost of $3\varphi_{\mathrm{cut}}$ is assigned to the wasted material due to offcuts, and a cost of $\max(U_{\mathrm{leg1}}) + \max(U_{\mathrm{leg2}})$ is assigned to the performance knockdown due to the flow, the optimization should result in the configuration depicted in Figure 5c, since it does not require offcuts and only requires a maximum in-plane flow distance of $0.5a$. Here U_{leg1} and U_{leg2} refer to the parts of the displacement field that belong to the legs of the L-shape, respectively.
The benchmark was run based on a differential evolution algorithm with a population size of 10. Since the number of plies n is an integer, l_1 and l_2 are not completely independent and instead, the design variable $x = [l_1 \quad n]$ was chosen. The n shapes resulting from the 3D-Slicing skill were shortened to form the plies, which were then transferred as vector graphics into a nesting algorithm based on open-source software. The nesting uses a no-fit-polygon placement strategy and evolutionary optimization for the placement order. To keep compute times manageable, the nesting was constrained to 10 s for each run. The processing simulation was significantly simplified, to concentrate on the remaining aspects of the optimization. Assuming each of the identical plies stays in-plane, their in-mould flow becomes a 2D-problem, applicable to FEA. Assuming Newtonian behavior, Stokes flow, and complete mould filling, we can induce the flow as a Dirichlet boundary conditions, which alleviates the need for exact material parameters [12]. The guesses leading to a converged solution and their respective fitness values are pictured in Figure 6a.

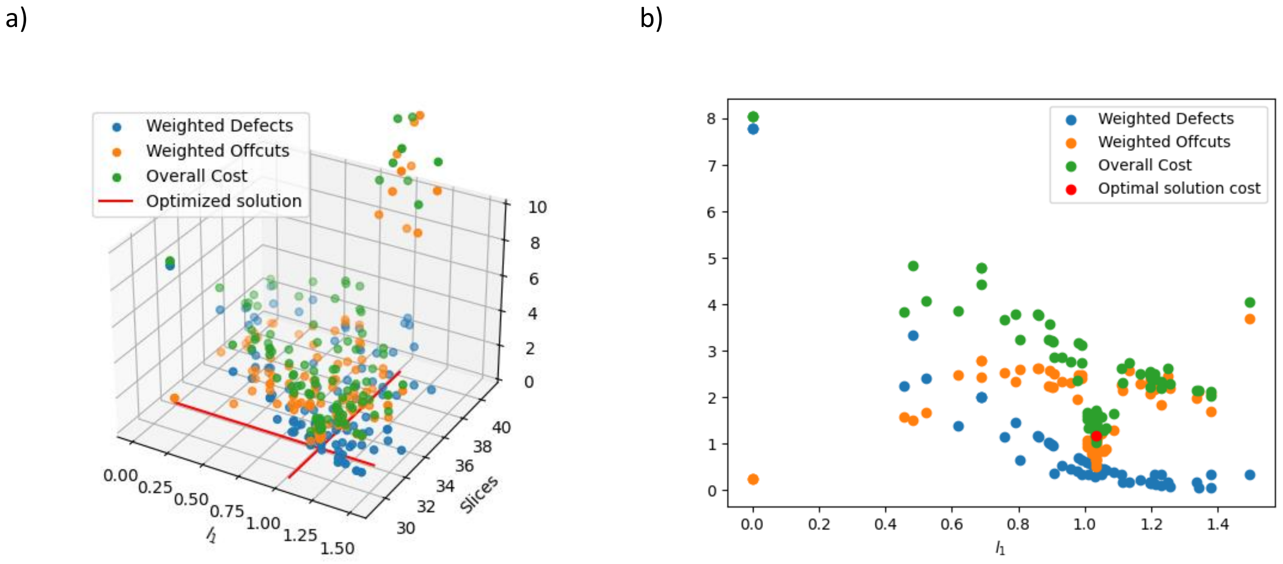

Figure 6: Optimization results of L-shape benchmark. a) Cost results based on the two design variables. b) Isolated l_1 design variable only showing runs with $n = 32 \mp 2$.

