Product Selection Using Optimization

by

Xiaoxia Fan

A research paper
presented to the University of Waterloo
in partial fulfillment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Prof. Thomas F. Coleman

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

Product selection is an important problem in many industries. In this essay, we present an optimization model to identify the optimal product portfolio. This model can be solved efficiently with simulated annealing. With an application example, we demonstrate that our model is highly effective and potentially brings significant profit gains. In particular, our model gives a benchmark for the optimized portfolio size. We also propose an alternative approach, Weighted Function, for the binary integer optimization problem.

Acknowledgements

I would like to thank my supervisor, Professor Thomas Coleman for his support and guidance. I would also like to thank Conrad Coleman for his help and supervision to make this project possible. Last but not least, I would like to thank my family and friends for their encouragement and motivation.

List of Figures

Contents

Chapter 1

Introduction

Global competition has forced manufacturing industries to adapt to the fast paced change of consumer demand on a wide variety of products. To keep competitive in the market, one crucial factor is to select optimal product portfolios among all existing products which could satisfy the market demand. Another crucial factor is to design the new products that could satisfy customer's needs and gain the lead in product efficiency and effectiveness.

There are many factors to consider when designing a product, such as the following:

- **Operations** What are the needs of the proposed new product and how do they match the existing resources? Will the company need new facilities and equipment? Does the company have the labour skills to make the product? Can the material for production be readily obtained?
- **Marketing** What is the potential size of the market for the proposed

new product? How much effort will be needed to develop a market for the product and what is the long-term product potential?

- **Finance** The production of a new product is a financial investment like any other. What is the proposed new product's financial potential, cost, and return on investment?

There is much literature about product selection and new product design. Peter T. Ward et al. studied the manufacturers' competitive priorities in [2]. R. Grussenmeyer and T. Blecker [3] researched the relevance of different complexity management requirements for new product development. DeVries in [4] studied a dynamic model for product strategy selection. Su and Pearn studied production selection for newsboy-type products with normal demands and unequal costs in [5]. Cao et al. presented a quantitative criterion of model selection for building material with delivery delay in [6]. Tang and Yin in [7] present a model to determine the conditions under which a particular product selection is optimal. The reader is referred to Eliashberg and Steinberg[8], Gaur and Honhon[9], Ho and Tang [10], Kok and Fisher [11] for more reviews.

This project focuses on developing an optimization tool for product selection from existing products as well as designs new products based on the technical evaluation as well as market information. More specifically, we develop an optimization tool which can identify a product portfolio that covers as much demand range as possible while maximizing the market performance. At the same time, we maintain flexibility in the optimization tool to allow adaptation of this method over time.

To develop this optimization tool, we should first answer the following questions:

(1) Which parameters should we use to describe products?

(2) What performance measure should we use?

(3) How should we interpret sales data?

For the first question, we propose to choose fundamental parameters by which all other parameters can be determined. After we answer the first question, the second question becomes more natural. We propose a function that expresses the unique characteristics and features of the company's product using the parameters chosen in first question. For the third question, we propose to use an interpolation method to get a smooth sales surface that could integrate into a performance measurement function. After answering these questions, we are ready to build an optimization model and find the optimal solution from a numerical perspective.

The remainder of the essay is structured as follows: Chapter 2 presents a complete description of the problem and formulates the optimization model. Chapter 3 describes the simulated annealing algorithm that could be used to solve the optimization model discussed in Chapter 2 and presents a numerical example. Chapter 4 presents an alternative approach to the binary optimization problem. Chapter 5 summarizes the essay with further discussion.

Chapter 2

Mathematical Modelling

2.1 Problem Description

In this chapter we will build an optimization model for the product selection problem. First, we determine fundamental input parameters from which all other parameters can be determined. For simplicity, we assume that a product can be defined by two parameters X and Y. We denoted $Range(X)$ and $Range(Y)$ as the range of X and Y, respectively.

A: Subscripts

Several subscripts are used in the model as follows:

The number of products we want to select is denoted by k, $1 \leq k \leq n$, where n is the total number of products.

B: Parameters

n : total number of products available.

X, Y : two fundamental parameters that would be used to define any product.

$\mathbb{F}$: feasible range of products, i.e, $\mathbb{F} = Range(X) \times Range(Y)$.

C: Decision variable

Decision variables are defined as follows:

$$w_i = \begin{cases} 1 & \text{if product } i \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

where $1 \leq i \leq n$.

D: Objectives

A number of objective functions are possible. For example, maximizing manufacturers' profit, minimizing production cost, and maximizing customer coverage. Here, we choose to identify a product portfolio that covers as much of the feasible region as possible while maximizing the market performance.

The feasible region $\mathbb{F}$ is a plane generated by X and Y. For each individual point (x, y) in the feasible region $\mathbb{F}$, the objective function value of a single product is evaluated as the multiplication of product performance and sales density. If there are multiple products defined at a point (x, y), then we choose the product with the maximum performance.

