AN ENGINEERING DESIGN STRATEGY FOR RECONFIGURABLE PRODUCTS THAT SUPPORT POVERTY ALLEVIATION

Patrick K. Lewis  
PhD Student  
Dept. of Mechanical Engineering  
Brigham Young University  
Provo, Utah 84602  
Email: elderpat@gmail.com

Christopher A. Mattson*  
Assistant Professor  
Dept. of Mechanical Engineering  
Brigham Young University  
Provo, Utah 84602  
Email: mattson@byu.edu

Vance R. Murray  
Research Assistant  
Dept. of Mechanical Engineering  
Brigham Young University  
Provo, Utah 84602  
Email: vancemurray17@gmail.com

ABSTRACT

Reconfigurable products can adapt to new and changing customer needs. One potential, high-impact, area for product reconfiguration is in the design of income-generating products for poverty alleviation. Non-reconfigurable income-generating products such as manual irrigation pumps have helped millions of people sustainably escape poverty. However, millions of other impoverished people are unwilling to invest in these relatively costly products because of the high perceived and actual financial risk involved. As a result, these individuals do not benefit from such technologies. Alternatively, when income-generating products are designed to be reconfigurable, the window of affordability can be expanded to attract more individuals, while simultaneously making the product adaptable to the changing customer needs that accompany an increased income. The method provided in this paper significantly reduces the risks associated with purchasing income-generating products while simultaneously allowing the initial purchase to serve as a foundation for future increases in income. The method presented builds on principles of multiobjective optimization and Pareto optimality, by allowing the product to move from one location on the Pareto frontier to another through the addition of modules and reconfiguration. Elements of product family design are applied as each instantiation of the reconfigurable product is considered in the overall design optimization of the product. The design of a modular irrigation pump for developing nations demonstrates the methodology.

Nomenclature

\( \delta \)  
Matrix dictating the desired progression that each module provides

\( D \)  
Set containing all design variable values

\( D_m \)  
Set containing all variable values of \( x_m \) and \( x_p \)

\( D^{(i)} \)  
Design \( (i) \) of set \( D \)

\( g \)  
Vector of inequality constraints

\( h \)  
Vector of equality constraints

\( \mu \)  
Vector of design objectives

\( n_d \)  
Number of designs comprising the adaptive design set

\( n_g \)  
Number of inequality constraints

\( n_h \)  
Number of equality constraints

\( n_\mu \)  
Number of design objectives

\( n_{\hat{\mu}} \)  
Number of additional objective constraints needed to define anticipated regions of interest

\( n_m \)  
Number of modules desired

\( n_x \)  
Number of design variables

\( n_a \)  
Number of adjustable design variables

\( n_mm \)  
Number of module design variables

\( n_{fp} \)  
Number of platform design variables

\( P \)  
Objective space performance of the base and target designs used to develop modules

\( P^{(i)} \)  
Objective space performance of a design when used with the \( i \)-th module

\( \Delta P^{(i)} \)  
Change in objective space performance from the base design to \( P^{(i)} \)

*Address all correspondence to this author.


1 Introduction

There is increasing evidence that one of the most sustainable ways to help those living in extreme poverty (20.5% of the world’s population who live on less than ~$1 a day) is through a market-based approach – where all in the supply chain benefit financially, including the poor [14, 18, 24, 52]. Among the most promising methods of producing profit for all in the supply chain is the development of products that increase the earning power of those that are living in extreme poverty [1, 14, 40]. Products such as treadle pumps, water drip irrigation kits, and coconut oil presses have generated millions of dollars in profit for poverty stricken countries and helped over 12 million people sustainably escape poverty [1, 14, 40]. However, millions of other impoverished people throughout the world are unwilling to invest in relatively costly income-generating products (~$100 – or 3 months of income) because of the high perceived and actual financial risk involved [14, 20, 52]. Additionally, a majority of the population cannot afford the investment under the traditional approaches, and therefore remain unaided by these poverty alleviating technologies.

Reconfigurable products, those that can adapt to new and changing customer needs, have the potential to significantly alter the impact that income-generating products can have on poverty alleviation efforts. An unequivocal need, as described by those leading the market-driven approach to sustainable poverty relief, is a method for bringing income-generating products to more people by reducing the risk associated with the investments [14, 23, 43]. The risks, irrevocably connected to product cost, would be effectively mitigated with engineering methods that embrace extreme affordability. When income-generating products are designed to be reconfigurable, the window of affordability can be expanded to attract more individuals, while simultaneously making the product adaptable to the changing customer needs that accompany an increased income over time.

Under the engineering design strategy presented in this paper, numerical optimization is used to identify a set of desirable designs that meet the varying needs of those living in extreme poverty. Moreover, the set of designs are selected based on the ability of the design to be realized though the addition of modules or reconfiguration. The goal of the strategy is to allow for a changing tradeoff preference between income generation potential and investment cost. As shown in this paper, this strategy significantly reduces the risks associated with purchasing income-generating products while simultaneously allowing the initial purchase to serve as a foundation for product and income expandability. Ultimately, the presented method allows income-generating products to be made more affordable by their strategic modularity and reconfigurable nature, which makes them progressively expandable, thus facilitating incremental investment and extreme affordability.

The remainder of this paper is presented as follows: A review of literature forming an enabling foundation for the development of the proposed method is included in Section 2. In Section 3, the theoretical development of the proposed design method is presented. In Section 4 the design of a manually powered irrigation pump is used to demonstrate the method. Concluding remarks are provided in Section 5.

2 Literature Survey

This section provides a brief review of previous pertinent research, and establishes a foundation for the strategy presented in the following section for designing reconfigurable products that support poverty alleviation.