As expected, the optimum values found using the framework are $l_1 \approx 1$, and $n = 32$, corresponding to the benchmark solution. Isolating the l_1-axis, and only showing guesses where $n = 32 \mp 2$, yields a visualization of the optimization steps as in Figure 6 b. Like in the arbitrary model in Figure 4, the weighted flow shows an overall downward trend, and the weighted offcuts have an overall upward trend, with some local minima where favorable nesting configurations are possible. The expected differences in the weighted offcuts due to the non-deterministic nature of the nesting algorithm can also be observed. Over 5 different runs, the average cost of the optimized solution was 0.90 with a standard deviation of 0.15. This contrasts with the theoretic optimal solution, which has cost 0.59 associated with it. This can be attributed to the nesting algorithm, which usually cannot find the theoretic optimal nesting pattern within the 10 s time frame.

4 CONCLUSIONS

An initial overview of the optimization system being developed to design CF-SMC charges was presented. The software framework facilitates easy expandability into the realms of compression moulding process simulation and online process monitoring and control. Marrying the two is expected to facilitate sizable gains in control over the part performance while managing the industrial feasibility of the process. An initial implementation of the charge design optimization loop was introduced, and the background of each step was laid out. A simple benchmark has demonstrated the overall feasibility of different software interacting within the framework.

5 ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada as well as PRIMA Québec. We kindly thank our industrial partner for their continued support and guidance.

6 REFERENCES

[1] Visweswaraiah SB, Selezneva M, Lessard L, Hubert P. Mechanical characterisation and modelling of randomly oriented strand architecture and their hybrids – A general review. J Reinf Plast Compos 2018;37:548–80.

[2] Martulli LM, Muyshondt L, Kerschbaum M, Pimenta S, Lomov S V, Swolfs Y. Carbon fibre sheet moulding compounds with high in-mould flow: Linking morphology to tensile and compressive properties. Compos Part A Appl Sci Manuf 2019;126.

[3] Johanson K, Harper LT, Johnson MS, Warrior NA. Heterogeneity of discontinuous carbon fibre composites: Damage initiation captured by Digital Image Correlation. Compos Part A Appl Sci Manuf 2015;68:304–12.

[4] Feraboli P, Peitso E, Deleo F, Cleveland T, Stickler PB. Characterization of prepreg-based discontinuous carbon fiber/epoxy systems. J Reinf Plast Compos 2009;28:1191–214.

[5] Evans AD, Qian CC, Turner TA, Harper LT, Warrior NA. Flow characteristics of carbon fibre moulding compounds. Compos Part A Appl Sci Manuf 2016;90:1–12.

[6] Li Y, Pimenta S, Singgih J, Nothdurfter S, Schuffenhauer K. Experimental investigation of randomly-oriented tow-based discontinuous composites and their equivalent laminates. Compos Part A Appl Sci Manuf 2017;102:64–75.

[7] Martulli LM, Kerschbaum M, Lomov S V., Swolfs Y. Weld lines in tow-based sheet moulding compounds tensile properties: Morphological detrimental factors. Compos Part A Appl Sci Manuf 2020;139:106109.

[8] Nikbakt S, Kamarian S, Shakeri M. A review on optimization of composite structures Part I: Laminated composites. Compos Struct 2018;195:158–85.

[9] Fengler B, Kärger L, Henning F, Hrymak A. Multi-Objective Patch Optimization with Integrated Kinematic Draping Simulation for Continuous–Discontinuous Fiber-Reinforced Composite Structures. J Compos Sci 2018, Vol 2, Page 22 2018;2:22.

[10] Herrero H, Moughlbay AA, Outón JL, Sallé D, de Ipiña KL. Skill based robot programming: Assembly, vision and Workspace Monitoring skill interaction. Neurocomputing 2017;255:61–70.

[11] Heuss L, Blank A, Dengler S, Zikeli GL, Reinhart G, Franke J. Modular Robot Software Framework for the Intelligent and Flexible Composition of Its Skills. IFIP Adv. Inf. Commun. Technol., vol. 566, 2019, p. 248–56.

[12] Favaloro AJ, Sommer DE, Denos BR, Pipes RB. Simulation of prepreg platelet compression molding: Method and orientation validation. J Rheol (N Y N Y) 2018;62:1443. https://doi.org/10.1122/1.5044533.

[13] Teuwsen J, Hohn SK, Osswald TA. Direct fiber simulation of a compression molded ribbed structure made of a sheet molding compound with randomly oriented carbon/epoxy prepreg strands—a comparison of predicted fiber orientations with computed tomography analyses. J Compos Sci 2020;4.