Let $S(x, y)$ be the sales surface (a continuous density function of sales potential), $P_i(x, y)$ be the performance function of product i, $f_i(x, y)$ be a function measuring the feasible region coverage of product i and sales scenario. At each individual point $(x, y) \in \mathbb{F}$, the value of $f_i(x, y)$ is the multiplication of performance of product i, and sales density at point (x, y). That is,

$$f_i(x, y) = S(x, y) \cdot P_i(x, y). \tag{2.1}$$

Covering the entire feasible plane we write the objective function as follows.

$$F(\mathbf{w}) = \int_{\mathbb{F}} \max_{1 \le i \le n} \{f_i(x, y) w_i\} dA. \tag{2.2}$$

where $\mathbf{w} = (w_1, \cdots, w_n)'$ is a vector of decision variables. The result of $\max_{1 \le i \le n} f_i(x, y)$ is the maximum value of $f_i(x, y)$ among all products defined at point (x, y), which eliminate the overlapping of different products.

D: Constraints

While considering the objectives function, several technical constraints can be considered, such as the budget constraint, and short-term minimum return constraint. Here, to simplify our model, we assume that the cost of each product is the same. This will not result in a loss of generality if we consider a product set with similar engineering design from a manufacturing perspective. Therefore, we only consider one constraint, which is the total number of products in our optimal portfolio: k. This could be formulated as:

$$\sum_{i=1}^{n} w_i = k.$$

To summarize, we select k products from a total of n products to form the optimal portfolio by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\textstyle\int_{\mathbb{F}} \max_{1 \le i \le n} \{f_i(x, y) w_i\} dA \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i = k \\ & w_i \in \{0, 1\}, \text{ for } i = 1, \cdots, n. \end{aligned} \tag{2.3}$$

2.2 Performance Measurement

In practice, the performance measurement function in (2.1) is crucial to the optimization model. This function is defined from an engineering design aspect to capture the properties of products over the feasible region. Also, we can define multiple performance measurement functions based on the emphasis on the different perspective of the products, and then take the weighted sum of these functions as the performance function. In this way, we could easily adjust the priority between different perspective of a product as needed by changing their weights in the sum.

2.3 Sales Surface

The mathematical method requires a continuous density surface as an approximation to the real sales surface. We grid the feasible region into small squares, count the number of sold products in each square, and then use these new sample points to interpolate the smoothed sales surface. This method eliminates the sample points that are close together but having extreme difference in sales quantity, which could make some interpolation methods, for example splines, very inaccurate.

There are many interpolation methods available. One example is a spline, which estimates values using a mathematical function that minimizes overall surface curvature, resulting in a smooth surface that passes exactly through the input points. It fits a mathematical function to a specified number of nearest input points while passing through the sample points. Users could choose different interpolation methods for different data.

2.4 Product Location

For product i, we define the best performance point to be the point over the feasible region where the performance function of product i obtains maximum value. That is, we define the *best performance point* as

$$(x_i, y_i) = arg \max_{(x,y)\in\mathbb{F}} P_i(x, y). \tag{2.4}$$

We also refer to this point as the *location* of the product i.

To calculate the best performance points for all products within the feasible region $\mathbb{F}$ we considered two methods.

The first method is solving the best performance points (x_i, y_i) for each product i, where $i = 1, \cdots, n$ from (2.4) continuously.

The second method is solving the best performance points (x_i, y_i) for each product i, where $i = 1, \cdots, n$ discretely. This is realized by confining x and y to a set of paired discrete values set $\mathbb{D}$, and then solving the following problem discretely.

$$(x_i, y_i) = arg \max_{(x_j,y_k)\in\mathbb{D}} P_i(x_j, y_k),\ \ i = 1, \cdots, n. \tag{2.5}$$

This method does not lose generality if we use a fine enough grid to discretize x and y.

In this project, we use the second method to localize the products. Moreover, if products distribution is not evenly, we can choose a series with certain properties to grid x and y. For example, if the product density decease as x and y increase, we could choose a geometric series with common ratio $q \geq 1$ to grid x and y.

2.5 Optimization Model

In the following text, we consider the discrete version of problem (2.3). Assume we grid x to $x_1, x_2, \cdots, x_N$ and y to $y_1, y_2, \cdots, y_M$. The objective function becomes the following:

$$\begin{aligned} -\int_{\mathbb{F}} \max_{1 \leq i \leq n} \{f_i(x,y)w_i\} dA &\approx -\sum_{j=1}^{N}\sum_{k=1}^{M} \max_{i=1,\cdots,n}\{w_i f_i(x_j, y_k)\} \\ &= -\sum_{j=1}^{N}\sum_{k=1}^{M}(S(x_j, y_k) \max_{i=1,\cdots,n}\{w_i P_i(x_j, y_k)\}) \end{aligned}$$

Then the optimization model becomes the following:

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\sum_{j=1}^{N}\sum_{k=1}^{M}(S(x_j, y_k) \max_{i=1,\cdots,n}\{w_i P_i(x_j, y_k)\}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i = k \\ & w_i \in \{0, 1\} \text{ for } i = 1, \cdots, n. \end{aligned} \tag{2.6}$$

This model can be used to select the optimal portfolio from an existing product pool as well as a guide to design new products. However, in general, this is a non-convex integer optimization problem and is NP-hard [13]. There are two different types of methods to apply to (2.6): exact and heuristic methods. In this project, we explore two heuristic methods in Chapter 3 and 4.