2.1 Engineering-based Poverty Alleviation

The literature characterizes poverty alleviation as providing those trapped in poverty with the ability to permanently escape [14, 17, 49, 51]. To provide this, sustainable development is required. Sustainability, however, is a complex attribute that involves economic, social, environmental, technological, and political factors [4, 40]. A close examination of the past 20 years of literature shows that evolutions in poverty alleviation strategies can be divided into two major phases: (i) appropriate technology [8, 14, 44] and (ii) market-based development [14, 18, 20, 23, 26, 42, 52].

The foundations of poverty alleviation through sustainable development were laid in the 1920’s when Mahatma Gandhi encouraged the development of rural technologies to help India’s villages to become more self-reliant [32]. As time progressed, the rich of the world continued to become richer and the poor remained impoverished [19]. In the 1970’s the engineering design community began to formally address the growing problem of poverty [8, 43], and with this the appropriate technology movement became the strategy implemented for the next 15 years [14, 40]. Appropriate technology is described in the literature as the design of technologies that carefully consider the environmental, ethical, cultural, social, and economic aspects of the community it is intended for [8]. Characteristics of this movement include: adherence to a community ownership/maintenance philosophy, implementation of simple
time saving technologies, development of labor-intensive solutions, low capital cost solutions, use of locally available natural resources where possible, and low maintenance cost solutions [8, 14, 44].

In the fifteen years that the appropriate technology movement was the foremost approach to poverty alleviation, some practitioners noticed that the movement failed to provide sustainable results [10, 14, 38, 40, 41]. Identifying the weaknesses of the movement (too much emphasis placed on time and labor saving devices; developed technologies did not create individual opportunities; abandonment of implemented technologies after a short period of time; etc.) [14, 40], adjustments were made, and the market-based development phase in poverty alleviation strategies began [14, 40, 41, 45]. Market-based development addresses the issue of poverty alleviation by “not only designing technologies ‘appropriate’ to the poor, but also designing technologies that people in poor places [can] themselves appropriate and use to advance their own ends” [14]. To create products that can generate more income for the poor requires an understanding of the dynamics of the local culture and product markets [23]. Characteristics of this approach include: focusing on the development of technologies that create business opportunities for individuals, selling products to promote self-sufficiency [20], and establishing permanent, profitable, and locally marketed/advertised supply chains [11, 14].

Since the establishment of the market-based approach, published literature has indicated that poverty alleviation approaches that create individual opportunities [14, 24, 42], encourage self-esteem [14, 23, 42], offer choices [6, 14, 42, 52], and foster self-sufficiency by increasing earning power [14, 18, 24, 52] for the poor have had the most measurable, and sustainable affect on poverty. A fundamental thread in these sustainable methods is their market-based nature, where everyone in the supply chain benefits from the alleviation efforts [14, 18, 24, 52]. Methods that do not do this have simply not been self-sustaining [14, 23, 52]. Recent reports by companies that design income-generating products (such as irrigation pumps and drip irrigation systems for small farm owners) have shown that these products have measurably and sustainably helped over 12 million people escape poverty [1, 14, 40]. These products, priced at roughly $100 (roughly 3 months income) [14], were introduced under a market-centric approach where all in the supply chain benefit.

2.2 Multiobjective Optimization as a Foundation for Reconfiguration

The need for extreme product-affordability changes over time for those living in poverty when income-generating products are used. As more income is generated, other products become more and more affordable. The ability to balance the competing nature of present customer needs for affordability against future needs for higher productivity is fundamental to the presented method. Multiobjective optimization is used as a foundation for finding the needed balance [21, 25, 39, 47, 53]. The set of design options that best balance the design tradeoffs belong to the Pareto frontier, which is the set of all non-dominated optimal solutions. We seek Pareto optimal designs because they represent feasible designs for which the objectives have been improved as much as possible without sacrificing another design objective’s performance by improving another objectives feasible performance [3, 9]. Under the developments of this paper, we ultimately seek conditions for which each instantiation of a reconfigurable product is close to the Pareto frontier.

The following formulation yields the Pareto frontier for any generic design case; \( \min \left\{ \mu_1(x, p), \mu_2(x, p), \ldots, \mu_n(x, p) \right\} \). Subject to \( g(x, p) \leq 0, h(x, p) = 0 \), and \( x_l \leq x \leq x_u \). Where \( \mu_i \) denotes the \( i \)-th generic design objective; \( g \) and \( h \) represent inequality and equality constraints, \( x \) is a vector of design variables; and \( p \) is a vector of design parameters.

For multiobjective optimization approaches, the decision of which Pareto optimal solution to select comes through the inclusion of objective function parameters, and sometimes constraints that capture customer needs or preferences for a single instance in time [29, 30]. The method presented in this paper builds on principles of Pareto optimality, by allowing the product to move from one location on the Pareto frontier to another through the addition of modules and through reconfiguration. This movement is intended to occur over time as customer preference and needs change.

2.3 Product Modularity and Reconfiguration

As described in detail in the next section, a desirable product (one residing near the Pareto frontier) may be adapted to become a different desirable product (another one residing near the Pareto frontier) through the addition of modules and reconfiguration. This calls for the strategic design of platforms and modules that make products progressively expandable [15, 27, 28]. Previous work in the areas of product family, modular product, and flexible system design serve as a starting point, and as a source for design principles. [12, 15, 16, 28, 46, 50, 55–57]

A module-based product family is a group of related products derived from independent functional or geometric units [2, 37, 48] that differ through the addition or subtraction of modules [37, 48, 56]. In the literature three basic types of modularity are identified: (i) Slot-modular architecture, (ii) Bus-modular architecture, and (iii) Sectional-modular architecture [35, 36, 48]. A slot-modular architecture provides each module with a unique interface in order to eliminate improper assembly [35, 36, 48]. Bus-modular architecture implements interfacing that is the same for all modules, thus making the platform design behave as a common connection platform for all modules [48]. Sectional-
modular architecture is similar to bus-modular in that all modules contain the same interface, but in this architecture no single element is identified as the platform to which all modules attach [35, 48]. In the present paper we use these definitions of modular architectures to specify the approach needed to develop module designs according the a desired architecture type.

Although similar to module-based products, flexible products focus on implementing design components of adaptability (design variable values change due to predictable functional need changes) and design robustness (setting of fixed design variable values to account for unpredictable environmental and other operational condition changes) [13, 33, 34]. Otherwise stated, the major difference between the two strategies is that flexible products adjust an existing design, while module-based products add or subtract modules to satisfy various needs [12, 13, 16, 33, 34, 46]. In the present paper we use the concepts of adjustability (reconfigurability) in flexible product design to further enhance the ability of a modular product to adjust to new and changing needs over time.

The coupling of product modularity design principles and multiobjective optimization also plays an important role in the developments presented in this paper. Recent developments in the literature show that a desirable product family can be identified from among the designs comprising the Pareto frontier [55, 57]. These previous developments evaluate and select product family members from among the set of Pareto designs by considering the design’s unique performance and common features compared to other designs in the product family (a critical part of product family design). In addition, one method of identifying module and platform variables is accomplished through the use of Pareto-filtering methods that explore the effects of each variable on the objective space performance [55, 57]. Building on these approaches as a foundation, in the proposed method a selection criteria based on known changes in customer needs over time is added to the evaluation – to ensure that a progression from one design on the Pareto front to another can be done through the addition of a module.

3. A New Strategy for Designing Reconfigurable Products

In this section, we present an optimization-based design strategy that can be used to identify a set of reconfigurable product instantiations realizable through the addition of modules. Moreover, the strategy also includes a process for designing the required modules. An illustration of the strategy is provided as a flow diagram in Fig. 1. The purpose of this section is to provide details and mathematical formulations for the steps of this process. Each step is now discussed.

3.1 Characterize the Multiobjective Design Space
The method first seeks to characterize the design space by identifying feasible and optimal values for the design objectives. These are identified by formulating and executing a Multiobjective Optimization Problem (MOP) of the form described in Sec 2.2.2. The result is a set of optimal solutions – those belonging to the Pareto frontier. This step is a fundamental part of the process as it is the step that allows the designer to understand the range of optimal performance possibilities. From a graphical perspective, Step A identifies a Pareto frontier, shown as a bold line in Fig. 2 for a problem where both objectives are minimized.

3.2 Define Anticipated Regions of Interest
This step of the process allows the designer to choose regions around portions of the Pareto frontier within which product reconfigurations will be sought. In that way, each region is assumed to contain candidates for handling a different set of customer needs that are anticipated to occur over time. Furthermore,
it is within these regions that designs, well-suited for modularity, are sought.

![Diagram](image-url)

Figure 2. Representation of the construction of Anticipated Regions of Interest for known changes in customer needs for three intervals. For the case shown, the anticipated regions of interest provide inequality constraints for both $\mu_1$ and $\mu_2$.

The fundamental concept of this step is illustrated in Figure 2. For each anticipated region of interest, a new MOP, with a reduced design space based on the region of interest (shaded), is defined by additional objective constraints. For example, for the region of interest presented in Figure 2, the $j$-th design objective ($\mu_1$) is constrained within the $i$-th region of interest by $\mu_{j,l}^{(i)} \leq \mu_{j,r}^{(i)}$, where $\mu_{j,l}^{(i)}$ and $\mu_{j,r}^{(i)}$ are prescribed. The result is a bounded MOP for searching the design space only within the anticipated region of interest.

### 3.3 Select Platform Variables

The third step of the process considers the Pareto designs within the regions of interest to identify the variables that are best suited as platform variables ($x_p$). This may be accomplished through the use of Pareto-filtering methods as described in Section 2.2.3 or any other suitable method. As a tradeoff to modularity, a shift in the Pareto frontier is likely to occur. Thus, once the platform variables are selected, steps A and B may be repeated, as shown in Figure 1, to insure that the resulting shift in the Pareto frontier is acceptable to the designer. In the context of poverty alleviation, this step selects the best possible platform variables that will allow for extreme affordability and subsequent addition of modules that satisfy desired objectives, particularly those related to affordability.

### 3.4 Select the Optimal Design Within Each Region of Interest

The purpose of this step is to develop and execute an optimization problem for selecting the best design in each anticipated region of interest and identify the accompanying design variable values. The resulting optimal design set ($D$) containing all variable values is obtained through the following MOP formulation:

**Problem 1a: MOP Formulation for Optimal Adaptive Product Identification**

$$D := \{ (x_{p,1}, x_{p,2}, \ldots, x_{p,n_p}, x_{a,1}^{(i)}, x_{a,2}^{(i)}, \ldots, x_{a,n_a}^{(i)}) | \forall i \in \{1, 2, \ldots, n_d\} \} \quad (1)$$

$x_{p}, x_{a}^{(i)}$ defined by:

$$\min_{x_{p}, x_{a}^{(i)}} \frac{1}{n_d} \sum_{n=0}^{n_d} f^{(i)} \quad (2)$$

subject to:

$$g_q^{(i)}(x_{p}, x_{a}^{(i)}, p^{(i)}) \leq 0 \forall q \in \{1, \ldots, n_g^{(i)}\} \quad (4)$$

$$h_k^{(i)}(x_{a}^{(i)}, p^{(i)}) = 0 \forall k \in \{1, \ldots, n_h^{(i)}\} \quad (5)$$

$$x_{a,j,l}^{(i)} \leq x_{a,j,r}^{(i)} \forall j \in \{1, \ldots, n_a\} \quad (6)$$

$$x_{p,r,l}^{(i)} \leq x_{p,r,u}^{(i)} \forall r \in \{1, \ldots, n_p\} \quad (7)$$

$$\mu_{y}^{(i)} \preceq \mu_{y}^{(i)} \forall y \in \{1, \ldots, n_y^{(i)}\} \quad (8)$$

where the adjustable variables ($x_{a}$) represent all non-platform design variables (variables that are either scaled or discretely adjusted); $m$ is a compromise programming power [5]; $w_{1}^{(i)}, w_{2}^{(i)}, \ldots, w_{n_{d}}^{(i)}$ are weights associated with the local preference within each region of interest; the set $D$ now represents the set of all design variable values of $x_{a}$ and $x_{p}$ obtained through the evaluation of the MOP; and the superscript $(i)$ on $p$, $g$, and $h$
indicate the possibility that parameters and constraints are different (non-constant) for each design in the set \( D \). It is important to note that Problem 1a will result in a single solution within each region of interest which are selected by finding the set of \( n_{d} \) designs with the best average aggregate objective function values (see Equation refP1:obj1). It is also important to note that because the platform variables are already identified and used in this step, the resulting designs already share commonality which is the basis for reconfiguration through modularity.