Chapter 3

Simulated Annealing

In this chapter, we will first give a description of simulated annealing and then describe how simulated annealing can be used to solve our optimization model (2.6). Finally, we will use a numerical example to illustrate that (2.6) can be solved effectively by simulated annealing.

3.1 Simulated Annealing Algorithm

Simulated annealing is a heuristic technique for solving non-convex optimization problems. It is designed to give an acceptable answer for typical problems in a reasonable time. It employs an iterative improvement strategy that attempts to perturb some existing suboptimal solution in the direction of a better solution.

Compared to the standard iteration methods, which only accept downhill movements and can easily be trapped in a local minima, simulated annealing allows perturbations to move uphill in a controlled fashion. The idea, as in iterative improvement, is to propose some random perturbation, such as moving the current solution to a new solution, then evaluate the resulting change in objective function $\triangle f$. If the objective function is reduced, the new solution is accepted as the starting point for the next move. However, if the objective function is increased, the move *may* still happen. The uphill movement is designed to escape from a local minima and is moderated by the current temperature T. Here, the temperature is simply a control parameter. At higher temperatures, the probability of large uphill moves in objective function value is large; at low temperatures the probability is small. One algorithm models this using a Boltzmann distribution: the probability of an uphill move of size $\triangle f$ at temperature T is $Pr[accept] = e^{\frac{\triangle f}{C.T}}$, where C is a constant.

By employing a cooling schedule, a sequence of decreasing temperatures, we moderate the acceptance of uphill moves over the course of the solution. Initially, the temperature parameter is high enough to permit an aggressive, essentially random search of the feasible region. Most uphill moves are allowed. As the temperature cools, fewer uphill moves are allowed. We tend to improve the value of the cost function here, but some local minima can also be avoided. At the coolest temperatures, the solution is close to freezing into its final form, and very few disruptive uphill moves are permitted. In this temperature regime, annealing closely resembles standard downhill-only iterative improvement.

In practice, we adjust the cooling schedule and trial time at each temperature by running multiple experiments and choosing heuristically. The algorithm terminates when the objective function is essentially "flat enough" for a few temperatures.

Note that the simulated annealing algorithm is not deterministic and will produce different answers each time it is run for the same problem. This is because of the probabilistic nature of choosing moves and accepting uphill moves. Theoretically, if we take enough trials at each temperature, simulated annealing will give a global optimum. In practice, to limit the running time, we place limits on trial numbers at each temperature. There is no guarantee of getting precisely the optimum answer in any annealing algorithm or even of getting the same answer on multiple runs. What annealing offers here is some probability of getting out of a local optimum. We could get a reasonably good optimum by setting appropriate temperature parameters and trial parameters.

Overall, simulated annealing is generally a practical approximation algorithm that is able to produce a good solution. It is extensively used in different applications. Much research has been devoted to both theoretical and experimental study. Recent examples include: [12], [14], [15] and [16].

In this project, to get a fast convergence rate and stable solution, we address the following:

(1) Keep track of the best solution encountered and its associated objective function value once the algorithm has been run.

(2) Choose the number of attempts at each temperature to be a multipli-

cation of the problem size and number of moves to attempt with a constant, which we refer to as the Trial Parameter. Adjust this parameter for different data sets to ensure convergence.

(3) Iteratively solve each problem for L times and choose the best solution among these L solutions as the approximate solution.

(4) Let $H(t)$ be the total number of times that algorithm hit the best approximate solution at temperature t. Define a convergence ratio $\rho = H(t)/L$. From the heuristic result, this ratio should be in $[0.3, 0.7]$. Use the convergence ratio to adjust the number of attempts at each temperature.

The resulting structure at temperature T is shown in Figure 3.1.

For the binary optimization problem (2.6), we choose the initial solution to be a vector with length n and k random entries with value 1. This ensures the feasibility of an initial solution. To find a better solution, we try to swap some entries of the current solution between 1 and 0 while keeping $\sum_{i=1}^{n} w_i = k$ based on the temperature T. The probability of uphill iterations is dependent on the temperature vector and defined as $P(T) = e^{\frac{-(fnew-fold)}{C.T}}$, where C is Boltzmann's Constant and T is the temperature parameter that decreases towards 0.

At each temperature, the structure is shown in algorithm 1.

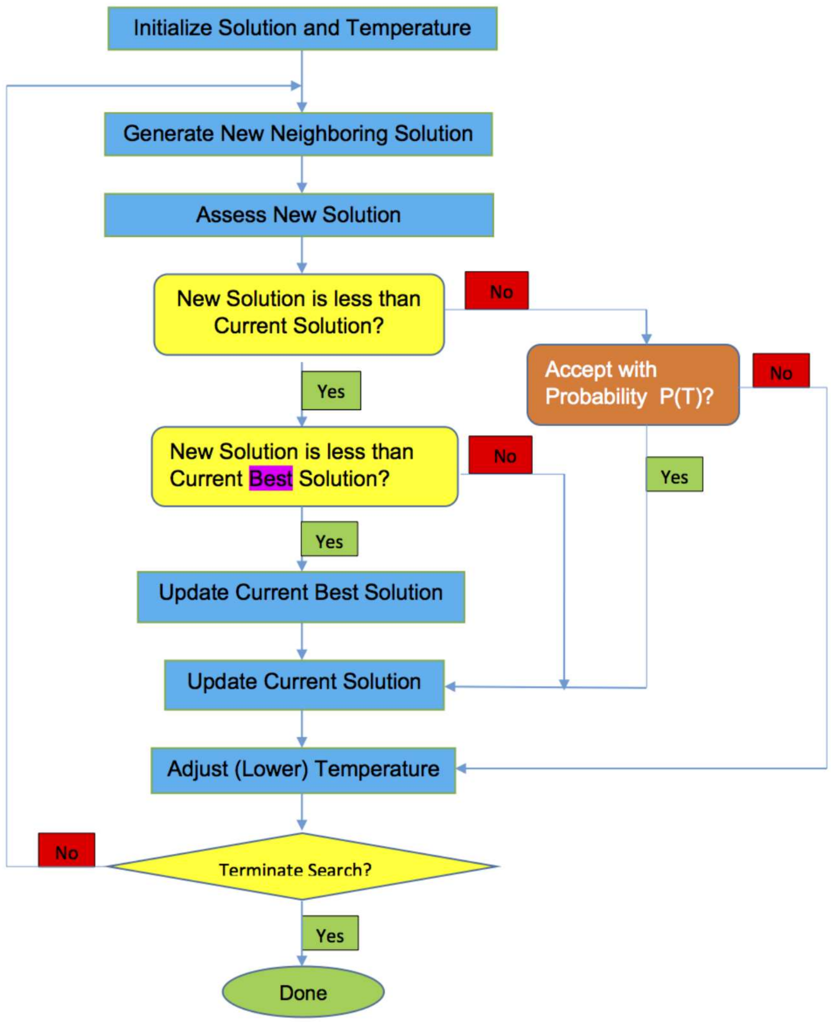

Figure 3.1: Simulated Annealing at Temperature T.

Algorithm 1 Simulated Annealing at temperature T

M = number of moves to attempt, T= current temperature

for $m = 1$ to M **do**

Generate a new neighbouring solution, evaluate the resulting objective function value $fnew$.

if $fnew < fold$ **then**

(downhill move: accept it)

Accept this new solution, and update the solution.

else

(uphill move: accept maybe)

Accept with probability $P(T) \leftarrow e^{\frac{-(fnew-fold)}{C.T}}$.

Update the solution if accepted.

end if

end for

3.2 Numerical Results

In this section, we apply the binary optimization model (2.6) to a pump manufacturer and use simulated annealing to solve the resulting optimization problem. We compare our method with the naive method, which picks the top ranked k items based on their individual market performance function values $\sum_{j=1}^{N}\sum_{k=1}^{M}(S(x_j, y_k)P_i(x_j, y_k)),\ 1 \leq i \leq n.$

We acquired the company's product design parameter data and three years sales transaction data from 2010 to 2013 for all of their products.

For the pump manufacture, we find that all the parameters are derived from two parameters: flow and pressure. Flow is measured in gallons per minute (GPM) and pressure is measured in feet (ft.). Hence, we choose x and y as flow and pressure, respectively. The range of flow is $[10, 700]$ and the range of pressure is $[6, 100]$. We choose the pump performance measure as a function of x, y. This function form is obtained from an engineering design perspective.

The sales data contains very detailed information about sold items, including: the two fundamental parameters flow and pressure, items sold in each transaction, time, and item price.