Although not the focus of this paper, it is recognized that the weights indicated in Equation refP1:obj2 will need to be determined for actual design scenarios. For the method presented herein it is assumed that this information is known, but potential methods for determining these weights could include the use of market surveys, focus groups, or other suitable methods.

### 3.5 Develop Modules That Move From One Region of Interest to Another

By this step in the process, the set \( D \) now contains all variable values that can be used to develop the module designs. Developing these designs is now a matter of constrained module design – modules are designed in a manner that constrains them to provide a specified progression in product performance when added to a specific embodiment of the product while only using the variable values from set \( D \). To complete this final step of the method and obtain the module designs requires the following: (i) Select a modular architecture type, (ii) Identify the platform design and module interfaces, (iii) Determine the desired number of modules and modular progression, and (iv) Identify and calculate the values of module design variables. Each of these four parts is briefly discussed.

**Select a modular architecture type:** Of the three types of modularity identified in the literature (see Section 2.2.3), Slot-modular architecture and Bus-modular architecture are best suited for implementation in the present method due to the use of platform designs. The decision of which architecture type to be used depends on the desired functionality of the product and modules as a whole.

**Identify the product platform design and module interfaces:** Prior to the identification of modules, one of the designs in set \( D \) must be identified as the product platform design. In order to facilitate adaptability, the platform design is generally identified as the design contained in \( D \) with the most commonality. In addition, the module interfaces must be specified according to the modular architecture type selected previously and the information provided in \( \delta \), and any other related interfacing design activities must be performed.

**Determine the desired number of modules and the modular progression:** With a knowledge of the modular architecture type that is desired, it is now possible to determine the number of modules \( (n_m) \) that are desired. The identification of \( n_m \) requires a knowledge of the manner in which the product is intended to expand – through module addition and/or reconfiguration. Using the integer value of \( n_m \) that is desired, an \( n_m \)-by-2 matrix \( (\delta) \) is constructed. The purpose of the matrix \( \delta \) is to dictate the desired progression from one design contained in set \( D \) to another, where the first entry of the \( \delta \) matrix \( (\delta_{1,1}) \) is generally the platform design from set \( D \) identified in the previous section. A generic construction of a \( \delta \) matrix is presented as follows:

\[
\delta = \begin{bmatrix}
\alpha_1 & \beta_1 \\
\alpha_2 & \beta_2 \\
\vdots & \vdots \\
\alpha_{n_m} & \beta_{n_m}
\end{bmatrix}
\]  

where \( \alpha \) and \( \beta \) respectively refer to the starting and the ending designs of the set \( D \) that each module is bridging. This information is used in the final step to refer to the values of \( x_a \) needed to design each module.

**Identify and calculate the values of module design variables:** The identification of module designs first requires that those variables that are best suited to characterize the modules be identified – module variables \( (x_m) \). This identification of module variables can be performed using the same methods described previously for identifying platform variables. In cases where a designer knows which variables are best suited as module variables for manufacturing a modular product, the providing of that information for the module design routine presented below is all that is required. Using this information, and the information provided in \( \delta \) and \( D \), a generic constrained module design routine is presented below.

**Problem 1b: Optimization Problem Formulation for Constrained Module Design**

\[
D_m := \{ (x_{p,1}, x_{p,2}, \ldots, x_{p,n_p}, x_{m,1}^{(i)}, x_{m,2}^{(i)}, \ldots, x_{m,n_m}^{(i)}) | \forall i \in \{1, 2, \ldots, n_m\} \}
\]  

\( x_m \) is defined by:

\[
\min_{x_m} f^{(i)} = \left( p^{(i)} - \bar{p}^{(i)} \right)^2
\]
where:

\[ \alpha = \delta_{i1} \]  
\[ \beta = \delta_{i2} \]  
\[ \tilde{p}^{(i)} = p^{(\alpha)} + \Delta p^{(i)} \]  

defined by:

\[ p^{(\alpha)} = \left( \mu_1(x_{a1}^{(\alpha)}, x_p, p^{(\alpha)}), \mu_2(x_{a2}^{(\alpha)}, x_p, p^{(\alpha)}) \right) \]  
\[ \cdots, \mu_{n\mu}(x_{a\mu}^{(\alpha)}, x_p, p^{(\alpha)}) \]  
\[ (n_{\mu} \geq 2) \]  
\[ p^{(\beta)} = \left( \mu_1(x_{a1}^{(\beta)}, x_p, p^{(\beta)}), \mu_2(x_{a2}^{(\beta)}, x_p, p^{(\beta)}) \right) \]  
\[ \cdots, \mu_{n\mu}(x_{a\mu}^{(\beta)}, x_p, p^{(\beta)}) \]  
\[ (n_{\mu} \geq 2) \]  
\[ \Delta p^{(i)} = \left( \Delta \mu_1(x_{m1}^{(i)}, x_p, \tilde{p}^{(i)}), \Delta \mu_2(x_{m2}^{(i)}, x_p, \tilde{p}^{(i)}) \right) \]  
\[ \cdots, \Delta \mu_{n\mu}(x_{m\mu}^{(i)}, x_p, \tilde{p}^{(i)}) \]  
\[ (n_{\mu} \geq 2) \]

where \( D_m \) is the set of values and variables of \( x_p \) and \( x_m \) for each module design; \( p^{(\alpha)} \) and \( p^{(\beta)} \) characterize the objective space performance of the base (\( \alpha \)) and target (\( \beta \)) designs; \( \tilde{p}^{(i)} \) represents the objective space performance of design \( \alpha \) when used in conjunction with the \( i \)-th module; \( \Delta p^{(i)} \) represents the change in objective space performance from design \( \alpha \) to \( \tilde{p}^{(i)} \); and \( x_m \) represents the value(s) and variable(s) that characterize \( \Delta P \).