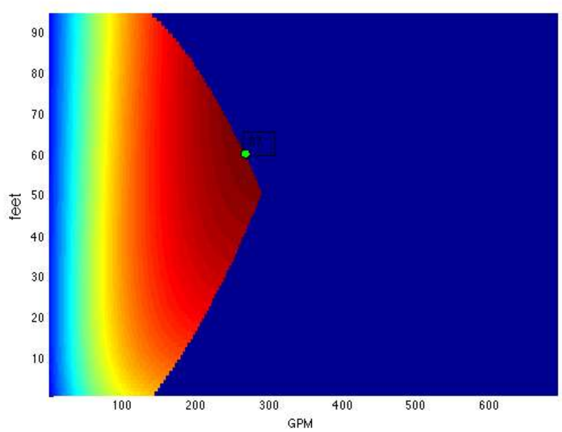

Figure 3.2: A Single Pump Plots.

Figure 3.2 is a two dimensional plot of function $f_i(x, y)$ for a single pump in the flow and pressure plane. The number in the figure is the index i for the plotted pump. The green point is the best performance point for the corresponding pump. Blue is background color of the plot. The red, yellow and cyan area is the feasible region of the corresponding pump. The value of $f_i(x, y)$ deceases as the color change from red to yellow and finally to cyan.

From the sales data, we found that the pump density decreases as its flow and pressure increases. Therefore, we use geometric series $1.5, 1.5^2, \cdots, 1.5^{10}$ and $1.3, 1.3^2, \cdots, 1.3^{10}$ to discretize the flow and pressure to 10 grids, respectively. With the method introduced in Section 2.3 and the interpolation method cubic spline we get the sales surface, which is plotted in Figure 3.3.

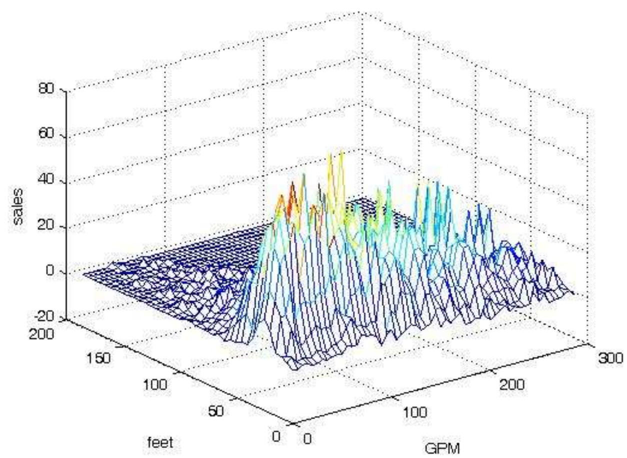

Figure 3.3: Interpolated Sales Surface

Using simulated annealing approach, we solve problem (2.6). We compare the numerical results obtained from (2.6) with the naive method, which picks the top ranked items based on their standalone value. Table 3.1 presents the results, varying the total number of items to be selected k.

Table 3.1: Profit Improvement Over Naive Method, $n = 100$

k	Naive Method Profit ($)	Optimal Profit ($)	Profit Improvement
3	11,536	12,651	9.665 %
5	11,640	13,337	14.582 %
10	11,806	13,844	17.261 %

As shown in Table (3.1), our model (2.6) outperforms the naive method. The reason is that there is quite a lot of overlapping effect among the products selected by the naive method. As for our model, we effectively eliminate the overlapping by using $\max_{1 \le i \le n} f_i(x, y)$. We use Figure 3.4 and Figure 3.5 to illustrate this this overlapping and its elimination, resp. Figure 3.4 is the ranked top 3 pumps. Figure 3.5 is the optimized pump portfolio for $k = 3$. Comparing these two figures, we observe that pump 76 and 87 have much more overlapping than pump 65 and pump 87.

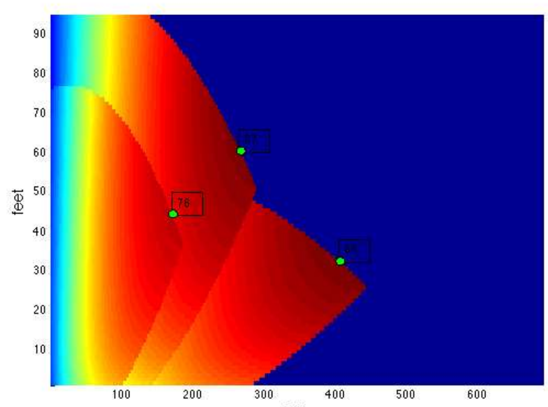

Figure 3.4: Ranked Top 3 Products Portfolio.

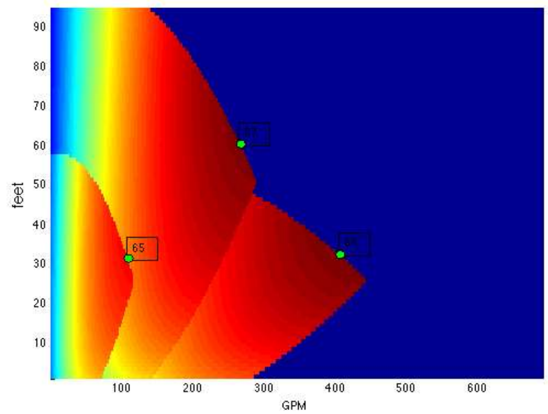

Figure 3.5: Optimized Pump Portfolio for $k = 3$.

Figure 3.6 plots the optimal portfolio for $k = 4$. Compare Figure 3.5 and 3.6. We can see that the optimal portfolio for $k = 3$ is not a subset of the optimal portfolio for $k = 4$. This illustrates that the optimal portfolio is not simply formed by progressively adding single items to the previous portfolio.

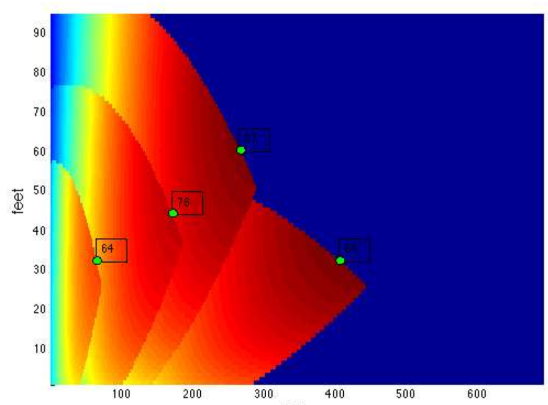

Figure 3.6: Optimized Product Portfolio for $k = 4$.

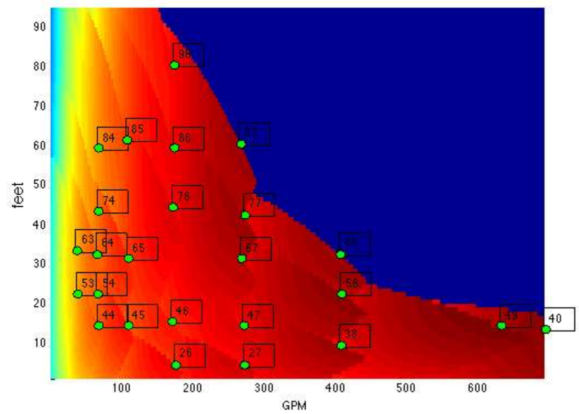

Figure 3.7: Optimized Product Portfolio for $k = 25$.

Another question that often arises in practice is the dependence of profit on the total number of product offerings. Figure 3.8 is the plot of the cumulative objective function value of the portfolio versus the number of pumps in the portfolio. Many companies choose to increase their new product offerings to keep competitive in the market. However, Figure 3.8 shows that the more product offerings, the less increase in portfolio value. The reason is that when the products offering exceeds certain level, the new adding products would results more increase in overlapping. Figure 3.7 is a plot of optimized portfolio with 25 products. From this plot, we can see that many products are overlapping each other. This profound result gives guidance to the number of products that should be selected.

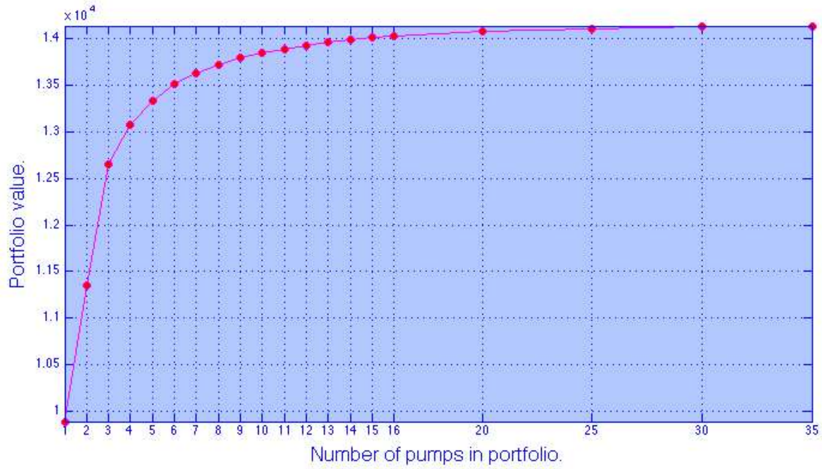

Figure 3.8: Optimized Portfolio Value v.s. Number of Products.

Remark 3.2.1. *Figure 3.8 illustrates that:*

1. *Relatively few pumps give very good coverage and profit.*
2. *The sensitivity of portfolio value to k decreases as k increases, where k is the number of pumps in the portfolio.*

3.3 Conclusion

The numerical result presented in Section 3.2 shows that simulated annealing can produce a good solution for the non-convex optimization problem (2.6). It is robust and efficient. With the convergence ratio, it is easy to adjust the simulated annealing parameters for different data sets.

Chapter 4

Optimization with Weighted Function

This chapter focuses on an alternative method for solving binary integer optimization problems.

4.1 Weighted Function

Consider the problem of minimizing a continuously differentiable function while restricting the variable to be binary, i.e., $\mathbf{x} \in \{0,1\}^n$. The problem is:

$$\min_{\mathbf{x}\in\{0,1\}^n} \{f(\mathbf{x})\}. \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^1$ is continuously differentiable.