Upon completion of Step E, the set \( D_m \) contains the necessary variables and values needed to define the module designs. As a result, a product capable of adapting to changes in customer needs over time through the addition of modules is achieved. In addition, each reconfigured iteration of the product obtained through the addition of modules to the identified platform design is required to provide the optimal performance according to the objectives provided in Problem 1a (see Section 3.3.4). As presented in the next section, this 5-Step method is used to design a reconfigurable, manually powered, irrigation pump for developing nations. The process results in a design that strategically moves along the Pareto frontier that starts at the lowest cost pump; and transitions to a large capacity pump by module addition.

4 Example: Manual Irrigation Pump

The example that follows shows the application of the methodology present in the previous section in the creation of a modular, manually operated irrigation pump. In addition, the example demonstrates the ability of the method to provide a modular income generating product that allows the purchaser to make a 3 stage investment to purchase a product that would otherwise be considered unaffordable.

Drive for the development of this modular pump is best illustrated though the plot provided in Figure 3. This figure provides a comparison of various non-modular irrigation pumps that are sold on the market today based on their sales price, \( S \) (horizontal axis), and potential water flow rate, \( Q \) (vertical axis). From Figure 3 it is seen that products are available to satisfy a range of current views of what is considered affordable, but none of these products are capable of expanding as an individual’s view of affordability changes due to increases in income potential (i.e. a Hip Pump cannot become a Super MoneyMaker). In short, drive for the development of a modular irrigation pump stems from the need to reduce the high perceived and actual financial risks involved with purchasing traditional irrigation pumps, [14,20,52] while still providing the needed pump performance that will increase the purchasers income. To overcome this disparity, analytical models of the fluid and financial aspects of an irrigation pump are developed (see Equations 29-45 of Problem 2a below) to predict the behavior of a pump design based on a set of model inputs.

![Graphical comparison of three non-modular water pumps that are currently sold on the market. The horizontal axis represents the sales price (S in US dollars), and the vertical axis represents the potential water flow rate (Q) in liters per second.](image)

Assumptions made in the development of the analytical financial and fluid models presented in Equations 29-45 of Prob-
lem 2a are as follows: (1) Water flow will always be turbulent. (2) The corresponding friction coefficients for flow in the pump cylinder \((f_c)\) and pipes \((f_p)\) are approximated by the average friction value for the expected flow speeds and the ratios of the surface roughness \((\varepsilon)\) to pipe/cylinder diameter \((d_p\) and \(d_c\) respectively). (3) The force transmission efficiency of the pump \((\eta)\) is assumed to be constant and equal to 80%. (4) During leg operation of the pump, the force applied by the pump operator \((F_l)\) is assumed to be constant and equal to 889.6 N. (5) During hand operation of the pump, the force applied by the pump operator \((F_h)\) is assumed to be constant and equal to 70% of \(F_l\) (622.72 N). (6) The design variable best suited for manufacturing a reconfigurable, modular irrigation pump is the piston cylinder diameter \((d_c)\).

Using the information provided in Figure 3 and the knowledge of the assumptions made in developing the analytical models, the anticipated regions of interest within the design space are developed based upon the values of \(Q\) and \(S\) for the three products shown. The limits describing the three anticipated regions of interest (ARI) within the design space of \(Q\) and \(S\) are provided in Table 1. Values of the limits for each region are based on the performance of the MoneyMaker Hip Pump (ARI 1), MoneyMaker Plus (ARI 2), and the Super MoneyMaker (ARI 3) in terms of \(S\) and \(Q\) [7, 14, 22, 54].

<table>
<thead>
<tr>
<th>ARI #</th>
<th>(S_{\text{min}}) ($)</th>
<th>(S_{\text{max}}) ($)</th>
<th>(Q_{\text{min}}) (L/s)</th>
<th>(Q_{\text{max}}) (L/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>40</td>
<td>0.25</td>
<td>0.80</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>70</td>
<td>0.90</td>
<td>1.15</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>110</td>
<td>1.35</td>
<td>2.00</td>
</tr>
</tbody>
</table>

\[ x_p, x_a \text{ defined by:} \]

\[
\min_{x_p, x_a} \frac{1}{3} \sum_{n=0}^{3} f_n^{(i)}
\]

where:

\[
f_n^{(i)} = -Q(x_a^{(i)}, x_p, p) + 1 \cdot S(x_a^{(i)}, x_p, p)
\]

\[
x_p = \{d_c\}
\]

\[
x_a^{(i)} = \left\{ l_c^{(i)}, l_0^{(i)}, n_c^{(i)} \right\}
\]

\[
p = \left\{ g \rho_w \varepsilon \eta F_l F_h l_s d_p l_p, l_p, l_p, l_p \right\}
\]

subject to:

\[
Q_{\text{min}}^{(i)} \leq Q^{(i)} \leq Q_{\text{max}}^{(i)}
\]

\[
S_{\text{min}}^{(i)} \leq S^{(i)} \leq S_{\text{max}}^{(i)}
\]

\[
0.5 \leq t_s^{(i)} \leq 6.0
\]

with \(d_c^{(i)}\) and \(n_c^{(i)}\) assuming discrete values according to:

\[
d_c^{(i)} = \left\{ 0.0525, 0.0627, 0.0779, \ldots, 0.1023, 0.1541, 0.2027 \right\}
\]