Definition 4.1.1. *Weighted function $h(z) : \mathbb{R} \to \mathbb{R}$ is a continuously differ-*

entiable function of a single variable $z \in \mathbb{R}^1$ with the following properties:

$$\begin{aligned} &i) \quad h(0) = h(1) = 0, \\ &ii) \quad h(z) > 0 \text{ if } 0 < z < 1, \\ &iii) \quad \max_{0 \le z \le 1} \{h(z)\} = 1. \end{aligned} \tag{4.2}$$

Example 4.1.2. *Let*

$$h(z) = -4z^2 + 4z. \tag{4.3}$$

Example 4.1.3 (Piecewise quadratic)**.** *Choose $0 < \gamma \le \frac{1}{2}$. Define the continuous piecewise quadratic $h_\gamma(z)$ on 3 pieces: $[0, \gamma], [\gamma, 1-\gamma], [1-\gamma, 1]$:*

$$\begin{aligned} h^1_\gamma(z) &= -\tfrac{z^2}{\gamma^2} + \tfrac{2}{\gamma} z \text{ on } [0, \gamma], \\ h^2_\gamma(z) &= 1 \text{ on } [\gamma, 1-\gamma], \\ h^3_\gamma(z) &= -\tfrac{(1-z)^2}{\gamma^2} + \tfrac{2}{\gamma}(1-z) \text{ on } [1-\gamma, 1]. \end{aligned} \tag{4.4}$$

Note that h_γ is continuously differentiable. In addition, it is twice continuously differentiable at all points except $z = \gamma, 1-\gamma$. Note also that if $\gamma = \frac{1}{2}$, then $h_\gamma(z)$ reduces to $h(z)$ in (4.3).

Example 4.1.4 (Piecewise cubic)**.** *Choose $0 < \gamma < \frac{1}{2}$. Define the continuous piecewise cubic $h_\gamma(z)$ on the 3 pieces: $[0, \gamma], [\gamma, 1-\gamma], [1-\gamma, 1]$:*

$$\begin{aligned} h^1_\gamma(z) &= \tfrac{z^3}{\gamma^3} - \tfrac{3z^2}{\gamma^2} + \tfrac{3z}{\gamma} \text{ on } [0, \gamma], \\ h^2_\gamma(z) &= 1 \text{ on } [\gamma, 1-\gamma], \\ h^3_\gamma(z) &= \tfrac{(z-1)^3}{\gamma^3} - \tfrac{3(z-1)^2}{\gamma^2} + \tfrac{3(z-1)}{\gamma} \text{ on } [1-\gamma, 1]. \end{aligned} \tag{4.5}$$

Note that in this case, h_γ is twice continuously differentiable.

4.2 Weighted Function Approach For Binary Constraint Problems

Approach 4.2.1. *Replace integer minimization problem* (4.1) *with a sequence of problems* $k = 0, 1, 2\cdots$ *using a sequence of weights* $\gamma_0, \gamma_1, \cdots$, *and weight function* $h_{\gamma_k}(x_i)$ *with* $\gamma_0 = 0, \gamma_k > \gamma_{k-1}$ *and* $\{\gamma_k\} \to \infty$

$$\min_{0 \le \boldsymbol{x} \le 1} \{f(\boldsymbol{x}) + \gamma_k \sum_{i=1}^{n} h_{\gamma_k}(x_i)\} \tag{4.6}$$

where each minimization begins with the solution from the previous problem.

Now the binary integer minimization problem (4.1) becomes a continuous minimization problem (4.6). The weight function $\gamma_k \sum_{i=1}^{n} h_{\gamma_k}(x_i)$ will push non-integer solutions to integer solutions as $\gamma_k \to \infty$. We could use the optimization strategy to solve this unconstrained continuous optimization problem.

Conjecture 4.2.2. *Assume* $x_*(\gamma_k)$ *solves for* γ_k, *then for* γ_k *sufficiently large*

(1) $x_*(\gamma_k) \in \{0, 1\}^n$.

(2) $x_*(\gamma_k)$ *is a solution to* (4.1).

(3) The solution we get from Approach 4.2.1 would generally be different than rounding to integer from fraction.

Remark 4.2.3. *This approach can be used for the constrained problem as well. Replace*

$$\min_{\boldsymbol{x} \in \{0,1\}^n} \{f(\boldsymbol{x}) : c(\boldsymbol{x}) \ge 0, e(\boldsymbol{x}) = 0\} \tag{4.7}$$

with a sequence of problems

$$\min_{0 \leq \boldsymbol{x} \leq 1} \{f(\boldsymbol{x}) + \gamma_k \sum_{i=1}^{n} h_{\gamma_k}(x_i) : c(\boldsymbol{x}) \geq 0, e(\boldsymbol{x}) = 0\}, \tag{4.8}$$

with weights $\gamma_0, \gamma_1, \cdots$ *, with* $\gamma_0 = 0$ *and* $\gamma_k > \gamma_{k-1}, \{\gamma_k\} \to \infty$*.*

Note that rounding does not work in this case since the rounded solution may not be feasible.

4.3 Conclusion

Although Approach 4.2.1 lacks a theoretical proof, it is generally applicable and as easy to implement as a heuristic algorithm. The weight parameter γ_k needs to be identified by trials. Different problems would have different γ_k. Numerical experiments were conducted that suggest Conjecture 4.2.2 is true. We believe that more research, both theoretical and experimental, is needed to further assess the potential of this approach.

Chapter 5

Summary

In Chapter 2 an optimization model for production selection is proposed. This model can be used for product selection in an existing product pool as well as the new design product pool based on the product usage coverage and performance. We explicitly presented the approach for choosing product performance functions and sales surfaces.

Chapter 3 focused on the approach of solving (2.6) with simulated annealing and numerical results. We propose a heuristic method to adjust the simulated annealing parameters to get a good convergence rate and stable solutions. We numerically illustrate that our model gives a better product portfolio compared to just choosing the top ranked products. The difference of portfolio value becomes larger when the number of total products in the portfolio increases. Further, our model gives a benchmark suggestion about the size of the product portfolio.

We introduced a weighted function in Chapter 4 and proposed an alternative approach (4.2.1) for the binary integer optimization problem. Although the approach lacks theoretical support, it presents good properties heuristically if we choose the appropriate parameters.

For our model (2.6), since we maintain the flexibility of embedding existing products with newly designed products, it is better to use the discretization method with simulated annealing. In addition, different products with close parameters are very similar. With a fine enough discretization scale, we would be able to generate an optimal product portfolio.

References

[1] R. A Rutenbar, *S*imulated Annealing Algorithms: An Overview, IEEE Circuits and Devices Magazine, 26, (1989).

[2] Ward, P. T. McCreery, J. K. Ritzman, L. P. and D. Sharma, *C*ompetitive Priorities in Operations Management, Decision Sciences, 29, 1035-1046, (2007).

[3] R. Grussenmeyer, T. Blecker, *Requirements for the Design of a Complexity Management Method in New Product Development of Integral and Modular Products*, International Journal of Engineering, Science and Technology, 5(2), (2013).

[4] M. G. DeVries, *A Dynamic Model for Product Strategy Selection*, Industrial Development Research Program, Institute of Science and Technology, University of Michigan, (1963).

[5] R. H. Su, W.L. Pearn, *Production Selection for Newsboy-type Products with Normal Demands and Unequal Costs*, International Journal of Production Economics, 231(2), 214–222, (2011).

[6] D.B. Cao, Y. Yin, I. Kaku, *A Criterion of Production Model Selection for Building Material with Delivery Delay*, International Journal of Production Research, 48 (12), 3429–3443, (2010).

[7] C. S. Tang, R. yin, *The Implication of Costs, Capacity, and Competition on Product Line Selection*, Production, Manufacturing and Logistics, 200, 439–450, (2010).

[8] Honhon J. Eliashberg, R.Steinberg, *Marketing-production Joint Decision Making*, Handbook in Operations Research and Management Science: Marketing. 827–880, (1993).

[9] V. Gaur, D. Honhon, *Product Variety and Inventory Decisions Under a Locational Consumer Choice Model*, Management Science, 52(10), 1528–1543, (2010).

[10] T.H. Ho, C.S. Tang, *Research Advances in Product Variety Management*, Kluwer Publishers, (1998).

[11] A.G. Kok, M.L. Fisher, *Demand Estimation and Assortment Optimization Under Substitution: Methodology and Application*, Operation Research, 55, 1001–1021, d (2007).

[12] K. Haridass, J. Valenzuela, A. D. Yucekaya, T. McDonald, *Scheduling a Log Transport System Using Simulated Annealing*, Information Sciences 264, 302-316, (2014).

[13] G.L. Nemhauser, L. A. Wolsey, *Integer and Combinatorial Optimization (1st ed.).* New York: Wiley-Interscience, (1999).

[14] R.S. Tavares, T.C. Martins, M.S.G. Tsuzuki, *Simulated Annealing with Adaptive Neighbourhood: A Case Study in Off-line Robot Path Planning*, Expert Systems with Applications, 38(4), pp.2951–2965, (2011).

[15] M. Khalfe Nadeem, L.K. W.K. Shiv , *Simulated Annealing Technique to Design Minimum Cost Exchanger*, Chemical Industry and Chemical Engineering Quarterly, 17(4), p.409, (2011).

[16] A. Solonen, *Proposal Adaptation in Simulated Annealing for Continuous Optimization Problems* , Computational Statistics, pp.1–17, (2013).