\[
n_c^{(i)} = \begin{cases} 1, & i = 1 \\ 2, & i = 2 \\ 4, & i = 3 \end{cases}
\]

where:

\[
Q^{(i)} = \begin{cases} 0.5 \cdot \hat{Q}^{(i)}, & n_c^{(i)} = 1 \\ \hat{Q}^{(i)}, & \text{otherwise} \end{cases}
\]

\[
S^{(i)} = \left( \sum_{j=0}^{p_{\text{comp}}} C_j(x_a^{(i)}, x_p, p) \right) \cdot (1 + M_m + M_d + M_e)
\]

\[ D := \{(x_p, x_a^{(i)}) \mid \forall i \in \{1, 2, 3\}\} \]
with supporting equations:

\[
h_{c}^{(i)} = l_s \cdot \left( \frac{d_c^{(i)}}{l_o^{(i)}} \right), \quad (31)
\]

\[
A_p = \pi \cdot \frac{(d_p)^2}{4}, \quad (32)
\]

\[
A_c^{(i)} = \pi \cdot \frac{(d_c^{(i)})^2}{4}, \quad (33)
\]

\[
m_w^{(i)} = \begin{cases} 
  \rho \left( A_c^{(i)} \cdot h_{c}^{(i)} + A_p \cdot l_{p,in} \right), & n_c^{(i)} = 1 \\
  \rho \left( A_c^{(i)} \cdot h_{c}^{(i)} + A_p \cdot \left( l_{p,in} + l_{p,out} \right) \right), & \text{otherwise}
\end{cases}, \quad (34)
\]

\[
h_{L,major}^{(i)} = \begin{cases} 
  \frac{f_{c} \cdot h_{c}^{(i)} \cdot d_{p}^2}{(d_c^{(i)})^2} + \frac{f_p \cdot l_{p,in}}{d_p}, & n_c^{(i)} = 1 \\
  0.5 \cdot c_{s,c} \cdot f_{c} \cdot d_{p}^2 + \frac{f_p \cdot \left( l_{p,in} + l_{p,out} \right)}{d_p}, & \text{otherwise}
\end{cases}, \quad (35)
\]

\[
h_{L,minor}^{(i)} = \begin{cases} 
  K_{L,1} + K_{L,2} \cdot s_{c} \cdot d_{c}^{(i)} (K_{L,1} + K_{L,2} \cdot s_{c}^{(i)}), & n_c^{(i)} = 1 \\
  K_{L,1} + \frac{K_{L,2} \cdot s_{c}}{2}, & \text{otherwise}
\end{cases}, \quad (36)
\]

\[
K_{L,1} = 0.5, \quad (37)
\]

\[
K_{L,2}^{(i)} = 0.45 - 0.625 \cdot \left( \frac{d_p}{d_c^{(i)}} \right), \quad (38)
\]

\[
K_{L,3}^{(i)} = \left( 1 - \frac{d_f}{d_c^{(i)}} \right)^2, \quad (39)
\]

\[
\lambda^{(i)} = 1 + h_{L,major}^{(i)} + h_{L,minor}^{(i)}, \quad (40)
\]

\[
F_{in}^{(i)} = \eta \cdot F_h = \left( \frac{n_c^{(i)}}{n_c} \right), \quad n_c^{(i)} = 1
\]

\[
\eta \cdot F_l = \left( \frac{n_c^{(i)}}{n_c} \right), \quad \text{otherwise}
\]

\[
V_p^{(i)} = \begin{cases} 
  \frac{2 \cdot \frac{l_{p,in} \cdot h_{c}^{(i)}}{\lambda^{(i)} \cdot m_w} + \frac{z_{in}}{\lambda^{(i)}} \cdot z_{in}}{2}, & n_c^{(i)} = 1 \\
  \frac{2 \cdot \frac{l_{p,in} \cdot h_{c}^{(i)}}{\lambda^{(i)} \cdot m_w} + \frac{z_{in} - z_{out}}{\lambda^{(i)}}, (z_{in} - z_{out})}{\lambda^{(i)}}, & \text{otherwise}
\end{cases}, \quad (42)
\]

\[
V_c^{(i)} = \eta \cdot V_p^{(i)} \cdot \left( \frac{d_p}{d_c^{(i)}} \right)^2, \quad (43)
\]

\[
l_s^{(i)} = h_c^{(i)} / V_c^{(i)}, \quad (44)
\]

\[
\hat{Q}^{(i)} = 1000 \cdot V_p^{(i)} \cdot A_p, \quad (45)
\]

where \( l_s \) is the distance from the pivot to the operator (m); \( l_c \) is the distance from the pivot to the pump cylinder (m); \( n_a \) is the number of cylinders; \( h_c \) is the distance traveled by the cylinder piston head (m); \( l_o \) is the length of the operator stroke (m); \( l_{p,in} \) is the length of the inlet pipe (m); \( l_{p,in} \) is the length of the outlet pipe (m); \( z_{in} \) is the vertical distance from the pump to the water source (m); \( z_{out} \) is the vertical distance from the pump to the pipe outlet (m); \( m_w \) is the mass of the water in the entire pumping system (kg); \( A_p \) is the cross sectional flow area of the pipe (m²); \( A_c \) is the cross sectional flow area of the cylinder (m²); \( h_{L,major} \) is a partial form of the major head loss in the pump system; \( h_{L,minor} \) is a partial form of the minor head loss in the pump system; \( K_{L,1} \) is the entrance loss coefficient; \( F_h \) is the force applied at the cylinder head (N); \( V_p \) is the average flow velocity in the pipes (m/s); \( V_c \) is the average flow velocity in the cylinders (m/s); \( Q \) is the flow rate in the system assuming a constant flow (L/s); \( t_c \) is the stroke time of the operator (s); \( C_j \) is the manufacturing cost of the \( j \)-th component of the pump ($) \( M_m \) is the manufacturing mark-up (%); \( M_d \) is the distributor mark-up (%); and \( M_s \) is the sales mark-up (%). The selected objectives for this problem are to maximize the predicted flow rate \((Q(i))\) and minimize the predicted sales price \((S(i))\) (see Equation 20).

It should be noted that, as was previously presented in Equations 27 and 28 for the possible variable values of \( d_c \) and \( n_c \) the variables contained within \( x_p \) and \( x_a \) are defined as discrete variables. The ranges and value steps of \( l_o \) and \( l_p \) are given in Table 2. In addition, the values of the parameters contained in \( p \) are provided in Table 3.

Values for the variables \( l_{p,in} \), \( l_{p,out} \), \( z_{in} \), and \( z_{out} \) presented in Table 3 indicate that the pump is being designed to pull water from a water source that is three meters below the pump and then discharge it into a ditch or furrow one meter below the pump. The equations used to evaluate the pump’s performance with respect to the objective \( Q \) (see Equation 29) are derived from the Energy Equation of the First Law of Thermodynamics presented in Munson et al (see Equations 35–41, 42, and 45 above) [31].

Evaluation of Problem 2a was performed using a Genetic Algorithm, and complete results indicating the variable and objective values of each design are presented in Table 4. It should be noted that the pump designs obtained through the evaluation of Problem 2a do not represent platform and module designs. Instead, they represent the non-modal product designs chosen by the method to be the best suited for conversion into platform and module designs while simultaneously providing the best average objective performance as defined by Equations 19 and 20.

<table>
<thead>
<tr>
<th>Variable (units)</th>
<th>Lower Limit</th>
<th>Upper Limit</th>
<th>Step Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_o ) (m)</td>
<td>0.3</td>
<td>2.0</td>
<td>0.01</td>
</tr>
<tr>
<td>( l_p ) (m)</td>
<td>0.2</td>
<td>0.4</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Table 3. Values of the parameters of p needed in Problem 2a to obtain the i-th design of set D.

<table>
<thead>
<tr>
<th>Variable (units)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>g (m/s²)</td>
<td>9.80665</td>
</tr>
<tr>
<td>ρw (kg/m³)</td>
<td>1000</td>
</tr>
<tr>
<td>ε (m)</td>
<td>0.0015</td>
</tr>
<tr>
<td>η (%)</td>
<td>80</td>
</tr>
<tr>
<td>F₁ (N)</td>
<td>889.6</td>
</tr>
<tr>
<td>Fh (N)</td>
<td>622.72</td>
</tr>
<tr>
<td>ls (m)</td>
<td>0.3048</td>
</tr>
<tr>
<td>dp (m)</td>
<td>0.0254</td>
</tr>
<tr>
<td>lp,in (m)</td>
<td>3.0</td>
</tr>
<tr>
<td>lp,out (m)</td>
<td>1.0</td>
</tr>
<tr>
<td>zn (m)</td>
<td>-2.0</td>
</tr>
<tr>
<td>zout (m)</td>
<td>-1.0</td>
</tr>
<tr>
<td>fc</td>
<td>0.05</td>
</tr>
<tr>
<td>fp</td>
<td>0.075</td>
</tr>
</tbody>
</table>

Table 4. Variable and objective values obtained through evaluation of Problem 2a for the i-th design of set D.

<table>
<thead>
<tr>
<th>i</th>
<th>dc (m)</th>
<th>la (m)</th>
<th>lc (m)</th>
<th>nc (m)</th>
<th>Q (L/s)</th>
<th>S ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1541</td>
<td>1.36</td>
<td>0.20</td>
<td>1</td>
<td>0.7216</td>
<td>37.1809</td>
</tr>
<tr>
<td>2</td>
<td>0.1541</td>
<td>0.53</td>
<td>0.25</td>
<td>2</td>
<td>1.1392</td>
<td>56.9606</td>
</tr>
<tr>
<td>3</td>
<td>0.1541</td>
<td>0.95</td>
<td>0.20</td>
<td>4</td>
<td>1.3847</td>
<td>93.7188</td>
</tr>
</tbody>
</table>

Through the evaluation of Problem 2a above, the set D now contains all variable values needed to develop the module designs (see Table 4). Prior to developing the module designs, information on the type, number, and desired progression of modules that are to be used to obtain the objective space performance of the Pareto designs contained within set D is needed. In order to limit the potential of operator assembly errors, a slot modular approach is selected. Examination of the nature of the xₚ variables reveals that the differences in the variable values for each design in the set D is geometric, and therefore the design with the most commonality is the design with the smallest value of nc (D[1]). Using this information, the desired number of modules to be developed (nₘ) is chosen to be two, and the δ matrix is constructed as follows:

\[
\delta = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}
\]  (46)

Formulation of a constrained module design routine of the form presented in the Section 3 is now presented.

Problem 2b: Irrigation Pump Example – Constrained Module Design

\[
D_m := \{(x_p, x_m^{(i)}) \mid \forall i \in \{1, 2\}\}
\]  (47)

xₘ is defined by:

\[
\min_{x_m} J^{(i)} = (p^{(\beta)} - \bar{p}^{(i)})^2
\]  (48)

defined by:

\[
\bar{p}^{(i)} = p^{(\alpha)} + \Delta p^{(i)}
\]  (49)

\[
p^{(\alpha)} = (Q^{(\alpha)}, S^{(\alpha)})
\]  (50)

\[
p^{(\beta)} = (Q^{(\beta)}, S^{(\beta)})
\]  (51)

\[
\Delta p^{(i)} = (\Delta Q^{(i)}, \Delta S^{(i)})
\]  (52)

where:

\[
x_m^{(i)} = \{l_c^{(i)}, n_c^{(i)}\}
\]  (53)

\[
x_p = \{d_c\}
\]  (54)

\[
\alpha = \delta_{i,1}
\]  (55)

\[
\beta = \delta_{i,2}
\]  (56)

\[
\Delta Q^{(i)} = Q \left( d_c, (l_c^{(\alpha)} + l_c^{(i)}), l_c, (n_c^{(\alpha)} + n_c^{(i)}), p^{(\alpha)} \right)
\]

\[+ Q \left( d_c, l_c^{(\alpha)}, l_c, n_c^{(\alpha)}, p^{(\alpha)} \right)
\]  (57)
\[ \Delta S^{(i)} = S \left( d_c, (i_o^{(\alpha)} + l_t^{(i)}), l_c, (n_c^{(\alpha)} + \hat{n}_c^{(i)}), p^{(\beta)} \right) - S \left( d_c, i_o^{(\alpha)}, l_c, n_c^{(\alpha)}, p^{(\alpha)} \right) \]  

(58)

\[ l_t^{(i)} = \begin{cases} 
    l_{t_o}^{(\beta)}, & i = 1 \\
    l_{i_o}^{(\beta)} - l_{i_o}^{(\alpha)}, & \text{otherwise}
\end{cases} \]

(59)

where the values and variables of \( x_p \) are the same as those obtained through the evaluation of Problem 2a; \( l_t \) is the length of the \( i \)-th treadle extension (m); and \( \hat{n}_c \) is the number of cylinders added by the \( i \)-th module.

The variable values of the Platform Design and the results of the evaluation of Problem 2b are presented in Tables 5 and 6 respectively.

In order to visually validate that the method has provided the optimal set of platform and module designs, Figure 4 is provided. Contained in this figure is a collection of plots that summarize the progression of the method as implemented in this section. Figure 4(a) provides an approximation of the regions of interest within the feasible design space assuming that all variables are allowed to vary (i.e. no platform variables are selected). Figure 4(b) shows the feasible design space within the regions of interest after selecting \( d_c \) as the platform design variable as described in Problem 2a. Figure 4(c) shows the feasible regions from Figure 4(b) and the Pareto and near Pareto designs produced by the Genetic Algorithm. From this plot it is observed that, based on the objectives to minimize \( S \) and maximize \( Q \), the designs are clustered toward the optimal boundary of the regions of the feasible design space. Finally, Figure 4(d) provides the same plot as Figure 4(a) provides an approximation of the regions of interest within the feasible design space assuming that all variables are allowed to vary (i.e. no platform variables are selected). Figure 4(b) shows the feasible design space within the regions of interest after selecting \( d_c \) as the platform design variable as described in Problem 2a. Figure 4(c) shows the feasible regions from Figure 4(b) and the Pareto and near Pareto designs produced by the Genetic Algorithm. From this plot it is observed that, based on the objectives to minimize \( S \) and maximize \( Q \), the designs are clustered toward the optimal boundary of the regions of the feasible design space. Finally, Figure 4(d) provides the same plot as...
shown in Figure 4(c), except that the only designs shown are the benchmark products used to develop the objective constraints of Problem 2a (indicated by the symbol “□”), the designs selected by the method according to Equations 20 and 21 (indicated by the symbol “◦”), and the platform and module designs obtained through Problem 2b (indicated by the symbol “×”). From this series of plots it is seen that the method is capable of selecting a set of designs that provides the best average objective performance as defined by Equations 19 and 20 as well as providing the platform and module designs that allow the product to provide the desired reconfigurability that was previously unattainable.

Having verified that the method has provided the optimal set of platform and module designs, preliminary developments of 3D solid CAD models of the irrigation pump are developed. Renderings of these models are provided in Figure 5. Inspection of Figure 5 shows that the intended progression of the pump, as identified through Problem 2b, is to begin by providing a platform pump design that is hand operated and only provides one cylinder (see Figure 5(a)). The first module requires reconfiguration of the pump by attaching an additional cylinder, two new levers (treadles) that the user can step on, and reconfiguring the handle and stabilizer plate to provide balance while operating the pump (see Figure 5(b)). The second module requires additional reconfiguration of the pump through the attaching of two additional cylinders, extensions for the treadles, and the necessary hardware to ensure proper pump function (see Figure 5(c)). From these illustrations it is seen that the goal of providing an income generating product that allows for a three stage investment to incrementally increase the performance of the product is realized. In addition, each reconfiguration of the product achieved through the addition of a module accounts for the changes in what is considered affordable due to increases in income potential.

5 Concluding Remarks

A multiobjective optimization design method that accounts for relationship between affordability and income potential has been developed and demonstrated in this paper. The presented approach involves the strategic use of a series of optimization formulations that ultimately result in modular products that can be reconfigured to adapt to changing consumer needs over time.
by moving from one Pareto design to another through the addition of a module. Through the example of the irrigation pump presented in this paper, the ability of the method to provide product designs that adapt to changes in customer needs through reconfiguration and the addition of modules is shown.

From the formulation of Problem 1a provided in Section 3.3.4, it should be noted that Equation 3 makes use of both weighted sum and compromised programming methods for identifying Pareto-optimal designs. The simultaneous implementation of both of these approaches is intended to overcome challenges in searching concave objective spaces, but still embodies limitations in searching/representing the objective space through this technique. Accordingly, future work on this topic will include the exploration and characterization of the effects of alternative formulations of Equations 2-3.

Other topics of future work related to the work presented in this paper include the exploration of methods for determining the modular architecture feasibility, identification of the methods and challenges of marketing income-generating products, exploration of additional design problems of increasing complexity to explore the limitations of the method, and exploration of the application of the method in other areas of affordability (i.e. aiding companies in enhancing their manufacturing capabilities as they grow without needing to invest in brand new machines).

6 Acknowledgments

We would like to recognize the National Science Foundation Grant CMMI-CAREER-0954580 for funding this research.

REFERENCES

[10] W. R. Easterly. The White Mans Burden: Why the Wests efforts to aid the rest have done so much ill and so little good. Penguin Group, 2006. 3
2009. 8


